



UNIVERSIDADE DA CORUÑA



comunicación
audiovisual+3d+interactiva

Facultade de Ciencias da Comunicación
Grao en creación dixital, animación e videoxogos
Curso 2022/2023

Flavor Savior

Ángela Arbe Núñez, Eva Quintela Balóira, Maite Real Mahía, Cristina Rodríguez Rodríguez

Tutor: Javier Taibo Pena

24 de junio de 2023

Español

Título: *Flavor Savior (La salvadora de los sabores)*

Palabras clave: videojuego, *RPG* por turnos, innovador, cocina, 2.5D

Flavor Savior es un videojuego *RPG* por turnos que utiliza minijuegos de cocina para combatir, su dimensionalidad es 2.5D y cuenta con una estética *cartoon* . El juego trata del viaje que emprende Tofi, la protagonista, para salvar el Reino Dulce de la amenaza que lo acecha. El jugador podrá explorar el reino, combatir contra diversos enemigos, recolectar objetos, crear poderosos compañeros que le ayuden en batalla y experimentar la historia del juego y sus ricos personajes. *Flavor Savior* está disponible para *PC (Windows)* y requiere un ratón y un teclado para ser jugado. El juego está disponible en tres idiomas: español, gallego e inglés.



Galego

Título: *Flavor Savior (A salvadora dos sabores)*

Palabras chave: videoxogo, *RPG* por turnos, innovador, cociña, 2.5D

Flavor Savior é un videoxogo *RPG* por turnos que emprega minixogos de cociña para combatir, é de dimensionalidade 2.5D e conta cunha estética *cartoon*. O xogo trata da viaxe que emprende Tofi, a protagonista, para salvar o Reino Doce da ameaza que o perturba. O xogador poderá explorar o reino, combatir contra diversos inimigos, recolectar obxectos, crear poderosos compañeiros que lle axuden en batalla e experimentar a historia do xogo e as súas ricas personaxes. *Flavor Savior* está dispoñible para *PC (Windows)* e require un rato e un teclado para ser xogado. O xogo está dispoñible en tres idiomas: galego, español e inglés.



English

Title: Flavor Savior

Key Words: videogame, turn-based RPG, innovative, cooking, 2.5D

Flavor Savior is a turn-based RPG videogame that uses cooking minigames as attacks, it has 2.5D point of view and a cartoon aesthetic. The game's story tells the journey that Tofi, the protagonist, has to go on in order to save the Sweet Kingdom from the threats that endanger it. Players can explore the Kingdom, fight against different enemies, collect items, craft powerful companions that will help them in battle and experience the game's story and rich characters. Flavor Savior is available on PC (Windows), and it requires mouse and keyboard to be played. The game is available in three different languages: english, galician and spanish.



Directores del proyecto:

Cristina García Aradas

Javier Taibo Pena

Testers:

Susana Fanego Rodríguez

David Alonso del Moral

Bruno Quintela Balóira



1. Información	2
2. Agradecimientos	5
4. Introducción y descripción	10
4.1 Introducción al proyecto	10
4.1.1 Tipo de proyecto	10
4.1.2 Motivación	10
4.1.3 Establecimiento de objetivos y retos principales	11
4.1.4 Metodología aplicada para la realización del TFG	11
4.2 Equipo de trabajo	12
4.2.1 Organigrama	16
5. Desarrollo del proyecto	17
5.1 Departamento de producción	17
5.1.1 Objetivos propuestos	17
5.1.2 Proceso	17
5.1.3 Conclusión individual	22



5.2 Departamento de diseño	23
5.2.1 Objetivos propuestos	23
5.2.2 Proceso	24
5.2.2.1 Jugabilidad y diseño de niveles	24
5.2.2.2 Historia y personajes	34
5.2.2.3 Sonido	39
5.2.3 Conclusión individual	43
5.3 Departamento de arte	44
5.3.1 Objetivos propuestos	44
5.3.2 Proceso	45
5.3.2.1 Estudio previo	45
5.3.2.2 <i>Concept</i>	46
5.3.2.3 <i>Sprites</i>	48
5.3.2.4 Animaciones	51
5.3.2.5 Cinemática	57
5.3.2.6 Modelado	63
5.3.3 Conclusión individual	66



5.4 Departamento de interfaces -----	<u>67</u>
5.4.1 Objetivos propuestos -----	<u>67</u>
5.4.2 Proceso -----	<u>68</u>
5.4.2.1 <i>Alpha</i>	<u>71</u>
5.4.2.2 <i>Beta</i>	<u>77</u>
5.4.2.2.1 Fondos	<u>86</u>
5.4.2.2.2 Diseño de portada	<u>88</u>
5.4.2.2.3 Pestañas y botones	<u>90</u>
5.4.2.2.4 Logos e iconos	<u>91</u>
5.4.2.3 <i>Golden</i>	<u>95</u>
5.4.3 Conclusión individual -----	<u>98</u>
5.5 Departamento de programación -----	<u>100</u>
5.5.1 Objetivos propuestos -----	<u>100</u>
5.5.2 Proceso -----	<u>101</u>
5.5.2.1 <i>Pre-Alpha</i>	<u>103</u>
5.5.2.2 <i>Alpha</i>	<u>109</u>
5.5.2.2.1 Sistema de exploración	<u>109</u>
5.5.2.2.2 Sistema de batalla	<u>113</u>
5.5.2.3 <i>Golden</i>	<u>121</u>
5.5.3 Conclusión individual -----	<u>125</u>



6. Conclusiones generales	127
7. Anexos y apéndices	128
8. Bibliografía	129
8.1 Bibliografía común	129
8.2 Bibliografía individual	130
8.2.1 Ángela Arbe Núñez	131
8.2.2 Eva Quintela Baloira	132
8.2.3 Maite Real Mahía	133
8.2.4 Cristina Rodríguez Rodríguez	134



4.1. Introducción al proyecto

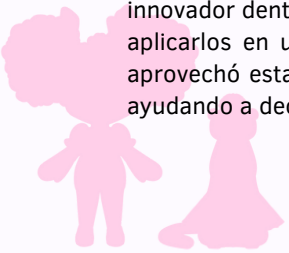
4.1.1. Tipo de proyecto

Flavor Savior es un videojuego *RPG* por turnos que permite a los jugadores explorar un mundo único y colorido. Los usuarios podrán descubrir los peligros que acechan al Reino Dulce conforme avancen en la historia principal. El juego cuenta con una gran cantidad de contenido secundario en el que los jugadores podrán desbloquear elementos coleccionables, fabricar compañeros que les ayuden en batalla y descubrir historias complementarias a la trama principal. El innovador sistema de combate mezcla el estilo clásico de *RPG* por turnos con interesantes mecánicas de minijuegos de cocina. Gracias a su versátil jugabilidad, cada jugador podrá personalizar su experiencia de juego, así como disfrutar el contenido en tres idiomas diferentes.

El público objetivo se centra en personas mayores de 7 años, independientemente de que cuenten con experiencia en el género, ya que el juego trata de ser inclusivo con aquellas personas novatas gracias a su gran cantidad de tutoriales.

4.1.2. Motivación

El equipo decidió escoger este producto como proyecto final de grado debido a que se quería crear un videojuego innovador dentro del género *RPG*. Además, se buscaba explorar todos los conocimientos aprendidos a lo largo del grado y aplicarlos en un proyecto complejo y de mayor profundidad, tanto en historia como en jugabilidad. Por otro lado se aprovechó esta oportunidad para crear el primer capítulo del videojuego y ver la respuesta del público general hacia él, ayudando a decidir si en un futuro se continuaría con el desarrollo del producto tras la creación de esta demo.



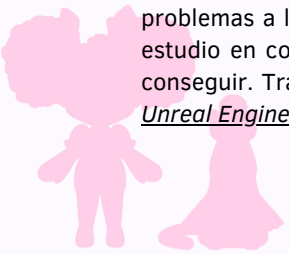
4.1.3. Establecimiento de objetivos y retos principales

El objetivo principal establecido a la hora de realizar el proyecto fue contar con una jugabilidad pulida e interesante. El eje central del juego es su *gameplay*, por lo que todas las mecánicas incluidas debían ser perfectamente funcionales. Se quería conseguir un producto final que contase con una amplia variedad de misiones que permitiesen explorar en profundidad la historia del juego, coleccionables que mantuviesen el interés del jugador y un gran elenco de personajes que enriqueciesen la experiencia.

4.1.4. Metodología aplicada para la realización del TFG

Para una correcta y equitativa división de tareas se decidió repartir la carga de trabajo entre diferentes departamentos. Cada uno de estos departamentos contaría con una persona responsable, pudiendo recibir asistencia de otras integrantes del equipo en caso de ser necesario.

Para afrontar el proceso de desarrollo se dividió el tiempo disponible en tres etapas denominadas *Alpha*, *Beta* y *Golden*. Antes de comenzar la etapa *Alpha* cada departamento realizaría una investigación. El objetivo era conocer los posibles problemas a la hora de realizar el desarrollo y, de esta manera, poder prevenirlos. Así mismo en esta etapa se realizó un estudio en conjunto para establecer los puntos clave a desarrollar del proyecto, así como las metas que se pretendían conseguir. Tras valorar todos los motores en los que el equipo contaba con soltura para desarrollar el proyecto, *Godot* y *Unreal Engine*, se optó por utilizar *Unreal Engine* en su versión 4.27.



4.2. Equipo de trabajo

El equipo de trabajo estuvo formado por Ángela Arbe, Eva Quintela, Maite Real y Cristina Rodríguez. Se decidió dividir al equipo por departamentos para organizar el reparto de tareas de manera más equitativa:

Departamento de arte

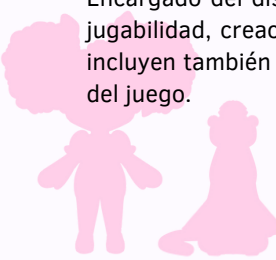
Encargado de la apariencia visual del juego. Dentro de este departamento están las tareas de *concept art*, diseño de personajes, creación de *sprites*, modelado 3D, *shading*, animación de *sprites*, cinemáticas y diseño de interfaces.

Departamento de producción

Encargado de la organización tanto del equipo como del proyecto. En él encontramos tareas como la creación del calendario del proyecto, identificación y distribución de tareas, seguimiento del proyecto, seguimiento de fechas de entrega y la creación de documentos tales como listas de *assets*, guías para *testers*, etc.

Departamento de diseño

Encargado del diseño del *gameplay*. Cuenta con tareas como el diseño de los minijuegos, diseño de niveles, balanceo de la jugabilidad, creación de misiones y la colocación de diferentes elementos interactivos en el mapa. En este departamento se incluyen también las tareas referentes al guion del juego y su adecuación a los diferentes idiomas, así como el diseño sonoro del juego.



Departamento de interfaces

Encargado de los elementos visuales que guían al jugador por el juego. Dentro de este departamento están las tareas de diseño, creación y programación de menús, creación del *HUD*, creación de pantallas de carga, creación de tutoriales, creación del sistema de inventario, creación del mapa interactivo y los sistemas de diálogo, idioma y compañeros.

Departamento de programación

Encargado del funcionamiento del juego, incluyendo su optimización. Sus tareas están divididas en el sistema de exploración, donde se incluye el sistema de progresión del personaje y de las misiones o las interacciones con el entorno; y el sistema de batalla basado en turnos, con el funcionamiento de los minijuegos y la batalla contra el enemigo final.



Ángela Arbe:

-Arte: Dirección artística, *concept art*, diseño de personajes, creación de *sprites*, animaciones de *sprites*, creación de las cinemáticas, arte de las interfaces.

Eva Quintela:

-Apoyo del equipo de producción: Implementación del proyecto en [GitHub](#).

-Programación: Creación del sistema de batalla por turnos con ataques basados en minijuegos, los combates tanto predefinidos como aleatorios, la batalla contra el enemigo final, el sistema de guardado, el sistema de niveles, el sistema de misiones principales y secundarias, la cámara adaptada a los cambios de zona, las mecánicas de exploración como objetos enterrados o *NPCs* interactivables y la optimización.

-Apoyo del equipo de interfaces: Implementación de los tutoriales de los minijuegos y las pruebas de Cantino.

-Apoyo del equipo de arte: *Shading* e implementación de los *sprites*.



Maite Real:

-Producción: Organización de tareas, calendario de fechas de entrega y documentos.

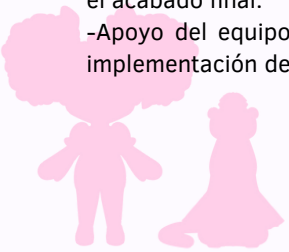
-Diseño: Diseño de los minijuegos, diseño de niveles (creación del mapa, colocación de elementos, creación de misiones), balance del juego, escritura del guion del juego (basada en la historia previa escrita por Eva Quintela), textos en inglés, español y gallego, búsqueda de melodías y elementos de audio, grabación de efectos de sonido y formación de una banda sonora coherente a partir de ellos.

-Apoyo del equipo de arte: ayuda en aquellas tareas en las que la encargada de arte, Ángela Arbe, necesitase asistencia. Esto incluyó *concept art* y la totalidad del modelado 3D del juego.

Cristina Rodríguez:

-Interfaces: Diseño de los menús y elementos visuales, programación de las mecánicas asociadas a las interfaces como el uso de objetos curativos, la fabricación de los compañeros, el menú de ajustes con el sistema de traducción, el *mapping* de controles, las pantallas de carga y demás elementos visuales con información para el jugador. Implementación del arte para el acabado final.

-Apoyo del equipo de diseño: Realización del guion de las misiones secundarias, implementación del audio en el juego e implementación de las traducciones.



4.2.1 Organigrama

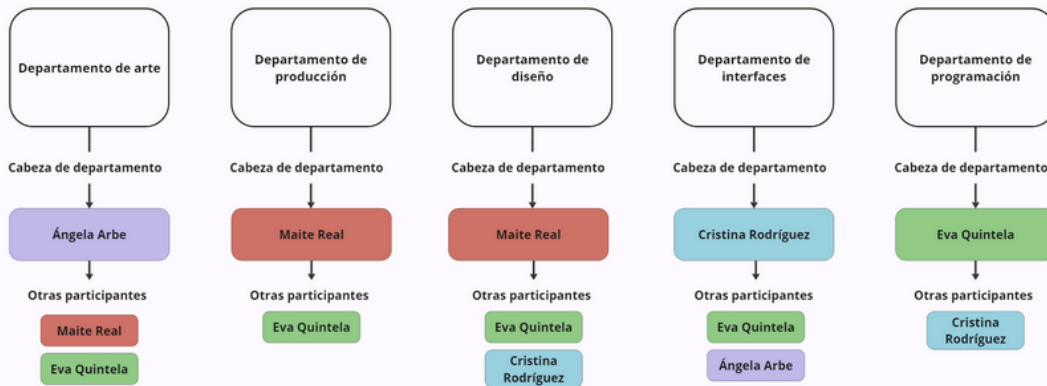


Fig.1 Estructura del equipo. Fuente propia.

En el organigrama (Fig.1) se muestra cómo el equipo está dividido por departamentos independientes que cuentan con una cabeza de departamento. Esta persona es la responsable de las decisiones tomadas acerca de dicho departamento. Debido al reducido tamaño del equipo y las dimensiones del proyecto, se decidió que aunque cada departamento contase con una persona responsable, que haría la mayor carga de trabajo, otras personas pudiesen ayudar en caso de ser necesario, éstas serían aquellas denominadas como “Otras participantes”.

5.1. Departamento de producción

5.1.1 Objetivos propuestos

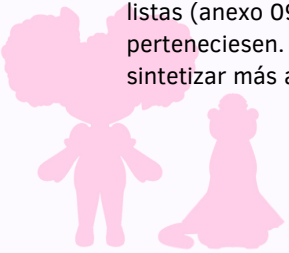
Los objetivos principales que se querían lograr en el departamento de producción estaban centrados especialmente en una exhaustiva organización de las tareas necesarias y su correspondiente seguimiento a lo largo del proyecto.

Una de las primeras metas marcadas fue la recopilación en listas de todos aquellos elementos que formarían parte del juego (anexo 09). A su vez, otro de los objetivos clave fue la organización temporal del trabajo, ya que el elevado número de tareas a realizar requería que aquellos elementos de mayor prioridad estuviesen completamente desarrollados en una etapa temprana.

Para finalizar, se estableció como meta el contar con un equipo de *testers*, que pudiese identificar problemas tanto en la jugabilidad como en la experiencia general del proyecto.

5.1.2 Proceso

Al comienzo del desarrollo del proyecto, se reunieron todos aquellos elementos necesarios para la realización del juego en listas (anexo 09). Dentro de dichas listas, los diferentes elementos fueron organizados en función del departamento al que perteneciesen. Se separaron las diversas etapas de desarrollo por las que debería pasar cada elemento, para poder sintetizar más adecuadamente el tiempo real que cada uno requeriría (Fig. 2).



INTERFACES

C: concepto · IMP: implementado · LOOK: Look final (Las etapas en AMARILLO son las actuales, las grises, las completadas)

Tarea	Prioridad	Etapas			Comentarios
Pantallas comunes					
Pantalla de título	Alta	C	IMP	LOOK	
Menú de opciones	Alta	C	IMP	LOOK	
In- game					
HUD	Alta	C	IMP	LOOK	
Inventario base	Alta	C	IMP	LOOK	
Mapa	Media	C	IMP	LOOK	
Recetario	Media	C	IMP	LOOK	
Compañeros	Media	C	IMP	LOOK	
Conjuntos	Media	C	IMP	LOOK	
Objetos	Alta	C	IMP	LOOK	
Combate HUD	Alta	C	IMP	LOOK	
Pantalla de muerte	Alta	C	IMP	LOOK	
Otros					
Créditos	Alta	C	IMP	LOOK	

Fig. 2 Lista de elementos estéticos pertenecientes a las interfaces. Fuente propia.

Tras reunir todos los elementos necesarios se llegó a la conclusión de que el volumen de trabajo no era adecuado para el proyecto, ya que se contaba con demasiados elementos innecesarios o secundarios que requerían invertir demasiado tiempo del equipo en ellos. Para optimizar el tiempo disponible para la realización del proyecto se decidió replantear los elementos necesarios, priorizando a su vez aquellos de mayor importancia para conseguir la experiencia de juego deseada por el equipo.

Una vez los elementos a desarrollar para el proyecto fueron definidos, se implementaron en la herramienta Plaky.

Plaky es una herramienta en línea que permite la organización de proyectos creativos en equipo. Se eligió esta herramienta tras realizar una previa investigación de aquellas disponibles. Para dicha investigación se tuvieron en cuenta experiencias pasadas en las que la organización de trabajos realizados en equipo también contaba con la ayuda de herramientas similares, como es el caso de asignaturas como “Proyecto de animación” del segundo curso del grado. Algunas de las alternativas exploradas fueron ShotGrid, que se encuentra más orientada a proyectos de animación, y Trello. Finalmente Plaky fue la elegida debido a que su interfaz resultaba más intuitiva para el equipo.

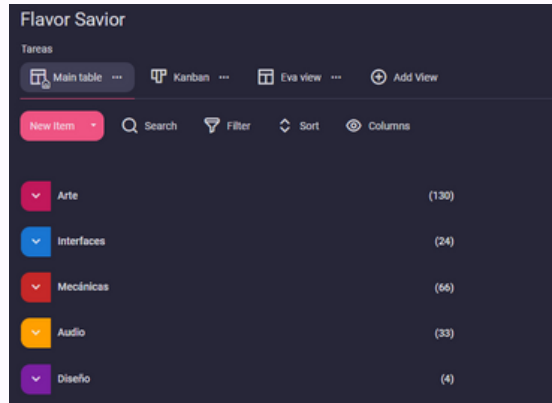


Fig. 3 Organización de los departamentos en Plaky.

Fuente propia.

5. Desarrollo del proyecto

El siguiente paso en el departamento de producción fue la creación de un calendario (Fig. 4) en el cual se estableciesen fechas límite para aquellas tareas que resultasen clave en el desarrollo. A la hora de crear el calendario, se dividió el proceso de desarrollo usando el sistema empleado en la asignatura “Proyecto de videojuego” del tercer curso. En dicha asignatura, el tiempo de desarrollo fue repartido en tres grandes versiones del proyecto: *Alpha*, en la que debía estar la jugabilidad básica implementada, *Beta*, en la que el apartado artístico debía añadirse junto con otras mecánicas más secundarias y *Golden*, en la que se habrían corregido los fallos menores y detalles de las anteriores etapas.

Se marcaron fechas clave para cada una de estas etapas, así como las tareas que debían estar completadas para cada una de ellas. Se marcaron reuniones semanales o bisemanales con el tutor del proyecto, de modo que se pudiese comprobar el correcto cumplimiento del calendario de desarrollo.

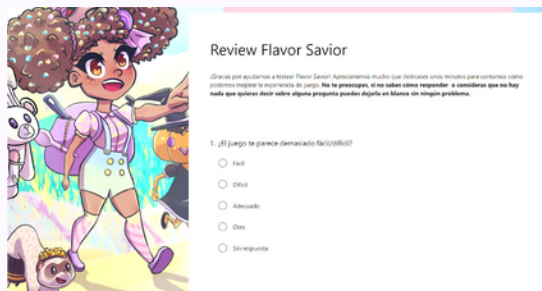
22 febrero	8 marzo	22 marzo	5 abril
PRE ALPHA Escenario completo Combate funcional Mecánicas de exploración	ALPHA Minijuegos Sistema de fabricación Misiones redactadas Procedimiento cinemáticas Inventario	ALPHA Funcionamiento de diálogos Funcionamiento de las misiones Redacción de diálogos	ALPHA Batalla de Cucumis Sistema de compañeros Sistema de logros Sistema de Cantino Sistema de guardado Funcionalidad al 100%
19 abril	3 mayo	17 mayo	31 mayo
BETA Estética de Tofi Estética de Ferri Modelado de casas básicas Sprite de árbol Shading Corrección de errores	BETA Modelado de cocinas de Lolo Cajas de plátanos Comienzo decoraciones Corrección de errores	BETA Estética personajes Modelo del árbol del bosque Decoraciones Sonido Corrección de errores	BETA Finalización de arte Arreglos menores Corrección de errores
7 junio	23 de junio		
TESTING Corrección de errores Detalles menores	GOLDEN		

Fig. 4 Calendario de tareas del equipo. Fuente propia.

Para asegurar que el producto final tuviese un acabado libre de errores de jugabilidad, así como para recibir sugerencias acerca del proyecto, se creó un programa de *testers*. Para este programa se contactó con personas dispuestas a emplear su tiempo en el juego. Se trató de que dichas personas tuviesen una experiencia variada en videojuegos del género, para poder tanto satisfacer las necesidades de los jugadores experimentados, como crear una experiencia accesible para personas que se iniciasen en este estilo de jugabilidad.

Se creó un servidor propio para *testers* en la red social [Discord](#), de manera que todos los participantes pudiesen ser notificados en el momento en el que una nueva versión del producto estuviese disponible. El contar con un servidor también permitía que los *testers* pudiesen interactuar entre sí y compartir su experiencia de juego.

Para aquellos participantes que no quisiesen crear una cuenta en [Discord](#), o prefiriesen interactuar únicamente con el equipo de desarrollo se creó un formulario con [Microsoft Forms](#) (Fig.5).



The image shows a screenshot of a Microsoft Forms survey. On the left is a colorful illustration of a character with large, curly brown hair, wearing a purple and white outfit, holding a white rabbit. The survey text is in Spanish and asks for feedback on the game 'Flavor Savior'. The first question is '¿El juego te parece demasiado fácil/difícil?' with radio button options: 'Fácil', 'Difícil', 'Adecuado', 'Otro', and 'Sin respuesta'.

Review Flavor Savior

¡Gracias por ayudarnos a probar Flavor Savior! Nos encantamos mucho que dediques unos minutos para compartir cómo podemos mejorar la experiencia de juego. No te preocupes, si no sabes cómo responder, o simplemente que no hay nada que quieras decir sobre alguna pregunta puedes dejarlo en blanco sin ningún problema.

1. ¿El juego te parece demasiado fácil/difícil?

- Fácil
- Difícil
- Adecuado
- Otro
- Sin respuesta

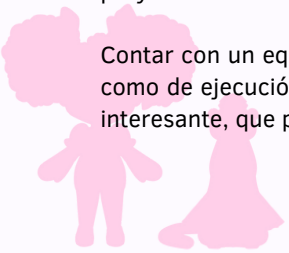
Fig. 5 Formulario creado para los participantes de *testing*. Fuente propia.

Dado a que el equipo compartiría un proyecto común realizado en Unreal Engine, se acordó emplear el sistema oficial de este motor para la nomenclatura de los diferentes elementos del proyecto. Dicha nomenclatura, disponible para su consulta en la documentación de Unreal Engine, se caracteriza por marcar un primer campo con el tipo de elemento y establecer un nombre distintivo en el segundo campo, siendo ambos divididos por una barra baja. Se eligió esta nomenclatura debido a ser la empleada a lo largo del grado en los proyectos desarrollados con este motor, por lo que todo el equipo se encontraba familiarizado con ella. Además, se trata de la nomenclatura recomendada por Epic Games y de uso habitual en la industria.

5.1.3 Conclusión individual

El trabajo realizado por el departamento de producción en etapas tempranas del desarrollo fue clave para que el equipo pudiese contar con una visión global del proyecto en cuanto a carga de trabajo y tiempo disponible para realizar el proyecto. La reducción de elementos realizada previamente al inicio del desarrollo resultó de extrema importancia, ya que ayudó al equipo a centrar los objetivos que se querían conseguir con el proyecto y a reducir aquellos elementos de menor importancia que hubiesen resultado un malgasto de recursos, así como haber obligado a retrasar elementos de mayor importancia. La cantidad de elementos recortados por el equipo podría haber sido mayor al contemplar la magnitud del proyecto en su fase final del desarrollo.

Contar con un equipo de *testers* dedicado al juego resultó de gran ayuda a la hora de encontrar errores tanto conceptuales como de ejecución. La opinión de las personas participantes permitió contar con una variedad de puntos de vista realmente interesante, que permitió al equipo crear un producto disfrutable tanto por gente experta en videojuegos como primeriza.



5.2. Departamento de diseño

5.2.1 Objetivos propuestos

Los objetivos principales que el departamento de diseño debía cumplir eran principalmente la creación de una jugabilidad balanceada e interesante, que pudiese satisfacer a los jugadores más experimentados así como garantizar una experiencia amena para los jugadores casuales o principiantes en el género.

Otro de los elementos de mayor importancia fue la creación de una historia sencilla pero que resultase divertida, repleta de personajes memorables. En el proceso de escritura del guion se buscaba introducir a los jugadores al *lore* del mundo de *Flavor Savior*, explicando aquellos elementos más importantes del universo narrativo del juego con toques de humor.

En este departamento también se incluyen las labores relacionadas con la música y los efectos sonoros, que debían ser coherentes tanto con la estética como con el tono general del producto.



5.2.2 Proceso

5.2.2.1 Jugabilidad y diseño de niveles

En la etapa previa al inicio del desarrollo fue de gran importancia definir al detalle los elementos pertenecientes a la jugabilidad. Al tratarse de un juego *RPG* por turnos, dichos elementos incluían fórmulas de daño tanto del jugador como enemigos, objetos consumibles y los diferentes minijuegos que servirían como ataques a los jugadores.

Para la realización de las fórmulas de daño del juego (anexo 03), se tuvieron en cuenta diferentes referencias del mundo de los videojuegos (Fig. 6 y 7).

A la hora de planificar las fórmulas era crucial tener en cuenta el tipo de combate del juego, ya que en este el jugador se enfrenta en solitario contra uno o varios enemigos a la vez. Esto condiciona la formulación de las estadísticas, dado que el jugador debe ser capaz de soportar los ataques de varios enemigos a la vez sin ser lo suficientemente fuerte como para poder ganar sin esfuerzo cuando lucha contra un solo enemigo.





Fig. 6 Referencia tomada para el combate.
(*Golden Sun*, 2001). Fuente propia.



Fig. 7 Referencia tomada para el combate.
(*Honkai: Star Rail*, 2023). Fuente propia.

Otro factor que se tuvo en cuenta a la hora de balancear el combate fue el hecho de que en el juego se encuentran cuatro tipos de enemigos diferentes: enemigos básicos, enemigos fuertes, jefe intermedio y jefe final. Cada uno de estos tipos de enemigo cuenta con unas estadísticas de combate diferentes, contando el jefe final incluso con una jugabilidad diferente. A la hora de balancear las estadísticas de combate era necesario tener en cuenta que el jugador debía ser lo suficientemente fuerte para derrotar a un enemigo poderoso, pero no lo suficiente como para convertir a los enemigos menores en irrelevantes.

Teniendo en cuenta estos factores se crearon cuatro minijuegos diferentes que servirían de ataques al jugador: "Bananarang", "Batidor de merengue", "Horneado de pastel" y "Bate piruleta". Tanto el "Bananarang" como el "Bate piruleta" corresponden a ataques individuales a un enemigo. Por otro lado, "Batidor de merengue" y "Horneado de pastel" son ataques de área (Fig. 8).

Estos minijuegos, se desbloquean a lo largo de la trama. Debido a ello, los ataques que se desbloquean al inicio de la historia serán menos fuertes que los minijuegos más avanzados.

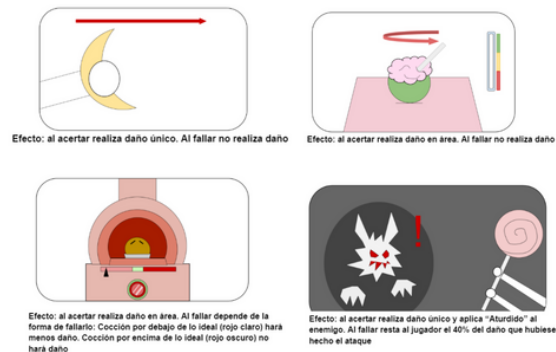


Fig. 8 Bocetos explicativos de los minijuegos. Fuente propia.

Para que el jugador contase con variedad en sus ataques desde el inicio de la historia se decidió distribuir los ataques de manera que al inicio se conseguirían un ataque único y un ataque en área. Los minijuegos reservados para más adelante en la historia realizarían más daño, pero para evitar que los otros dos minijuegos pudiesen quedar obsoletos, estos dos minijuegos finales son más complejos de realizar. Del mismo modo, para que el jugador no utilizase únicamente los ataques más fuertes al llegar a determinado punto de la historia, se implementó un sistema de *cooldown* en los minijuegos. Este sistema implantó una cantidad determinada de turnos de recarga para cada ataque. Durante este tiempo de recarga dicho ataque no podría ser usado. La cantidad de *cooldown* dependía de la fuerza del ataque:

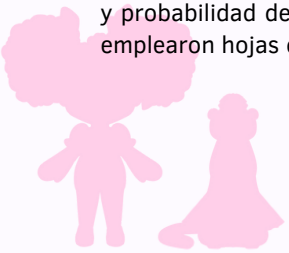
"Bananarang": un turno.

"Batidor de merengue": dos turnos.

"Horneado de pastel": tres turnos.

"Bate piruleta": cuatro turnos.

Se crearon diferentes estadísticas en el personaje principal que influirían en las fórmulas de daño realizado, daño recibido y probabilidad de huida. Para organizar de manera más visual los diferentes cambios de balance de estas estadísticas se emplearon hojas de cálculo (Fig. 9).



5. Desarrollo del proyecto

Level	Salud jugador	Ataque jugador	Salud enemigo	Daño enemigo	Fuerte Salud	Fuerte ataque	Defensa jugador
1	200	20	50	12			2
2	225	20	50	12			2
3	250	25	50	12			2
4	275	25	50	12			2
5	300	30	100	12			2
6	325	30	100	24			4
7	350	35	100	24			4
8	375	35	100	24			4
9	400	40	150	24			4
10	425	40	150	24	250	56	4
11	450	45	150	36	250	56	6
12	475	45	150	36	250	56	6
13	500	50	200	36	250	56	6
14	525	50	200	36	300	56	6
15	550	55	200	48	300	68	8
16	575	55	200	48	300	68	8
17	600	60	250	48	300	68	8
18	625	60	250	48	350	68	8
19	650	65	250	60	350	80	10
20	675	65	250	60	350	80	10

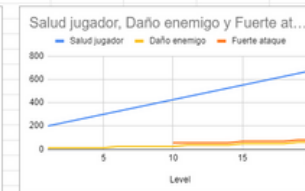


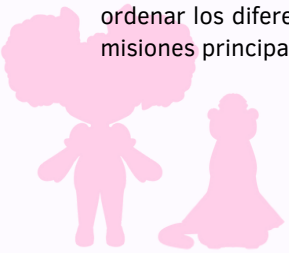
Fig. 9 Ejemplo de hoja de cálculo empleada en el balance de las estadísticas del jugador. Fuente propia.

Uno de los mayores problemas encontrados a la hora de realizar el balance fue crear un equilibrio entre una dificultad que resultase divertida para los jugadores experimentados y una dificultad que no excluyese a los jugadores principiantes. Teniendo en cuenta las diferentes opiniones del equipo de *testing*, se realizaron diversos balances, hasta alcanzar una solución satisfactoria para ambos públicos. El juego cuenta con diferentes objetos curativos y potenciadores, que permiten a los jugadores menos experimentados disfrutar de batallas de menor dificultad. Estos objetos permiten además a los jugadores que buscan un mayor reto enfrentarse a batallas para las que no cuentan con el nivel suficiente siempre que planifiquen adecuadamente su inventario.

Para facilitar la experiencia a los jugadores más casuales se crearon puntos de curación que recuperan la vida completa del jugador cada vez que se usan, cuentan además con un número infinito de usos.

Todos los detalles acerca de los minijuegos, las fórmulas de daño y los objetos consumibles, se encuentran recopilados en el anexo 03.

En cuanto al diseño de niveles, en la etapa previa al desarrollo se realizó una investigación sobre la historia original, para ordenar los diferentes sucesos de la historia en el tiempo y de esta manera poder realizar una primera aproximación a las misiones principales y el mapa del juego (Fig. 10).



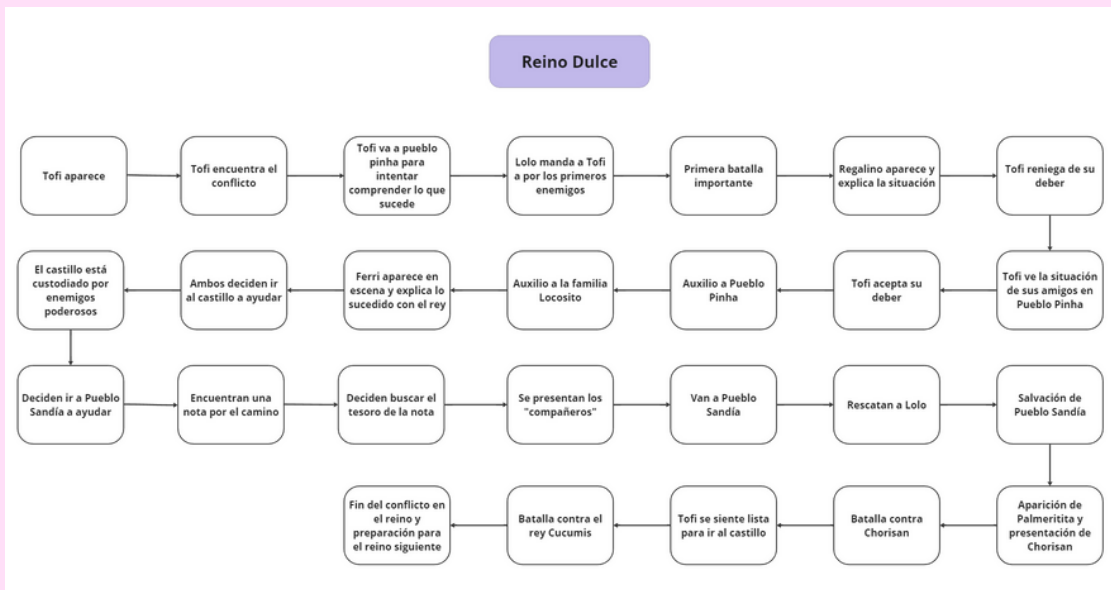


Fig. 10 Esquema organizativo de los sucesos de la historia. Fuente propia.

Partiendo de este esquema (Fig. 10), se realizaron las primeras aproximaciones del mapa, con el objetivo de colocar las zonas clave de la historia de manera coherente en el espacio (Fig. 11).

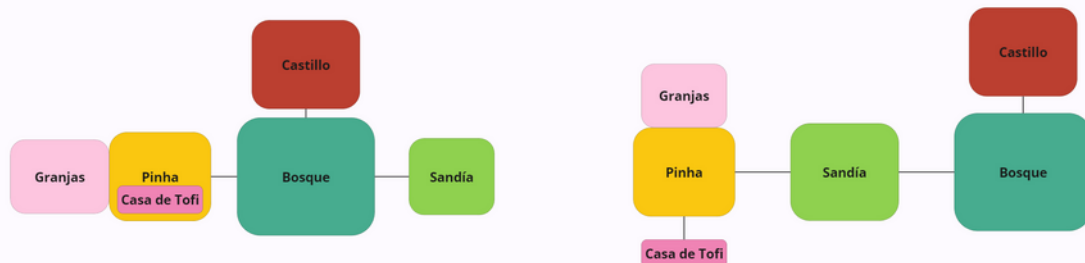


Fig. 11 Primeras aproximaciones al aspecto del Reino Dulce. Fuente propia.

A la hora de realizar el mapa se barajaron diversas opciones. El castillo resultó una zona complicada de situar. Posicionarlo en el centro del mapa resultaba interesante desde el punto de vista narrativo, puesto que de esta manera sería el corazón del reino, pero no resultaba lógico desde el punto de vista del jugador, ya que se le obligaría a pasar constantemente por un área que toma importancia hacia el final de la historia. Por este motivo se decidió no situar el castillo en el centro del mapa.

Otro conflicto a la hora de crear el mapa fue la posición de los pueblos. En un principio ambos pueblos se situaban juntos (Fig.11, derecha), pero esto dejaba poco margen para crear una zona explorable amplia entre ambos. Dado que la mayoría de misiones secundarias orbitan alrededor de ambos pueblos se descartó esta posibilidad para no concentrar demasiado contenido en una única zona del mapa.

Tras realizar diversas iteraciones, se llegó a un boceto final (Fig. 12), que más adelante se refinaría en una cuadrícula para visualizar el tamaño real que tendría en el motor de juego (Fig. 13).

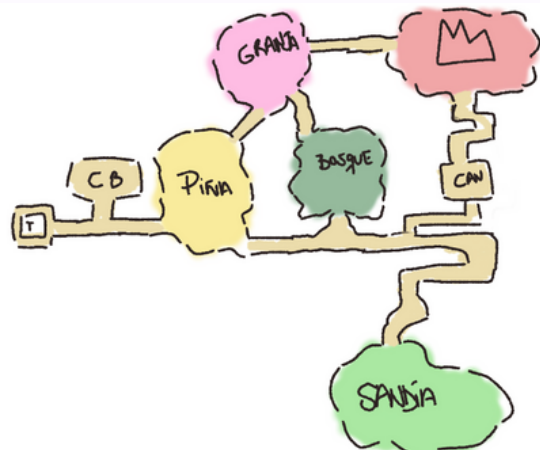


Fig. 12 Boceto conceptual final del aspecto del Reino Dulce. Fuente propia.

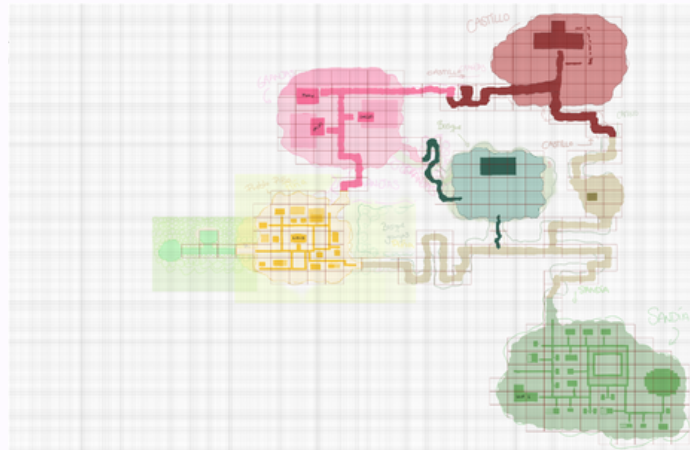


Fig. 13 Aspecto final del mapa en la etapa previa al diseño de niveles en Unreal Engine. Fuente propia.

Una vez dentro del motor de juego, se realizó un *blocking* que dividía las diversas áreas en colores. Ya en la fase *Beta* del proyecto, se comenzaron a introducir los elementos estéticos en el mapa, hasta llegar a su aspecto final. Dichos elementos se dividían en modelos 3D como edificios o diversas decoraciones y *sprites* 2D correspondientes a personajes o elementos de follaje.

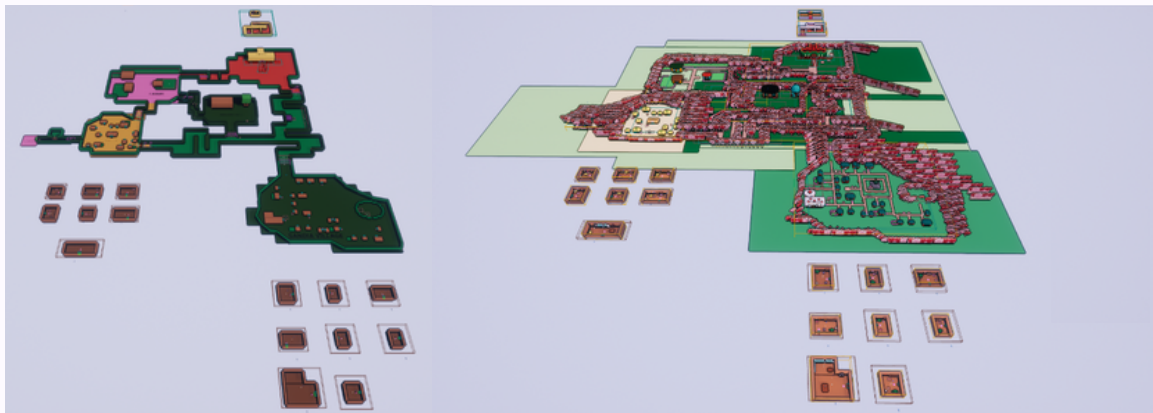


Fig. 14 A la izquierda: aspecto del mapa en la etapa de *blocking*. Fuente propia. A la derecha: aspecto del mapa en la etapa final del desarrollo. Fuente propia.

5.2.2.2 Historia y personajes

La historia perteneciente al arco del Reino Dulce fue originalmente escrita por Eva Quintela Balóira. Desde el departamento de diseño se sintetizó la historia original y se guionizó para ser adaptada al formato de videojuego. Dada la naturaleza cómica y desenfadada del juego se trató de aportar toques humorísticos a la historia, así como diversas referencias culturales contemporáneas que el público objetivo pudiese comprender, dichas referencias varían según el idioma elegido para jugar.

Para guionizar la historia se tomaron como referencia aventuras clásicas como "*Banjo-Tooie*" (*Banjo-Tooie*, 2000). En este tipo de videojuegos se acostumbraba a introducir una gran cantidad de humor en la historia, además de crear personajes fácilmente recordables por sus marcadas y únicas personalidades.

Siguiendo el esquema (Fig. 10) realizado en una etapa temprana del desarrollo para ordenar los sucesos de la historia y crear las primeras aproximaciones del mapa del juego, se identificaron seis sucesos principales de la historia:

1. Tofi se encuentra con la situación catastrófica: esta parte introductoria permite explicar el contexto al jugador, ya que la protagonista tampoco sabe nada al respecto. De esta forma tanto el jugador como el personaje principal conocen las circunstancias de la historia al mismo tiempo. Esto ayuda a que el usuario se identifique mejor con Tofi, además de permitir introducir tutoriales y diálogos explicativos de forma más coherente.
2. Pueblo Pinha es amenazado: se trata del primer gran obstáculo del juego, en él el jugador podrá familiarizarse con el conflicto y el estilo de combate.

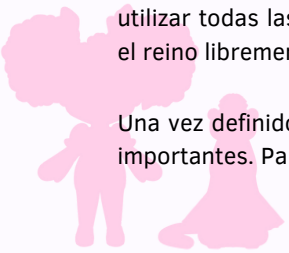
3. Encuentro con Ferri: este personaje se convertirá en el compañero fiel de Tofi. Él explica la situación del rey, lo que se convertirá en el objetivo principal, y es quien sugiere el camino a seguir hacia Pueblo Sandía, que da un objetivo inmediato al jugador.

4. El libro de los ayudantes: esta breve misión marca el inicio de las misiones secundarias para el jugador. Al conseguir la capacidad de crear a los ayudantes, se le brinda la oportunidad al jugador de continuar con la historia principal o desviarse y explorar para conseguir estos elementos opcionales.

5. Pueblo Sandía: este gran segundo obstáculo está pensado para que el jugador muestre lo aprendido en Pueblo Pinha y la exploración que realizase. Similar a lo sucedido en Pueblo Pinha, los habitantes están en peligro, pero en Pueblo Sandía se encuentra una amenaza mayor, el primer jefe del juego. Este jefe está pensado para ser una prueba que permita al jugador saber si sus habilidades de combate son lo suficientemente buenas como para vencer al jefe final. Una vez vencido, el jugador ya cuenta con todos los ataques disponibles y lo necesita superar el combate final, lo que le permite explorar el mapa si así lo desea para prepararse para la gran batalla.

6. Castillo: se trata del obstáculo final, aquí el combate contra Cucumis está pensado para ser un reto que haga al jugador utilizar todas las técnicas y estrategias que haya aprendido. Una vez finalizado el combate, el jugador es libre de explorar el reino libremente.

Una vez definidos los puntos clave de la historia se comenzó a crear las diferentes personalidades de los personajes más importantes. Para más profundidad acerca de este tema, consultar el anexo 03.



Tofi

Al ser Tofi la protagonista se quería evitar caer en clichés demasiado usados, como el héroe altruista que ayuda por su propia moral impecable. Durante su travesía, el personaje de Tofi se guía por un deseo egoísta, resolviendo los conflictos que se le presentan para poder volver a su vida despreocupada.

Sin embargo, conforme se enfrenta a los diversos conflictos que se le presentan, Tofi madura y comienza a interesarse cada vez más en ser una heroína al desarrollarse su sentido del deber.

Ferri

Ferri fue ideado para ser el compañero de la heroína, por ello su personalidad debía ser graciosa para poder ser un alivio cómico. Tofi y Ferri son parecidos en cuanto a sus deseos egoístas, pero se diferencia de Tofi por su cobardía. Ferri acostumbra a protestar fácilmente y a menudo expresa su cansancio por tener que ayudar a otras personas.

Tofi y Ferri se complementan el uno al otro. La cobardía de Ferri hace que Tofi deba tener valor para ayudar a los demás, ya que nadie más lo hará. A su vez, esta determinación de ayudar que Tofi desarrolla a lo largo de la trama inspira a Ferri, que se emociona viendo cómo Tofi se convierte en una heroína.



Lolo

En un principio se ideó para ser un personaje amable que ayuda a Tofi de forma incondicional. Aunque esta personalidad encajaba con su papel en la historia, se decidió dotarlo de más profundidad.

Al tener Lolo los dos negocios más importantes del reino, se exploró la posibilidad de que fuese un poco avaro, aunque buena persona en el fondo. Esto hace que Lolo resulte mucho más memorable, ya que es un personaje que crea situaciones divertidas por culpa de su obsesión con el dinero. Esta obsesión por el dinero ayudó también a dotar de más coherencia el altercado ocurrido en Pueblo Sandía, provocado porque Lolo decidió aprovechar ingredientes contaminados para ahorrar.

Regalino

Regalino cumple el papel de guía espiritual. Es quien explica a Tofi las circunstancias del conflicto principal y quien le pide que haga algo al respecto. Regalino es calmado y tiene un fuerte sentido de la justicia.

Cucumis

Este personaje fue creado para ser el antagonista principal del Reino, aunque no es un villano. Se trata de un rey justo y amable que fue corrompido por el poder del sabor salado. Durante el enfrentamiento contra él, está completamente fuera de sí y no es consciente de sus acciones. Dado que es el soberano del Reino Dulce, Cucumis tiene una personalidad agradable y suave.

El guion fue primeramente realizado en inglés, y traducido más adelante al gallego y al castellano.

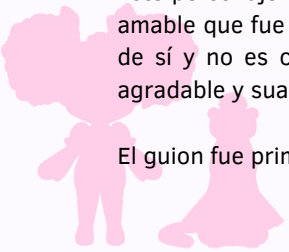




Fig. 15 Diálogo en diferentes idiomas. A la izquierda, en español. A la derecha en gallego. Ambos transmiten la misma información pero cuentan con referencias cómicas diferentes dependiendo de la cultura. Fuente propia.

5.2.2.3 Sonido

El departamento de diseño se encargó de la creación de una banda sonora a partir de música libre de derechos. Del mismo modo también recopiló todos los efectos de sonido del juego.

En una etapa temprana del desarrollo se barajó la posibilidad de contar con una banda sonora original compuesta por el equipo, pero esta idea fue descartada ya que suponía un gasto excesivo de recursos. Por este motivo se optó por recurrir a la construcción de una banda sonora formada por diferentes melodías libres de derechos de autor.

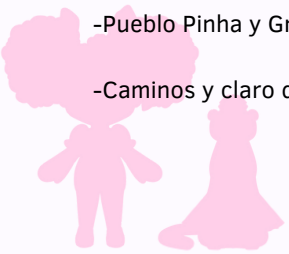
El procedimiento a la hora de recopilar la banda sonora fue identificar las personalidades de las diferentes zonas y realizar una búsqueda de melodías que resultasen coherentes entre sí.

Se establecieron las siguientes personalidades a las diferentes partes del proyecto:

-Menús: melodía alegre pero que no resultase demasiado infantil, debía ser sencilla y reconocible.

-Pueblo Pinha y Granjas: melodía tranquila, debía inspirar un ambiente familiar y agrario.

-Caminos y claro del bosque: debía recordar a un entorno natural e invitar a la exploración.



- Castillo exterior: la melodía debía hacer entender al jugador la presencia del monarca en el área, debe ser regia.
- Castillo interior: debía inspirar un ambiente real, pero también resultar misteriosa dada su situación de zona de jefe final.
- Bosque profundo: se trata de una zona misteriosa y oscura, por lo que la melodía debía reflejar este ambiente.
- Pueblo Sandía: esta zona destaca por ser más urbanita, por lo que la melodía debía dotarla de modernidad.
- Batalla común: la melodía debía ser energética, de forma que invitase al jugador a vencer a los enemigos.
- Batalla contra el jefe intermedio: la melodía debía hacer comprender al jugador que no estaba ante un enemigo común. Debía encajar con el caos que se está viviendo en la historia en ese momento.
- Batalla contra el jefe final: esta melodía debía reflejar seriedad y poder.



En cuanto a los efectos de sonido, se identificaron aquellos demasiado complejos o específicos como para ser encontrados en bancos de audio. Esto supuso que una gran cantidad de sonidos fuesen grabados por el departamento.

A la hora de grabar los diferentes efectos se buscaba alejarse del realismo y recurrir a efectos de un estilo *cartoon*, que encajasen con el resto de la estética del juego.

Tanto los sonidos grabados como la gran mayoría de los encontrados en bancos de audio fueron editados utilizando el programa *Audacity*. En este programa todos los audios debían ser pasados por un filtro de ruido ambiente, para que el sonido tuviese la mejor calidad posible.

A la gran mayoría de audios se les añadió un efecto de reverberación para evitar que el sonido resultase demasiado duro. Otras de las modificaciones realizadas fueron variaciones en el ritmo, cambios en el tono o mezclas de diversos audios.

Desde el equipo de diseño también se realizó el sonido de la cinemática inicial. Para este propósito se empleó *Audacity* para el tratamiento de los audios y *DaVinci Resolve* para el montaje de los sonidos acorde a la imagen (Fig. 16).

Todos los artistas cuya propiedad intelectual haya sido empleada en este proyecto se encuentran debidamente acreditados en el anexo 03.



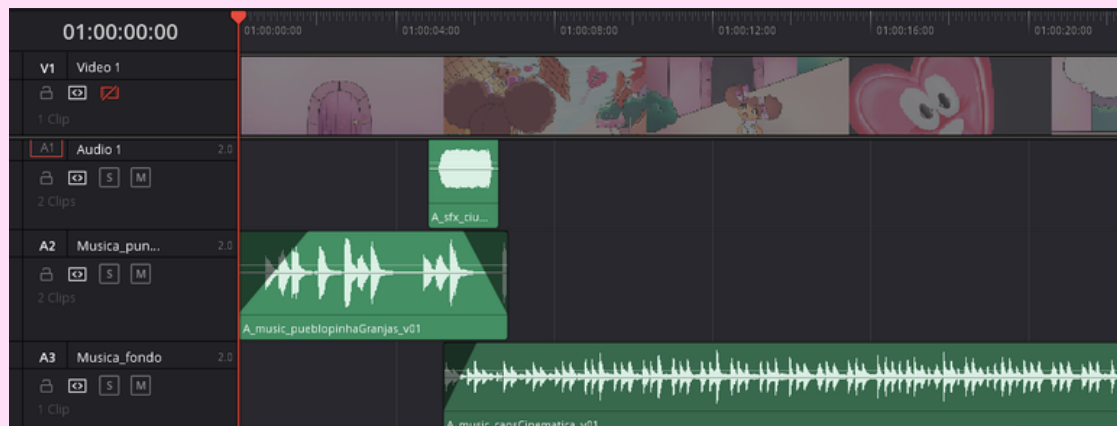


Fig. 16 Montaje de audio de la cinemática inicial en *DaVinci Resolve*. Fuente propia.

5.2.3. Conclusión individual

Balancear el juego de forma que pueda ser disfrutado por usuarios con diferente nivel de habilidad abre el producto a un mayor público además de hacerlo inclusivo para todo tipo de personas. Esto hace que el producto sea más accesible y resulta un punto fuerte.

A pesar de ello, balancearlo resultó complejo, puesto que había una gran cantidad de variables a considerar. Sin las opiniones de los *testers* y la gran cantidad de iteraciones realizadas no habría sido posible lograrlo.

Uno de los mayores temores que había al crear el mapa era que resultase pesado moverse por él. Al haber realizado un estudio previo de las zonas necesarias y el orden en el que deberían visitarse cada una para cumplir las misiones principales, fue más sencillo organizar el mapa de manera que la exploración no obligase al jugador a dar demasiadas vueltas.

En cuanto a la historia y los personajes, se invirtió una gran cantidad de tiempo en traducir el guion y los diversos textos dentro del juego, aunque esto supuso que el producto final fuese accesible a un mayor público. Al dotar de personalidades más marcadas a los personajes que toman parte en las misiones principales, se consiguió que resultasen más memorables y ayudó a dotar al juego de una fuerte personalidad.

La grabación y edición de los efectos de sonido ralentizaron la finalización de las tareas de audio. Sin embargo si se hubiese optado por usar más efectos pertenecientes a bancos, una gran cantidad de elementos contarían con efectos sonoros que no encajarían correctamente, por lo que se considera que esta inversión de tiempo fue acertada.

5.3. Departamento de arte

5.3.1 Objetivos propuestos

Los objetivos principales para el departamento de diseño se centraron en la creación y elaboración de un estilo reconocible y único, así como de su correcta interpretación por parte de los usuarios.

Para lograr dichos objetivos, se propuso la búsqueda de un estilo sencillo y llamativo que uniese los elementos 3D y 2D del juego de forma visual, así como generar diseños de personajes variados en formas y tamaños y que fuesen reconocibles a simple vista. Destacando también el planteamiento de diversas animaciones que le dan al usuario una retroalimentación visual clara de lo que sucede durante el juego, al igual que de una cinemática introductoria al universo del juego que pone al usuario en situación.



5.3.2 Proceso

5.3.2.1 Estudio previo

Después de haber realizado diversas pruebas de estilo y buscado referencias (Fig. 17) en otros juegos los cuales contaban con un acabado visual similar al que el equipo tenía como objetivo, como pueden ser (*Paper Mario: Sticker Star*, 2012), (*Octopath Traveler*, 2018) o (*Bear and Breakfast*, 2022), se optó por apostar por un estilo 2.5D. Con el conocimiento adquirido de previos proyectos llevados a cabo durante la carrera, como "Proyecto de videojuego", "Animación 2D" y "Dirección y realización", se optó por acuerdo de todas las integrantes por un estilo visual *cartoon* sencillo, esto para minimizar el contraste entre los objetos tridimensionales del entorno frente a los *sprites* y animaciones de personajes en 2D.

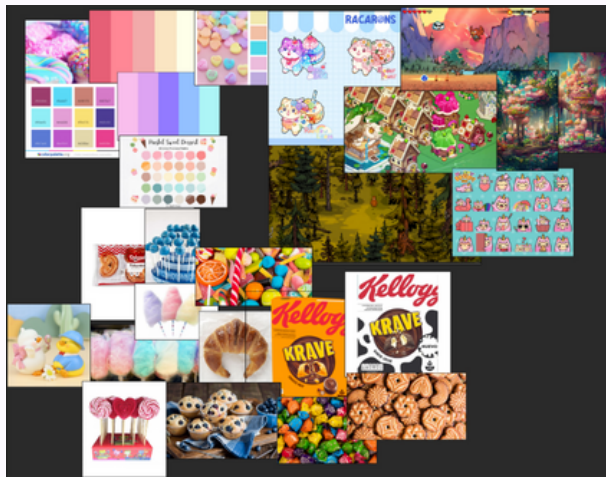


Fig. 17 Tablón de referencias de arte y estilo.
Fuente propia referencia de autorx et al.

En cuanto a la paleta de color general del juego, se realizó un estudio de qué colores podrían funcionar y cuáles no, así como diversas pruebas de colores sobre personajes y otros elementos. Resultando finalmente en una paleta de colores suaves con saturación media, principalmente con tonos fríos, destacando entre estos especialmente los tonos rosados. Esto para mantener la sensación de un mundo dulce y alegre, a la par que tranquilo.

5.3.2.2 *Concept*

En cuanto a los bocetos previos del juego, para la totalidad de los personajes de este se realizó un estudio de siluetas, formas y tamaños, tal como se nos había instruido en la asignatura de dibujo anatómico, comparándolos principalmente con la protagonista. Se optaron por siluetas principalmente redondeadas, ya que se buscaba esa sensación de amabilidad y tranquilidad, a la vez que dotaba a los personajes de un aspecto más adorable. Una vez se conseguía una silueta adecuada, se procedía al primer boceto de cada personaje y su paleta de color (Fig. 18 y 19). Todo esto siempre se consultaba con el resto de las integrantes del equipo en caso de cualquier posible ajuste, ya fuese de color o de diseño. En cuanto a aquellos personajes que se mantuvieron en la historia, se mantuvo como base su concept original, pero recibieron pequeños ajustes de silueta o detalles, así como de paletas de color, todo con la intención de que encajasen con el resto del nuevo set de personajes. Todos estos diseños se pueden ver en el anexo 02.



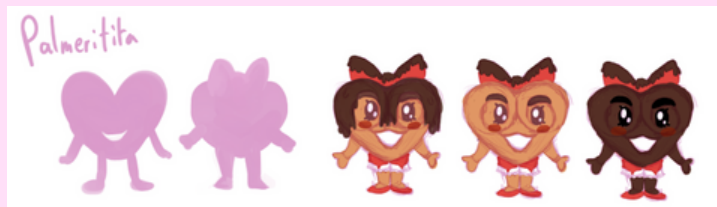


Fig. 18 Diseño y pruebas de color de Palmeritita. Fuente propia.



Fig. 19 Diseño y pruebas de color del pato Lolo. Fuente propia.

Aunque esta fue una regla básica para todos los personajes, uno de estos la elude. Cantino, el cual desde el inicio se tenía planteado que fuese un personaje completamente cubierto y misterioso, por lo que usa colores más oscuros y apagados a diferencia del resto de los personajes.

5.3.2.3 Sprites

Una vez los bocetos y paletas de color de los personajes eran aprobados por el resto del equipo, se procedió a la elaboración de los *sprites*.

Para mantener correctos los tamaños de todos los personajes de la misma forma en la que se habían diseñado, se creó un lienzo de un tamaño específico para que todos los personajes tuviesen el tamaño relativo que se había planteado anteriormente (Fig. 20). Sólo aquellos personajes que son de tamaño más grande, como Coruasana o Cucumis, rompen esta regla del lienzo, ya que se necesitaba uno mucho más grande debido a su ancho y alto con respecto a la protagonista.

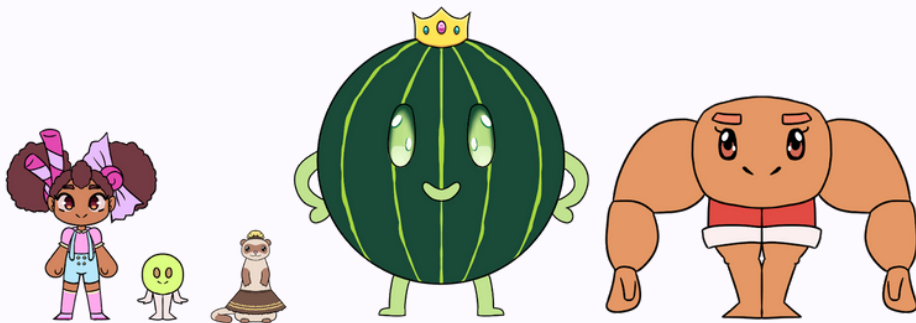


Fig. 20 Referencia de escalas de personajes. Fuente propia.

Para que todos los diseños tuviesen un mismo estilo, se usó un mismo pincel sin textura para el *lineart*, con el objetivo de que este tuviese un trazo lo más limpio posible. Para evitar variaciones de grosor, se utilizó un grosor mínimo y máximo de pincel con poco margen entre ellos, así todos los personajes cuentan con un grosor de línea lo más similar posible, manteniendo la coherencia visual entre ellos.

Respecto a los *NPCs* que se encuentran por el mapa para llenar el mundo, se crearon dos tipos de *sprites* para ellos: unos en los que se muestran asustados, y otros alegres, con el objetivo de que los personajes demostrasen diferentes expresiones a medida que la historia avanza y cambia. A la vez, para otros personajes relevantes de la historia, ya que otorgan misiones o diálogos importantes, se crearon *sprites* de solo sus cabezas para que fuesen fácilmente reconocibles desde el minimapa del *HUD* del juego.

Además de los *sprites* que se encuentran en la aventura y mapa del juego, también contamos con los *sprites* de diálogo. Estos están planteados como diseños más elaborados de los personajes, con una línea el doble de gruesa que recorre su silueta. Su principal funcionalidad es mostrar las expresiones de los personajes a medida que transcurren los diferentes diálogos. Para esto se siguió el mismo procedimiento que con los anteriores *sprites*, pero esta vez añadiendo detalles de luces y sombras, al igual que eliminando cualquier facción en sus rostros (Fig. 21), generando diferentes *sprites* intercambiables en las caras de los personajes, ya fueran ojos, cejas o bocas.

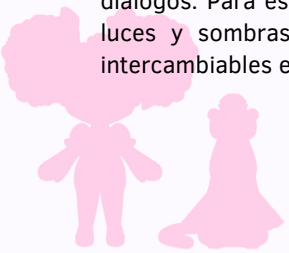




Fig. 21 *Puppet* y elementos de *sprites* de animación. Fuente propia.

5.3.2.4 Animaciones

Para la creación de los *sprites* de animación, se dio uso especial de lo aprendido en asignaturas como "Animación 1" y "Animación 2D", así como de los 12 principios de la animación. Muchas de las animaciones se han intentado exagerar, para darle ese toque *cartoon* de animación y que así coincidiese con el estilo de dibujo. Primero se crearon unas listas con todas las animaciones de cada personaje (anexo 09). Se decidió que, al ser animaciones pequeñas, se harían de forma manual fotograma a fotograma cada uno de éstos. Una vez se tenían claras las animaciones, se daba paso a una primera instancia, donde se realizaban los bocetos de los fotogramas clave, que daban la intención general de la animación. Estos fotogramas clave se consultaban con el equipo y una vez se les otorgaba el visto bueno, se daba paso a un boceto de animación más refinado con todos los fotogramas necesarios.



Para las animaciones de los diferentes *sprites* del juego, se decidió que se harían fotograma a fotograma dibujados en el programa Clip Studio Paint. Se consultaban con el resto del equipo y, una vez aprobado el boceto de todos los fotogramas, se pasaba al entintado de estos y se rellenaba la silueta de un color plano, como en los casos anteriores para evitar transparencias, a los que posteriormente se añadían los colores base de cada personaje (Fig. 22).



Fig. 22 Proceso de creación de *sprite* de Tofi. Fuente propia.

Cabe destacar que, al permanecer la mayoría de animaciones un corto período de tiempo en pantalla, se buscó crear movimientos rápidos a través de efectos, principalmente líneas de movimiento y deformaciones rápidas (Fig. 23). Aunque la mayoría de estos utilizan colores planos, para ayudar con esta sensación de movimiento, en algunos de ellos se llegaron a usar capas de efectos, como añadir brillo o degradados, para destacar aún más dicha acción.



Fig. 23 Sprites de efectos de movimiento. Fuente propia.

En el caso de animaciones como las de herido o aturdido, es donde se hizo mayor uso de estas capas de efectos. Con respecto a las animaciones de herido, tan solo se empleaba un fotograma del personaje con una expresión de dolor, pero intercalando entre colores blanco y rojo a través de capas de superposición, multiplicación y añadir brillo, para dar la sensación de golpe y daño. Por otra parte, en las de aturdido se usaba un tono azul suave en una capa de superposición para aclarar los colores base del personaje, en este caso la acción principal era la de los ojos y las pequeñas estrellas de mareo, para dar el mayor *feedback* visual posible al jugador.

Con respecto a la protagonista, las acciones más destacadas son las de batalla de minijuegos, contando con animaciones en caso de que falle el ataque, o lo haga de forma diferente al perfecto (Fig. 24). Para situar, por ejemplo en el ataque de horneado de pastel, si el ataque sale bien, el pastel estará cocinado, si no lo hace perfecto, el color de este pastel es diferente, en este caso fue un simple cambio de los colores base del *asset*. Por el contrario si el jugador hace mal el ataque, para aportar mayor *feedback* visual, en este caso el pastel se quema y la masa estalla en la cara de la protagonista. También para estos mismos minijuegos, como el "Batidor de merengue" o el "Bananarang", se crearon pequeñas animaciones de estos objetos arrojados (Fig. 25), para que el jugador pudiese ver cómo golpean al enemigo.





Fig. 24 Diferentes resultados de *sprites* del "Horneado de pastel". Fuente propia.



Fig. 25 Pasteles arrojadizos. Fuente propia.

Algo que ayudó a tener también una consistencia en los personajes, fue el duplicar las capas de *lineart* de sus cabezas. Estas se deformaban a través de la herramienta licuar, y sufrían modificaciones en las expresiones, pero esta base común aseguraba que aunque las animaciones fuesen rápidas, siempre se tuviese un punto poco cambiante en el que fijarse y mantener de referencia.

El mayor objetivo que se buscaba durante este proceso era asegurarse que el *feedback* visual para el jugador fuese correcto, lo que supuso que algunas animaciones que en principio parecían adecuadas, tras probarlas los *testers* dicesen lugar a confusión. Como es el caso de la animación de cavar, que en una primera instancia estaba correcta, pero resultaba confusa dentro del juego ya que parecía que estaba tapando el agujero en lugar de cavarlo, por lo que se volvió a la mesa de ideas para ajustarla. Así como *sprites* que se modificaron, en especial de Tofi, que se añadió una cuchara a su diseño y animaciones que no estaban planteadas anteriormente, y que sufrió modificaciones en su versión del mundo de juego, eliminando la nariz de personaje en este entorno para simplificar su diseño, ya que apenas se podía apreciar.



5.3.2.5 Cinemática

Siguiendo las bases que se nos impartieron en asignaturas como “Lenguaje y narrativa gráficos y audiovisuales” y “Dirección y realización”, primero siguiendo el guion de la historia se creó un *storyboard* (Fig. 26), pensando principalmente en la continuidad y los planos, de forma que aquello que estaba escrito en el guion se interpretase correctamente. Este se realizó ya directamente en *Blender*, con la intención de poder convertirlo rápidamente en animática, así se podría ver qué tan bien se entendía cada plano y cómo funcionaba en conjunto la escena, además de poder ajustar unos tiempos en pantalla de referencia de cada plano.

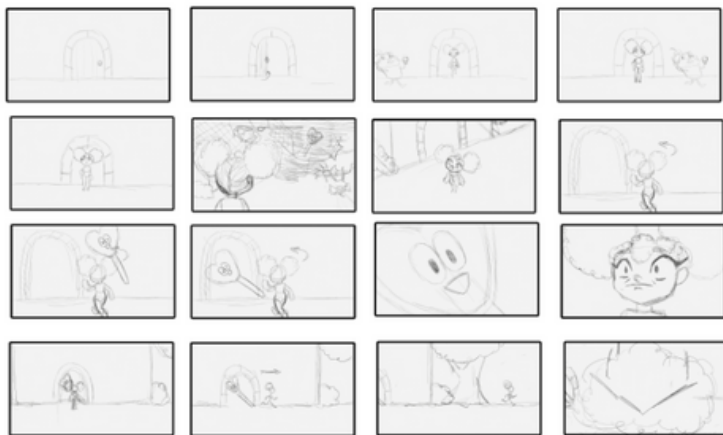


Fig. 26 *Storyboard* de la cinemática inicial. Fuente propia.

La parte más complicada fue encontrar un estilo de animación para estas cinemáticas. Teniendo de referencia algunas cinemáticas de estilo recorte y de deformación de imágenes de juegos que las trataron de forma similar, como es el caso de (*Identity V*, 2018) o (*Cookie Run Kingdom*, 2021) (enlaces de cinemáticas de referencia: [Identity V](#), [Cookie Run Kingdom](#)). Se realizaron diversas pruebas (Fig. 27), así como diversas investigaciones de cómo animar de esta forma en *Blender*, pero ninguno de los resultados iniciales se acomodaba a lo que se buscaba y pretendía, hasta que finalmente tras muchas pruebas, se llegó al estilo intencionado. La idea principal era una animación de estilo recorte por deformación para darle una apariencia de imágenes de un libro en movimiento, y muchos de los métodos que se sugerían eran a través de *rigging*, que no terminaba de ajustarse al estilo de animación por deformación que se buscaba. Durante una de estas investigaciones finalmente se encontró exactamente lo que se pretendía: animación por deformación de imágenes. Por lo que una vez con el estilo indicado, se comenzaron a preparar todas las imágenes necesarias en un programa diferente a *Blender* por comodidad personal, en este caso *Clip Studio Paint*, para más tarde añadirlas a *Blender* y así poder animarlas.

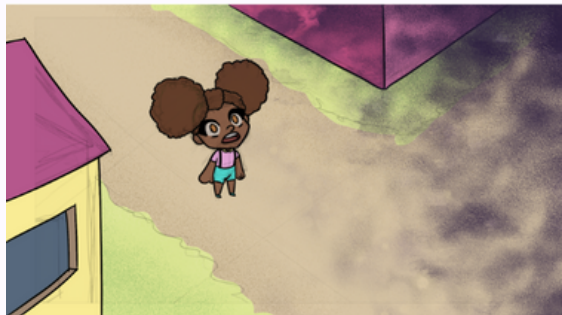


Fig. 27 Primera prueba de estilo de cinemática en *Blender*.
Fuente propia.

En este caso, ya que la animación era por imágenes, sí se dieron más detalles de luces y sombras, sobre todo para generar la tensión requerida por la escena, además de dar ese acabado de ilustración de un libro mediante el uso de diferentes texturas de coloreado.

En este proceso se siguió un método similar a la elaboración de los *sprites*: se utilizó un lienzo del tamaño de la resolución máxima del juego, con el *storyboard* de referencia para las imágenes, se pasaba a la creación de *lineart*, en esta ocasión usando diversas capas, tantas como objetos hubiese. Nuevamente se rellenaron las siluetas con un color plano para evitar transparencias, y se dieron los colores base. Cuando la escena estaba completa, se empezaba por el fondo a iluminar y dar sombras con diferentes capas de efecto, en especial multiplicación y añadir brillo. Una vez el fondo de la escena tenía las luces y sombras principales para la iluminación general de la escena, se pasaba a hacer lo mismo con el resto de elementos (Fig. 28). Se siguió esta regla para todas las escenas e imágenes para que fuese el fondo el que marcara la luz y su dirección y evitar posibles descuidos de iluminación.

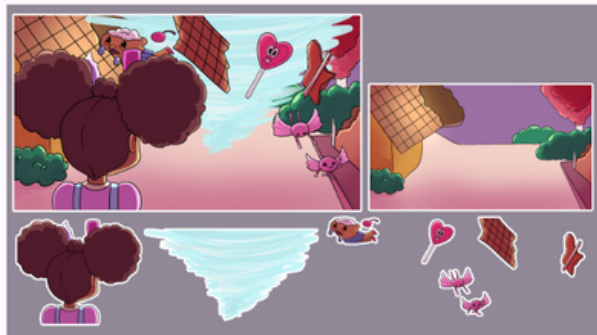


Fig. 28 Escena de la cinemática y sus respectivos elementos. Fuente propia.

Ya con las imágenes preparadas y separadas, todas estas se añadieron en *Blender*, separadas en colecciones según el plano para mantener un orden entre ellas, para así empezar a animar. Cabe destacar que para que en el *render* del vídeo se viesen correctamente, todas las imágenes debían tener un *shading* plano con el parámetro *emmit*, ya que de no tenerlo la imagen saldría oscurecida o en negro, y aplicar luces a la escena distorsionaba los colores de la imagen.

Se colocó la cámara en la escena de modo que coincidiese con el tamaño de la imagen del fondo, ya que esta es la que ocupaba el tamaño total de la pantalla (Fig 29).

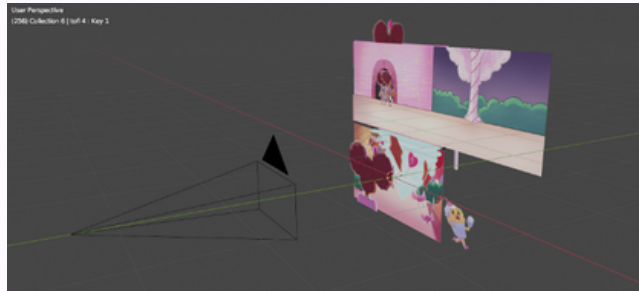


Fig. 29 Colocación de cámara e imágenes en la escena de *Blender*. Fuente propia.

Para animar las imágenes, lo primero que se hizo fue subdividir cada una de ellas individualmente, exceptuando aquellos fondos que no requerían deformación, para darles más geometría y que a la hora de deformar tuviesen más fluidez y movilidad. Posteriormente se crearon *key shapes* (Fig. 30). De estos *key shapes*, el base se mantenía la imagen sin deformar, mientras que se creaban nuevos *key shapes* para las deformaciones, dependiendo de cuántos necesitase en total la imagen. Siguiendo el *storyboard* que se había creado anteriormente como referencia, se hicieron ajustes de *timing* con las nuevas imágenes.

Es importante mencionar también que la última escena de la cinemática principal se creó de forma que ocupase casi el doble de ancho de la pantalla. A la hora de preparar la imagen, se planteó así para que la cámara pudiese seguir a la protagonista a lo largo de la escena de huida. En esta misma escena, en la caída del árbol, se prepararon dos imágenes intercambiables para dar la sensación del movimiento descendente de este.

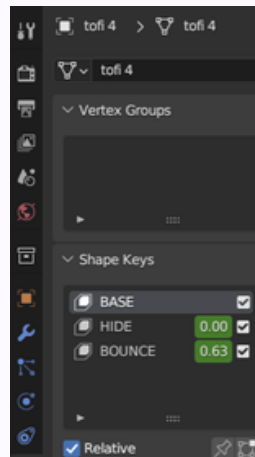


Fig. 30 *Key shapes* creados para la animación de una imagen de Tofi. Fuente propia.

Todas las imágenes contaban al menos con dos *key shapes*: el base, donde se veían en cámara, y el de desaparición, para dejar paso a las nuevas imágenes. A partir de ahí se emplearon todo tipo de deformaciones en las imágenes a través de sus vértices. Los más importantes siendo desplazamientos y escalados de estos mismos, aunque también se llegó a emplear en alguna ocasión la rotación.

En el caso de escenas con varias imágenes, como puede ser la escena del tornado, se pasó a una vista 3D para colocar las imágenes de forma que se evitase un posible solapamiento, y que estuviesen en el orden de visualización correcto, también facilitando la animación de estas mismas.



5.3.2.6 Modelado

El primer paso en el desarrollo del modelado 3D del juego fue el *concept art*, que se realizó siguiendo las instrucciones de la líder del departamento de arte.

Para los diferentes modelos se tomaron como inspiración alimentos dulces del mundo real, de esta manera conseguiría que encajasen con la temática general del reino.

El *concept art* fue realizado en la herramienta de dibujo *Procreate*, y se empleó un estilo *cartoon* estilizado.

Todos los bocetos realizados están recogidos en el anexo 02.

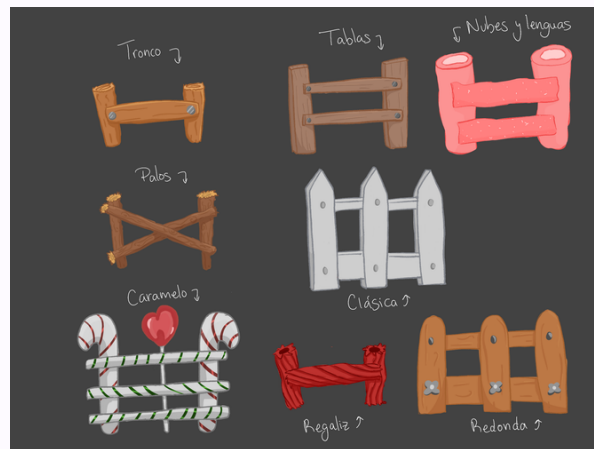


Fig. 31 *Concept art* de vallas. Fuente propia.

Una vez el *concept art* fue aprobado por el departamento, se continuó en la fase de modelado. El modelado 3D se realizó en *Blender*. El objetivo estético se fijó en modelado de baja densidad de topología, con acabados suaves. Al perseguir una estética *cartoon* se acordó no utilizar texturas demasiado detalladas, en su lugar se emplearían colores planos y vibrantes.

Todos los modelos 3D realizados para este proyecto están descritos en el anexo 02.

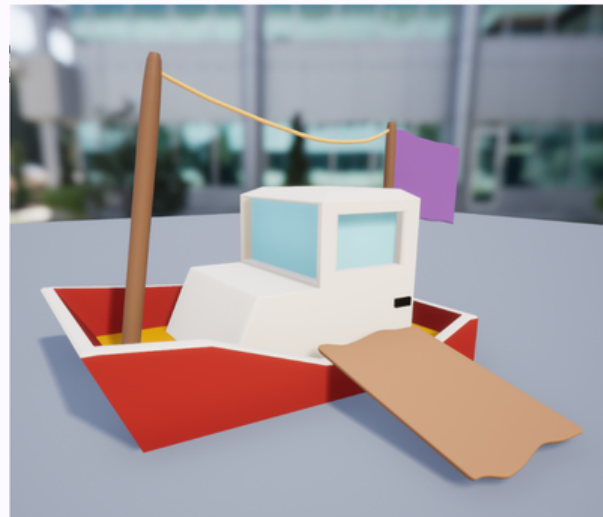


Fig. 32 Modelado 3D del barco de Cantino. Fuente propia.

Para el modelado de las viviendas de los pueblos, se acordó modelar una variedad de piezas por separado. Esto permitiría crear una amplia variedad de viviendas sin tener que ser modeladas de manera individual (Fig. 33).



Fig. 33 Casa modular perteneciente a Pueblo Pinha. Fuente propia.

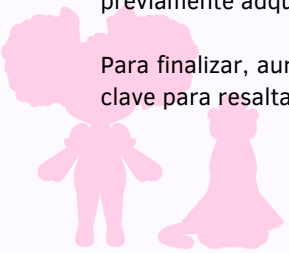
5.3.3 Conclusión individual

El departamento de arte sirvió para aportar el apartado visual al juego, de forma que refuerce la experiencia de juego y la retroalimentación para el usuario, tanto de las interfaces como de las diferentes mecánicas.

Lo más relevante de este departamento fue dotar a los diferentes *assets* del juego de carácter y personalidad, con diseños únicos a través de un estilo atractivo de colores vibrantes. Se notó especialmente la relevancia de una buena comunicación, tanto con el equipo como con las personas fuera de éste que estaban probando el juego, principalmente para detectar ciertos detalles o problemas que a la hora de preparar las imágenes y animar, como desarrollador y creador, no eres capaz de ver cómo una persona que no está dentro del proyecto lo interpreta.

También fue muy destacable la organización y flujo de trabajo que se tuvo que desarrollar en el departamento debido a la gran cantidad de *assets* que se necesitaron manejar, así como la investigación de nuevas técnicas al conocimiento previamente adquirido en la carrera para llegar al resultado que se buscaba.

Para finalizar, aunque este proyecto se centró más en las mecánicas y la historia, el peso del departamento de arte fue clave para resaltar todos estos aspectos y hacer que el juego fuese lo más atractivo y vistoso posible.



5.4. Departamento de interfaces

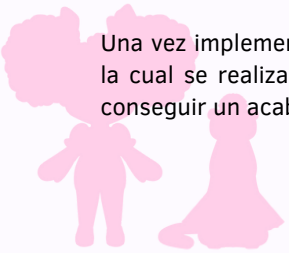
5.4.1 Objetivos propuestos

Los trabajos realizados dentro del departamento de interfaces se dividen en tres etapas del proyecto denominadas *Alpha*, *Beta* y *Golden*.

Para la versión *Alpha* se crearon las pantallas principales del proyecto, se implementaron los sistemas más importantes para el funcionamiento del juego como son el sistema de inventario, el sistema de diálogo, la creación de compañeros, el *HUD* de exploración y el *HUD* de batalla (conceptos explicados en el apartado 5.3.2.1 *Alpha*).

Durante la *Beta* se llevó a cabo la mayor parte del proceso, ya que se implementaron todos los menús y las funcionalidades restantes. La implementación incluye una creación simplificada de las pantallas del juego en el motor del proyecto y realizar el acabado estético final. Por otro lado, las funcionalidades principales que se implementaron en esta etapa fueron el sistema de ajustes, el sistema de traducción y el *mapping* de controles (conceptos explicados en el apartado 5.3.2.2 *Beta*).

Una vez implementada la estética final de los menús, el siguiente paso fue llegar a la versión *Golden* del proyecto, para la cual se realizaron los arreglos pertinentes y se implementaron elementos sugeridos por parte de los *testers*, para conseguir un acabado final lo más completo posible.



5.4.2 Proceso

Se realizó un desglose de las mecánicas y de las pantallas del juego, descritas en el Documento de mecánicas esenciales (desarrollado primeramente para la asignatura “Diseño de Niveles y Jugabilidad” y adaptado posteriormente para este proyecto, anexo 07). Junto al desglose se estableció una lista de prioridad de los elementos, con la que posteriormente el departamento de producción elaboró un calendario para establecer fechas y organizar al equipo (anexo 09). También se realizó un flujo de pantallas como referencia visual de las conexiones entre los distintos menús del juego (Fig. 34).

Se llevó a cabo una investigación que consistía en la recopilación de información sobre los distintos sistemas a implementar y, posteriormente, se llevó a cabo la búsqueda de soluciones a dudas que surgieron al plantear teóricamente las mecánicas a realizar. La búsqueda de información comenzó por los documentos y trabajos realizados de las asignaturas “Desarrollo de Videojuegos 1”, “Desarrollo de Videojuegos 2” y “Programación de Videojuegos”. Aunque también se consultaron diversos canales de *YouTube* de profesionales del sector (enlaces disponibles en el apartado 8.2.4 Bibliografía utilizada por Cristina Rodríguez Rodríguez) y el foro oficial de *Unreal Engine*.



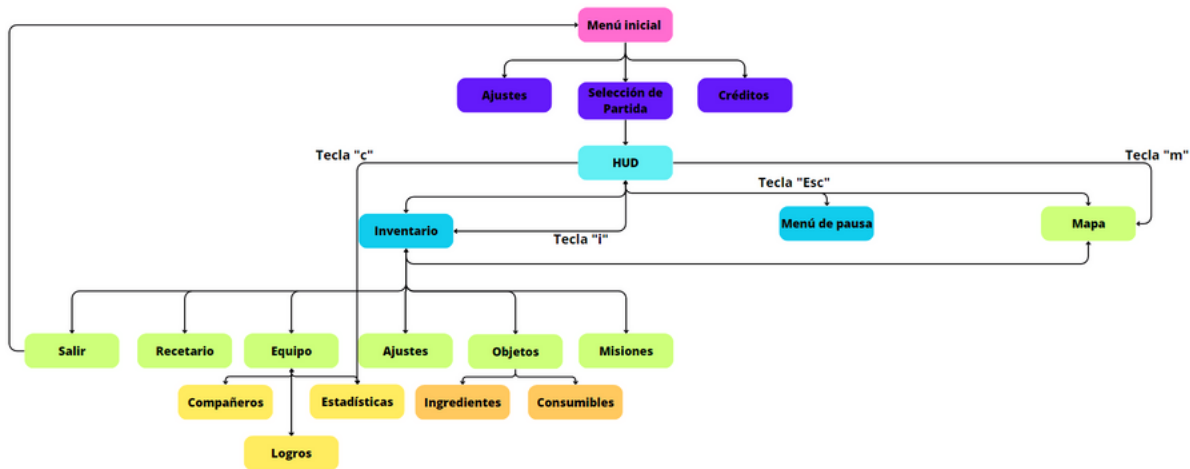


Fig. 34 Flujo de las pantallas de las que dispone el juego incluyendo los accesos directos por teclas. Fuente Propia.

Se realizó un análisis de las interfaces ya diseñadas con anterioridad para el trabajo final de la asignatura “Diseño narrativo y de interfaces” y se rediseñaron aquellas que el equipo desarrollador del proyecto consideraba que no encajaban finalmente con la esencia del juego. Para esta nueva versión, se realizó una búsqueda de distintos ejemplos de interfaces en diferentes juegos en la página web [Interface in Game](#). Para la elaboración de los nuevos diseños se utilizó la página web [Penpot](#).

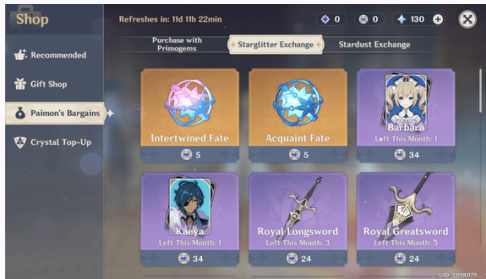


Fig. 35 Referencia del sistema de inventario.
(Genshin Impact, 2020)
Fuente [Interface in Game](#).



Fig. 37 Referencia de la subida de nivel.
(Pokémon Mystery Dungeon Rescue Team DX, 2020)
Fuente [Interface in Game](#).



Fig. 36 Referencia del sistema de diálogo.
(Hades, 2018)
Fuente [Interface in Game](#).

5.4.2.1 *Alpha*

El primer sistema que se desarrolló fue el sistema de inventario ya que se consideraba uno de los menús más importantes del juego. El proyecto se basa en la mecánica del combate por turnos y las estadísticas del jugador (vida, ataque, defensa y velocidad) son una parte esencial de ésta, por lo que necesidades básicas como poder recuperar vida o aumentar el ataque se solventan en este menú.

El menú cuenta con cinco espacios, los cuales se llenan a medida que el jugador recolecta objetos (anexo 03). El usuario puede utilizarlos si tiene al menos una unidad y si cumple los requisitos necesarios según el tipo: en el caso de querer usar un objeto curativo, necesita no tener la vida ya al máximo, y si es un objeto de aumento de ataque o defensa, debe no tener el efecto ya aplicado. Si por el contrario no puede consumir el objeto, se informará al jugador con un mensaje en pantalla describiendo el motivo.



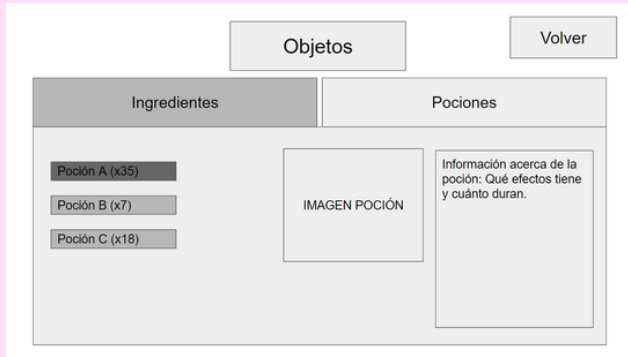


Fig. 38 Diseño original del menú de objetos.
Fuente propia.



Fig. 39 Rediseño del menú de objetos.
Fuente propia.

Otra de las mecánicas esenciales desarrolladas en esta etapa fue el sistema de diálogo, necesario para que el jugador pudiese descubrir la historia a través de sus conversaciones con los personajes. A la hora de crearlo fue prioritario diseñar un sistema que fuese compatible con la programación de las misiones, además de fácilmente reutilizable para las demás mecánicas. Para ello se programó un sistema basado en recorrer una lista de datos donde se podrían añadir las líneas de diálogo que se precisasen y enlazarlas al personaje que las diría. Luego se creó la interfaz del diálogo, añadiendo tanto la imagen del personaje que habla junto a su nombre (personalizable según la emoción) como la imagen del personaje que escucha, como puede verse en la Fig. 40. Se implementó una animación en el texto de la línea de diálogo para simular una conversación más realista y conseguir con ello una experiencia más inmersiva al crearse las palabras en el tiempo que se tardaría en decir las en voz alta.

La parte más tediosa de su desarrollo fue la de incorporar todas las frases que existían en el guion del juego, puesto que la única manera de darle esa información al motor era de forma manual, línea por línea.



Fig. 40 Sistema de diálogo. Fuente propia.

La creación de los compañeros es un sistema relacionado con la exploración del mapa, las misiones secundarias y el sistema de combate, los cuales son pilares del proyecto. Este sistema consiste en crear los “compañeros”, personajes que ofrecen mejoras al jugador durante la batalla. Para crearlos el jugador deberá conseguir ingredientes a través de las misiones secundarias que activan distintos *NPCs*. El jugador podrá crear un compañero en el momento que obtenga los ingredientes necesarios para ello.

Los ingredientes que corresponden a cada compañero se le muestran al jugador a través de su silueta, diferenciando los conseguidos de los no recolectados mediante el color, permitiéndole al jugador identificar de un vistazo cuántos le faltan (Fig. 41 y Fig. 42).

El juego cuenta con tres compañeros “Biais”, “Caralleta” y “Doffy” donde cada uno proporciona una habilidad diferente en batalla (habilidades descritas y detalladas en el apartado 7.2.1 Sistema de batalla del departamento de programación).

Una vez que los compañeros han sido creados, el jugador tendrá que equiparlos en el menú de estadísticas para conseguir las ventajas que le proporcionan. Una vez equipados, durante el combate serán visibles debajo del indicador de vida del jugador.





Fig. 41 Sistema de creación de compañeros.
Fuente propia.



Fig. 42 Sistema de creación de compañeros.
Fuente propia.

Por otro lado, se implementó el HUD de exploración (Fig. 43), el cual cuenta con un minimapa que sigue al jugador en todo momento y que permite visualizar los elementos importantes cercanos, y un botón para acceder al inventario. Se implementó también el HUD de batalla (Fig. 44) siguiendo el diseño establecido previamente. En este se añadieron diversos elementos para que el jugador supiese cuando podía atacar y se implementaron mensajes para indicar el número de daño recibido y si se había esquivado el ataque.



Fig. 43 HUD exploración.
Fuente propia.



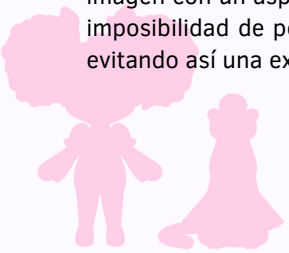
Fig. 44 HUD batalla.
Fuente propia.

5.4.2.2 Beta

En esta fase se realizó la mayor parte del trabajo necesario para completar el proyecto, añadiendo los elementos necesarios para obtener un producto completo y jugable. Se incluyeron todos los menús, elementos visuales y sistemas que faltaban, se añadieron las pantallas de carga, los tutoriales, el guardado de los objetos y de los ajustes y se implementaron las texturas finales de las interfaces.

Uno de los menús que se desarrolló fue el de los ajustes. Éste consta de ajustes de audio y de vídeo, pero además ofrece la oportunidad de personalizar la experiencia de juego y de hacerlo accesible a un público mayor, puesto que cuenta con la opción de *mapping* de controles de todas las acciones que suceden en el juego y un sistema de traducción de hasta tres idiomas diferentes. Este último suponía una prioridad para el equipo de desarrollo, puesto que se consideró una manera de abrirse a un público mayor.

El sistema de ajustes tuvo que ser modificado, puesto que la funcionalidad no era la esperada. Al botón se le asociaba una imagen con un aspecto muy diferente a un rectángulo (área de colisión que tienen todos los botones de *Unreal Engine*) y ante la imposibilidad de poder modificar el área de colisión para que encajase con la imagen utilizada, se optó por rediseñar el menú, evitando así una experiencia incómoda o frustrante (Fig. 45).



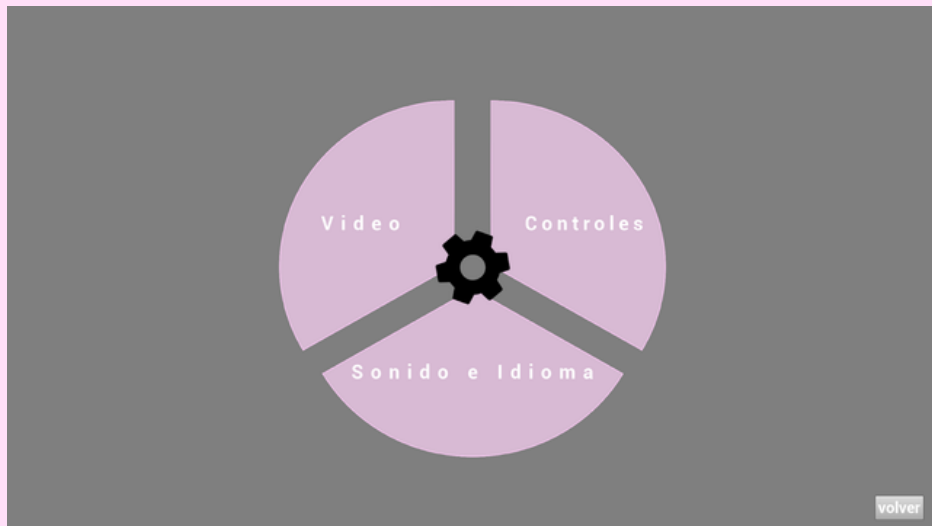


Fig. 45 La forma de los botones no encajaba con la colisión rectangular.
Fuente propia.

El *mapping* de controles consta de una serie de botones asociados cada uno a una acción y, cuando el jugador presiona una tecla nueva, el sistema programado registra la nueva tecla seleccionada y la guarda como nuevo control, permitiendo así que el jugador pueda jugar desde unos controles adaptados a sus preferencias.

El sistema de traducción, se llevó a cabo con funciones que ofrecía el propio motor del juego. El programador debía especificar a cuántos y a qué idiomas se quería traducir el videojuego, decantándose el equipo por el español, el gallego y el inglés. Posteriormente debía seleccionar las carpetas del proyecto en las que se encontrasen las variables que contenían la información que se debía traducir. Éstas consistían únicamente en aquellas mostradas visualmente en el producto jugable, puesto que traducir una variable errónea o innecesariamente podría dar lugar a problemas internos en las funcionalidades hasta el momento implementadas en el proyecto. Una vez seleccionadas las rutas pertinentes se realizaba un recuento de palabras y/o frases. Finalizado el recuento, todas las palabras aparecían en el “Panel de localización” (“*Localization Dashboard*” en *Unreal Engine*) y una vez ahí el equipo debía traducir a mano los valores correspondientes al idioma seleccionado. Una vez realizado el sistema de las traducciones, se le dio al jugador la posibilidad de jugar al videojuego en los idiomas español, gallego e inglés según su preferencia.





Fig. 46 Inventario en inglés.
Fuente propia.



Fig. 47 Inventario en español.
Fuente propia.

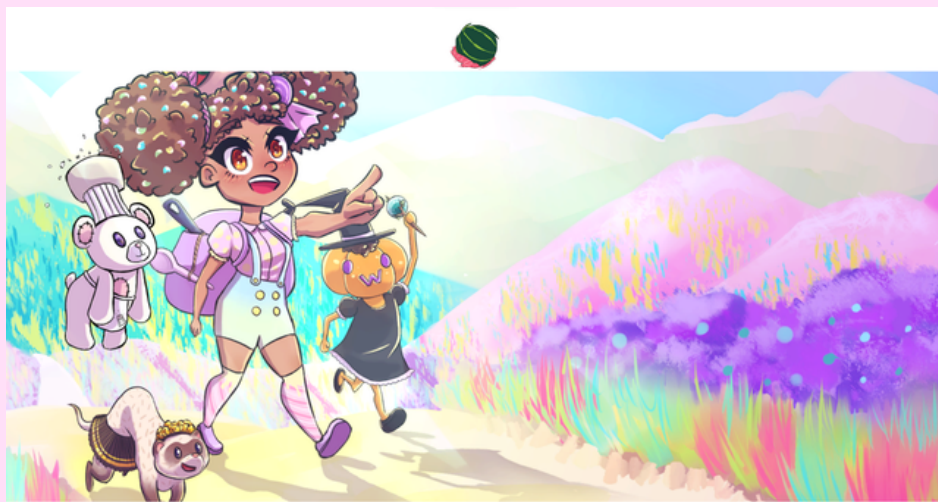


Fig. 48 Inventario en gallego.
Fuente propia.

Se implementaron los tutoriales, siendo el primero de ellos el tutorial de batalla. En él uno de los personajes llamado Regalino enseña al jugador cómo funciona el sistema de batalla y sus características. El usuario deberá seguir paso a paso las indicaciones o de lo contrario no avanzará. El segundo tutorial implementado fue el del inventario, en él se le enseñan los distintos menús que el jugador tiene disponibles al principio de la partida y sus funciones. Para la creación de los tutoriales se utilizó el sistema de diálogo implementado anteriormente y se adaptó para que no interfiriese con las distintas pantallas del juego.

También se crearon otros menús como el menú de inicio, el recetario, el mapa, los logros, el menú de misiones... (anexo 05). Para la realización de las pantallas de los diferentes menús se crearon las *blueprints* de Unreal Engine correspondientes, utilizando los elementos básicos que el propio motor proporciona pero personalizándolos y colocándolos según las propias características de cada menú. Éstas se realizaron tomando como referencia el diseño previo y prestando especial atención a la posición de las anclas de los elementos, ya que en el caso de realizarlo mal afectarían negativamente a la posición de los elementos al cambiar la resolución del videojuego.

Asimismo, se introdujeron las pantallas de carga entre la carga y descarga de niveles, para ello se utilizó un *plugin* de la plataforma "Epic Games" para Unreal Engine, denominado "Async Loading Screen". Este traía el funcionamiento básico, la única labor a realizar era personalizar el aspecto que se quería mostrar y la información que se quería transmitir. Las pantallas de carga (Fig. 49) cuentan con una secuencia de imágenes en la parte superior que le indica al usuario que el producto no se ha quedado congelado, que sigue en funcionamiento. En la parte central cuenta con una imagen relacionada con el videojuego y por último en la parte inferior, cuenta con unos consejos para el jugador acerca de la experiencia de juego.



Se alguén che da un ingrediente, gádao. Pode ser de utilidade!

Fig. 49 Pantallas de carga.
Fuente propia.

En esta versión también se implementó el acabado final, sustituyendo las interfaces simples por las texturas realizadas por el departamento de arte (cambio observable en las Fig. 50 y 51). Se realizó el trabajo pantalla por pantalla, incorporando las texturas finales y adaptando aquellas que fueran necesarias. Por otro lado también se realizó el cambio de la tipografía de todos los textos por la elegida por el equipo para el proyecto, “*Abalone Smile*”, y se creó un material para añadirle degradado a los títulos, donde se podían modificar los colores para que quedasen acorde a las texturas de las interfaces.



Fig. 50 Diseño sin texturas.
Fuente propia.

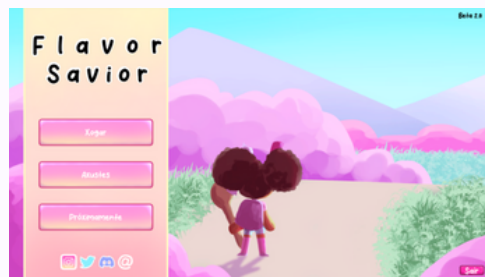


Fig. 51 Implementación del acabado final.
Fuente propia.

A mayores se personalizó la experiencia de cada botón de modo que el jugador tuviese siempre información sobre si tenía el cursor sobre un elemento interactuable o no y se añadieron animaciones para dotar a las interfaces de movimiento. En definitiva, se cuidó hasta el más mínimo detalle a la hora de conseguir el acabado final.

Se añadieron al sistema de guardado todos los objetos del inventario, compañeros, ingredientes, logros, etc. conseguidos por el jugador en su partida hasta el momento, ya que sino este perdería todo su progreso. Para ello se elaboró un sistema que consistía en añadir toda la información del jugador en un archivo de guardado externo al videojuego y éste se asociaba a la ranura de juego en la que el jugador hubiese empezado la partida. Cuando el jugador volviese a entrar en su partida, la información guardada serviría para asociar las imágenes de los objetos, la cantidad que se tenía, los nombres, las descripciones... toda la información que se necesitaba para que al iniciar el juego de nuevo éste continuase en las mismas condiciones.



Fig. 52 Si las ranuras cuentan con un objeto animado en ellas significa que hay una partida guardada.

Fuente propia.

También se realizó el guardado de los ajustes siguiendo el mismo sistema que el guardado del resto de elementos, la única diferencia es que los ajustes son los mismos para todos los usuarios, no dependen de la ranura seleccionada. Uno de los problemas encontrados en el proceso de guardado de los ajustes fue la opción de expandir la ventana del juego con el icono de la barra de *Windows*, ya que ésta permitía al jugador escalar la ventana momentáneamente, pero como en los ajustes el modo de la ventana era distinto, ésta se volvía a redimensionar, confundiendo así al jugador. Para poner solución se optó por bloquear el botón de maximizar de la pestaña de *Windows*, de manera que si el jugador quisiese jugar con una resolución distinta debería cambiar sus preferencias utilizando las opciones disponibles en el menú de ajustes.



5.4.2.2.1 Fondos

Para la creación de los fondos de las interfaces, se generaban imágenes del tamaño de resolución máximo del juego, al que se añadían posteriormente diferentes tipos de efectos en el programa de *Clip Studio Paint*. Para esto, se seleccionaba una paleta de colores, uno de estos se aplicaba al fondo, mientras que el otro se aplicaba a los efectos, como podrían ser degradados o tramas.

Se usaron diferentes tipos de tramas, principalmente líneas, puntos y cuadrados, para dar un acabado más alegre y texturizado a cada fondo (Fig. 53 y 54). También para facilitar que estos estuviesen correctamente simétricos y rectos, se utilizaron herramientas de forma con el mismo grosor de línea.



Fig. 53 Ejemplo de fondo. Fuente propia.



Fig. 54 Ejemplo de fondo. Fuente propia.

Para fondos de apartados comunes, se mantuvo la misma forma de fondo, pero se hicieron ajustes de colores para variación y *feedback* de cambio de pantalla para el jugador. La mayoría de fondos utilizan una paleta de colores análogos para más armonía y menor distracción del usuario, con la intención de que pueda mantener la atención en los aspectos importantes de la interfaz como son los botones e iconos.

Se preparó también una nueva imagen del mapa basada en la vista y cámara de *Unreal Engine*. Se utilizó una captura de dicha vista como plantilla y se procedió a crear límites más orgánicos, con bordes más redondeados, así como a marcar todos los caminos y edificios de forma clara y visible. Se añadieron líneas de grosor para delimitar los bordes y un fondo de color con trama para darle un mejor acabado (Fig. 55).

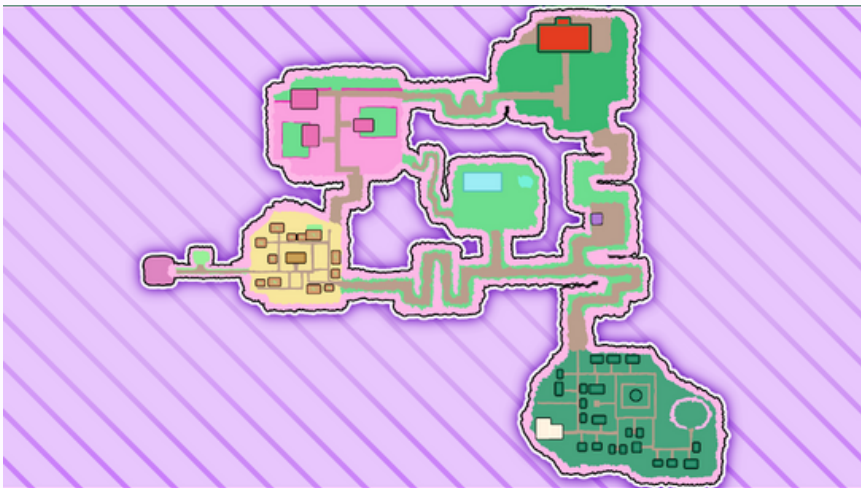


Fig. 55 Mapa del juego actualizado. Fuente propia.

5.4.2.2 Diseño de la portada

Para el diseño de la portada, se dio uso de los conocimientos adquiridos durante la asignatura de "Dibujo de entornos y arte de concepto", por lo que se empezó con unas miniaturas de siluetas para plantear la composición de la escena. Teniendo como referencia la interfaz ya diseñada como base, esta fue relativamente compleja, dado que la portada ocupa el doble de la pantalla, por lo que no podían quedar elementos centrados en la imagen, o se quedarían cortados. Finalmente se encontró una composición adecuada y agradable, teniendo a la protagonista a un lado de la pantalla y el castillo final en la otra mitad.

Con la miniatura seleccionada, se procedió a hacer un boceto más detallado con un sólo color y por diferentes capas de profundidad, con el objetivo de dividir los objetos en la escena según lo cercanos o lejanos que estén desde el ángulo de la cámara (Fig. 56). Aquellos objetos más cercanos emplean un color más oscuro, mientras que cuanto más se aleja del punto de vista, más suave y claro era el color.



Fig. 56 Boceto con formas de color de la portada. Fuente propia.

Una vez se llegó al resultado esperado, se añadieron los colores base, nuevamente por capas según la profundidad (Fig. 57). Para unificar todos los colores de la escena, se añadió una capa de efecto superior en un tono rosado, esto para que todos los colores mantengan un mismo tipo de color e iluminación general, así como unificar colores que podrían quedar disonantes entre sí. Posteriormente, a cada capa se le añadían efectos para generar luces y sombras y aportar más dimensión a la escena. Por último, entre cada capa de profundidad, se añadía una capa de efecto de tono azul, similar al del cielo, con poca opacidad, esto para simular la lejanía de aquellos objetos que se encontraban más cerca de la línea del horizonte (Fig. 58).



Fig. 57 Portada con colores base. Fuente propia.

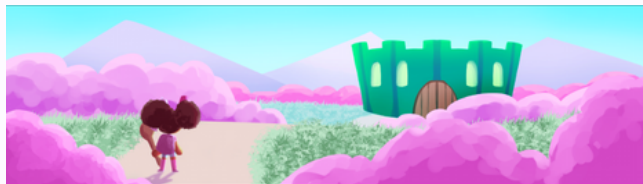
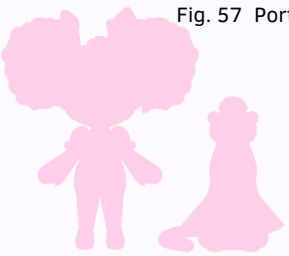


Fig. 58 Portada con efectos y diseño final. Fuente propia.



5.4.2.2.3 Pestañas y botones

En el caso de las pestañas, se empleó un método similar al de los fondos anteriormente mencionados. Se elegía un color base, al que luego se le añadían efectos, principalmente degradados, ya que estas no deberían interferir con la lectura del texto correspondiente a cada interfaz. En caso de llevar efectos más llamativos como tramas, se generaba otro cuadro en el lugar en el que aparecería el texto, para que este se pudiese apreciar y leer correctamente (Fig. 59).

Respecto a los botones, nuevamente la intención principal era que se leyese el texto, por lo que la mayoría de estos utilizan tan solo un color base plano con un borde. No obstante, sí se le hicieron ajustes con capas de efectos para aportar luces y sombras, esto para que resaltasen, aparentasen volumen y se apreciaran mejor que dichas imágenes se pueden pulsar e interactuar con ellas (Fig. 60).

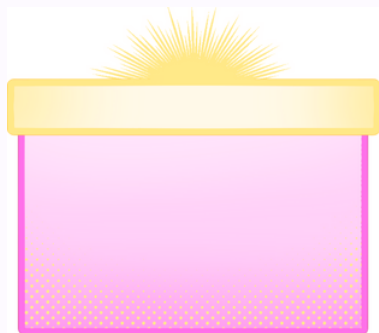


Fig. 59 Ejemplo de pestaña de nuevo minijuego. Fuente propia.



Fig. 60 Ejemplo de botón. Fuente propia.

5.4.2.2.4 Logos e iconos

Teniendo como referencia el diseño de logo anterior desarrollado para asignaturas de "Desarrollo de niveles y jugabilidad" y "Diseño narrativo y de interfaces", se generaron nuevas ideas y siluetas para ofrecer alternativas de diseño (Fig. 61). En estos primeros diseños principalmente se planteó la silueta, la colocación y la tipografía del título del juego. Una vez seleccionado un diseño, se pasó a su refinado de grosor de línea y forma final, para luego hacer nuevamente pruebas, esta vez de color (Fig. 62). Ya con todo seleccionado y aprobado por todas las integrantes del equipo, se dio paso a los retoques y detalles finales del logo para darle el mejor acabado posible (Fig. 63)



Fig. 61 Pruebas de diseño del logo.
Fuente propia.



Fig. 62 Pruebas de color del logo.
Fuente propia.



Fig. 63 Diseño final del logo.
Fuente propia.

En cuanto al icono de escritorio del juego, nuevamente se prepararon bocetos que se presentaron al equipo (Fig. 64). Estos estaban planteados para ser una imagen legible, especialmente por ser una imagen tan pequeña. Una vez elegido el diseño, se dio paso a la elaboración final, que siguió el mismo proceso que diseños anteriores: con el boceto como base, se preparó el entintado, se rellenó la silueta de un color plano para evitar transparencias, se añadieron los colores base, y se finalizó con capas de efectos para mayor resalte (Fig. 65).

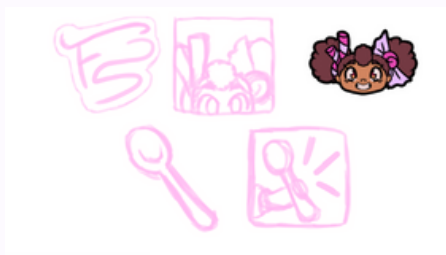


Fig. 64 Bocetos de iconos de escritorio.
Fuente propia.



Fig. 65 Diseño final del icono del juego.
Fuente propia.

Respecto a los iconos de imágenes y objetos del juego, se prepararon diferentes imágenes siguiendo los mismos pasos que con las anteriormente mencionadas. En este caso lo principal era que el jugador reconociese claramente todos los objetos que se le presentaban en pantalla, así como su funcionalidad. Para situar, por ejemplo aquellos consumibles que otorgan mejoras como defensa o ataque, tienen un aura y un brillo en su objeto del inventario, y una vez consumidos, un escudo o una espada también con brillos, retroalimentando al jugador de forma visual de lo que acaba de pasar al consumir dicho objeto. Para los logros de cantino, se decidió optar por crear un sistema de monedas, de plata y de oro, que cada una de estas otorgaba un premio diferente. El primer premio de las monedas de plata toma referencia de un asset de otro juego “*Five nights at Freddy’s: Sister Location*” (Fig. 66) , mientras que el segundo es de diseño original, siendo una estatuilla de Cantino sosteniendo un trofeo (Fig. 67).





Fig. 66 Imagen del juego (*Five nights at Freddy's: Sister Location*, 2016), referencia del premio de Cantino, fuente freddy-fazbears-pizza.fandom.com

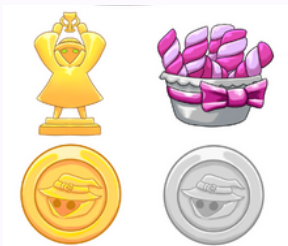


Fig. 67 Recompensas de Cantino.
Fuente propia.

También se diseñaron diferentes barras de estado para el tiempo restante de los minijuegos, vida de la protagonista y enemigos o estamina. El diseño de estas resultó sencillo, se utilizaba una herramienta de forma con una línea del grosor estipulado y un color de relleno, con el que se creaba la forma. A algunas, para ser más llamativas o vibrantes se les daban ciertas texturas como tramas o degradados (Fig. 68). Se debían diseñar dos tipos: una completamente llena y otra completamente vacía.



Fig. 68 Barra de vida de jefe. Fuente propia.

5.4.2.3 Golden

Esta fase se centró en el arreglo de errores y en la implementación de sugerencias de los *testers* para obtener el resultado final del proyecto. Fue una de las fases de menos carga de trabajo, pero las tareas realizadas aquí eran de vital importancia, puesto que el resultado final del proyecto se obtendría al finalizar esta fase.

Se solucionó uno de los mayores problemas que encontraron los testers, relacionado con el menú de objetos. El rediseño que se elaboró en un primer momento para el sistema de inventario no se llegaba a entender bien, llegando incluso a ser confuso para la gente que probaba el juego. Este menú constaba con diversas animaciones, que se activaban si el cursor estaba encima de un objeto. En muchas ocasiones no funcionaban correctamente y la solución que se encontró consumía muchos recursos por lo que se optó por realizar un nuevo diseño, más simple, pero con una funcionalidad correcta (Fig. 69).



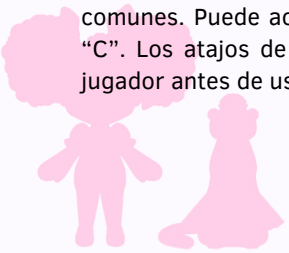
Fig. 69 Rediseño e implementación final del inventario. Fuente propia.

Se le añadió al *HUD* de batalla qué ataque se encontraba seleccionado en el momento de seleccionar los enemigos, ya que se comentó que en algunas ocasiones cuando luchabas contra varios enemigos no se sabía muy bien si ya podías atacar o no por que no se sabía si el ataque estaba pulsado. Se añadió también una manera visible de ver el tiempo de espera de cada ataque, para que el jugador pudiese crear mejor su estrategia ya que hasta el momento no se sabía en qué turno podría lanzar el ataque de nuevo.

Se añadió la opción al diálogo de ver todo el texto clicando dos veces saltándose así la animación del texto, puesto que en ocasiones se hacía lento o repetitivo.

Se añadieron resoluciones de modo ventana adaptadas a la pantalla de cada equipo, puesto que en muchos de los equipos de los *testers* algunas de las resoluciones propuestas originalmente eran demasiado grandes, por lo que no cumplía la función de “modo ventana”.

Se añadieron los atajos para acceder a los menús, ahora el jugador puede acceder más rápidamente a los menús más comunes. Puede acceder al mapa pulsando la “M”, al inventario pulsando la “I” y a las estadísticas del jugador pulsando la “C”. Los atajos de teclado también cuentan con la función de regresar a la pantalla de juego en la que se encontraba el jugador antes de usarlo.



Se realizó una modificación completa en el funcionamiento del mapa. El funcionamiento original era incómodo e inestable, por lo que muchos *testers* preferían no usarlo, perdiendo así una herramienta valiosa. La nueva versión se llevó a cabo realizando todos los cálculos necesarios manualmente, a diferencia de la anterior en la que se aprovechaban herramientas que brinda el propio motor. Se añadieron visualmente letras al minimapa, algunos *testers* mencionaron que en las misiones se indicaba ir en una dirección concreta y que un elemento visual que indicase las direcciones ayudaría al jugador.

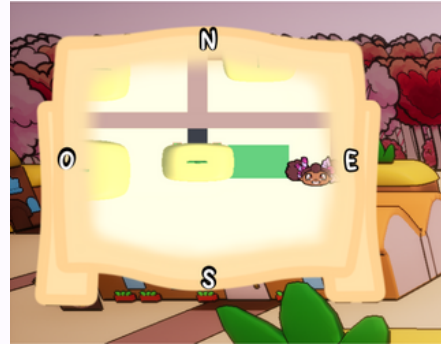


Fig. 70 Implementación de indicadores de direcciones. Fuente propia.

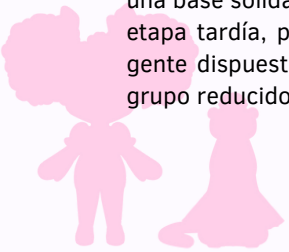
Se añadieron frases y avisos al jugador sobre la función de ciertas pantalla de juego que se desbloquean según se avanza en la historia.

Los botones que tuviesen la misma funcionalidad pero en pantallas distintas se sugirió que se colocaran en el mismo sitio ya que cambiar la posición despistaba al jugador o le hacía pensar que tenían una funcionalidad diferente.

5.4.3. Conclusión individual

El resultado final que se buscaba en el departamento de interfaces era conseguir transmitirle al jugador toda la información posible y que no se sintiese fuera de lugar en ningún momento. También se buscaba crear unas interfaces intuitivas y fáciles de usar, aunque con un estilo propio y unificado. Otra de las prioridades era que pudiese adaptarse a todo tipo de público, sobre todo mediante la implementación de un sistema de traducción y la posibilidad de adaptar los controles. Por otro lado, hubo metas que no se pudieron alcanzar o las cuales se tuvieron que descartar, priorizando tareas de mayor importancia. Un ejemplo de ello fue la priorización de interfaces más sencillas pero completamente funcionales, a interfaces más dinámicas a través del uso de animaciones pero que podían dar lugar a confusión al realizarse a medias.

Por otro lado, uno de los problemas o conflictos más importantes que se tuvieron a lo largo de todo el proyecto fue la falta de pruebas de las diferentes versiones del proyecto. Se fueron sacando diversas versiones del proyecto a medida que las mecánicas se iban incrementando, puesto que era mejor que se probaran poco a poco las cosas. Así se aseguró una base sólida sobre la que seguir trabajando sin temor a que un error en primera instancia afectase al proyecto en una etapa tardía, pues su solución podría requerir más tiempo del disponible. Pero a pesar de disponer de un equipo de gente dispuesta en un principio a probar el juego, las circunstancias no permitieron que esto sucediera así y solo un grupo reducido de personas pudieron probar el proyecto y transmitir su opinión al respecto.



Los fallos reportados fueron arreglados y sirvieron de gran ayuda, pero a medida que pasaba el tiempo y más mecánicas iban apareciendo muchos de los fallos que en su momento nadie percibió, empezaron a ser reportados junto a los fallos nuevos de las últimas mecánicas añadidas, por lo que en un momento el equipo se encontró delante de una lista extensa de elementos a arreglar y el tiempo restante hasta la entrega no era favorecedor a la situación. Por lo que el equipo se vio obligado a realizar un período de *crunch* para poder arreglar lo que fallaba hasta el momento y continuar con las tareas pendientes para poder conseguir un resultado que se pudiese considerar final.

A nivel personal, con este proyecto he conseguido poner en práctica una gran parte de los conocimientos que he adquirido durante estos últimos 4 años en las diversas asignaturas del grado. También he ampliado mi conocimiento gracias a la constante investigación, como al afrontar los problemas encontrados. He experimentado que el buen resultado de un proyecto consiste en priorizar el funcionamiento de los elementos esenciales y no en añadir elementos genéricos, que al final lo que consiguen es distraer del objetivo y la esencia del proyecto.



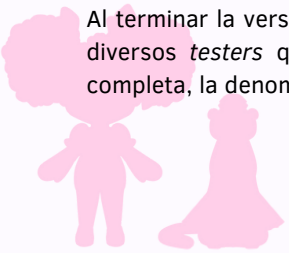
5.5. Departamento de programación

5.5.1. Objetivos propuestos

Dentro del departamento de programación se dividió el trabajo en la publicación de tres versiones con diferentes objetivos a cumplir. En la versión *Pre-Alpha* se buscó programar la funcionalidad primordial para luego poder ampliar el videojuego desde una base sólida. En esta fase destaca la implementación del combate por turnos, el sistema de guardado y la optimización del mundo mediante la carga y descarga de subniveles.

En la siguiente versión del videojuego, la denominada *Alpha*, se buscó conseguir un producto entretenido a la par que funcional. Para ello se dividieron las tareas a realizar en el sistema de exploración y el sistema de batalla. Para el primero se creó el sistema de misiones tanto principales como secundarias y diversas mecánicas para guiar al jugador y llenar de vida el mundo del videojuego. Por otro lado, el combate se completó implementando los ataques basados en minijuegos y la batalla especial contra el enemigo final.

Al terminar la versión *Alpha*, se desarrolló la *Beta* añadiendo el apartado visual. Tras ello se compartió el videojuego con diversos *testers* que ayudaron a identificar los errores y mejoras necesarias para conseguir una versión final más completa, la denominada *Golden*.

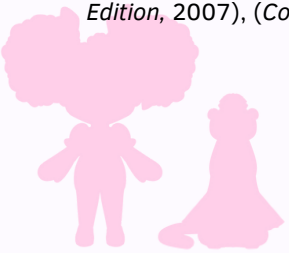


5.5.2. Proceso

Antes de comenzar a implementar en el motor las mecánicas descritas en el Documento de mecánicas esenciales, documento de diseño desarrollado en la asignatura “Desarrollo de Niveles y de Jugabilidad” (anexo 07), se clasificaron las tareas por rango de importancia en un listado de tareas (anexo 09) para poder informar a la productora Maite Real de qué partes deberían estar desde el principio y cuáles se podrían ir añadiendo según el tiempo que tuviésemos disponible.

Tras ello se investigó cómo conseguir programar todas esas mecánicas de forma eficiente. Se revisó lo aprendido en las asignaturas “Fundamentos de Programación”, “Programación Orientada a Objetos”, “Desarrollo de Videojuegos 1”, “Desarrollo de Videojuegos 2”, “Programación de Videojuegos”, “Videojuegos 2D” y “Proyecto de Videojuego”. También se consultaron los foros de Unreal Engine y diversos canales de *YouTube* especializados en desarrollo de videojuegos (enlaces disponibles en el apartado 8.2.2 Bibliografía utilizada por Eva Quintela Balloira).

Por otro lado, se jugaron y visualizaron numerosos análisis de videojuegos que guardaban algún tipo de relación con el producto que se buscaba obtener. Algunos de ellos fueron (*Pokemon Gold Version*, 2001), (*Final Fantasy 20th Anniversary Edition*, 2007), (*Cooking Mama: Let's Cook!*, 2015) y (*Wario Ware: Smooth Moves*, 2007).



Gracias a esta investigación se consiguió razonar cómo implementar mecánicas como el combate por turnos contra uno o varios enemigos y con diversos ataques a escoger (ejemplo en la Fig. 71) o los minijuegos basados en gestos simples que deben cumplirse en un tiempo limitado (ejemplo en la Fig. 72).



Fig. 71 Combate por turnos con diversos ataques.
Imagen tomada de (*Final Fantasy 20th Anniversary Edition*, 2007)
Fuente: Golden, M. (2020). *Final Fantasy 20th Anniversary Edition* (PSP). Medium. [Enlace al artículo](#).



Fig. 72 Minijuego de cocina basado en gestos simples.
Imagen tomada de (*Cooking Mama: Let's Cook!*, 2015).
Fuente propia.

5.5.2.1. Pre-Alpha

Lo primero que se implementó en el motor fue la estructura básica del nivel principal (Fig. 73). Éste se dividió en diversos subniveles para poder cargarlos y descargarlos según la posición del jugador, optimizando así el futuro funcionamiento del videojuego al solo tener una zona del mapa cargada. Para disimular este proceso, se implementó un área que tras interactuar con el jugador oscurece la pantalla para poder cargar y descargar los subniveles correspondientes sin que se note el cambio.



Fig. 73 Estructura del nivel principal.
Fuente propia

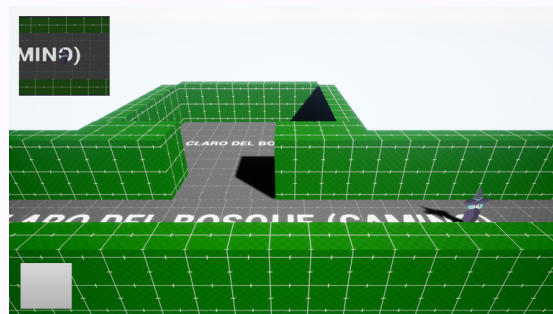
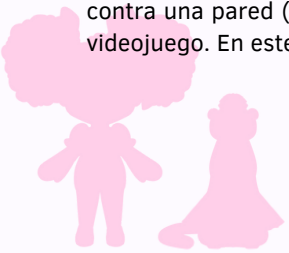


Fig. 74 La cámara se detiene cuando el jugador llega al borde de la zona. Fuente propia

Para cambiar entre zonas del mapa, se planteó que la cámara se detuviese y dejase de seguir al jugador al acercarse al borde, para evitar que viese los niveles sin cargar (ejemplo en la Fig. 74). Al principio se intentó usar los llamados “*Camera blocking volumes*” proporcionados por *Unreal Engine*, pero se descartó esta opción porque no se conseguía el efecto esperado y se decidió crear un sistema propio. En la primera versión, este sistema consistía en un volumen cerca del cambio de subnivel que al interactuar con el jugador recogía la posición de su cámara y se la colocaba a una cámara externa. Una vez colocada esta cámara fija, se cambiaba el punto de vista del jugador a esta cámara. Con esto se consiguió que mientras el jugador estuviese cerca del cambio de zona, la cámara se encontrase fija en el borde y no siguiese al jugador.

Esta versión fallaba cuando el jugador salía del área de bloqueo de la cámara por una zona diferente de la que había utilizado para entrar, ya que la cámara daba un salto y sacaba al usuario de la inmersión conseguida. Para solucionarlo se reprogramó el sistema creando un área para delimitar la zona del mapa con una cámara acoplada. Mientras el jugador se encontrase dentro de esta zona, la cámara le seguiría, pero en el momento que éste saliese por alguno de sus límites, la cámara se quedaría quieta. Ahora bien, si el jugador saliese por la parte horizontal, la cámara dejará de moverse en ese eje pero continuará siguiéndole en el eje vertical. Con esto se consiguió que la cámara pareciese que se estaba chocando contra una pared (contra el límite de la zona) y en ningún momento hay saltos que impidan la inmersión del jugador en el videojuego. En este [enlace](#) se puede observar un vídeo resumen de este sistema con sus distintas versiones.



Por otro lado, para empezar a añadir jugabilidad, se creó una zona de tamaño variable donde aparecerán enemigos escondidos en posiciones al azar (Fig. 75). Si el jugador colisiona con alguno de ellos, entrará en combate. Éste se realiza en un nivel diferente, por lo que para evitar perder información al cambiar entre sistemas, se implementó un guardado temporal donde se registra el progreso del jugador. Este sistema se reutilizó posteriormente para añadir un autoguardado que impidió que el usuario perdiese su desarrollo en el videojuego aunque no utilizase los botones designados para guardar la partida.



Fig. 75 Zona de combates aleatorios donde los cubos representan a los enemigos escondidos. Fuente propia.

En esta versión *Pre-Alpha* también se implementó el funcionamiento primordial del sistema de batalla. Primero se programaron las reglas para la colocación de los enemigos según su tipo y la cantidad (Fig. 76 y 77). Los enemigos débiles siempre se colocan en la primera fila y dentro de ésta su posición depende de la cantidad: si sólo hay un enemigo éste se pondrá en el carril central, si hay dos se crearán en carriles separados y si hay tres se ocuparán todos los carriles.



Fig. 76 Esquema de batalla con un enemigo débil y dos enemigos fuertes. Fuente propia.



Fig. 77 Esquema de batalla con tres enemigos débiles. Fuente propia.

Tras crear y colocar a los enemigos, se hace una media de sus estadísticas de velocidad para saber quién recibe el primer turno, si el bando del jugador o el de sus contrincantes. Si la velocidad del jugador es mayor, éste recibirá el primer turno y podrá escoger entre tres opciones: atacar, usar el inventario o huir (Fig. 78).

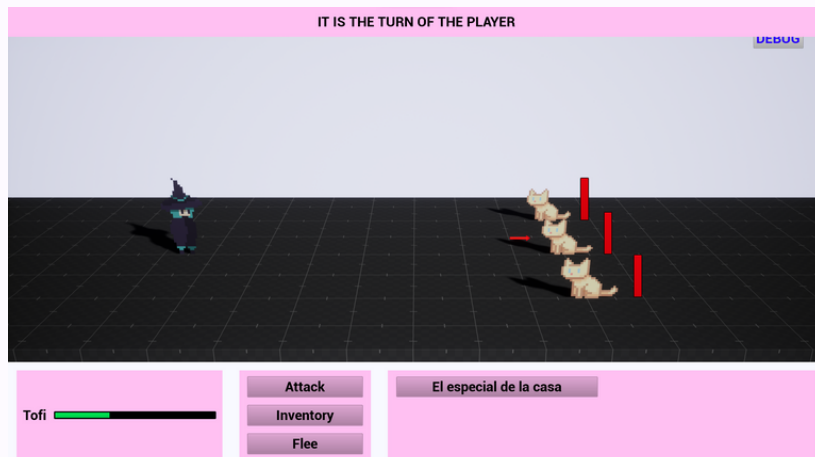


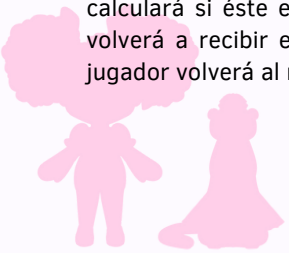
Fig. 78 Sistema de batalla en la versión *Pre-Alpha*.
Fuente propia.

-Atacar: el jugador debe seleccionar el ataque y el enemigo que quiera dañar. Una vez hecho esto, se calcula mediante un porcentaje relacionado con la velocidad si el enemigo esquivará el ataque o no. Si éste lo esquiva, no recibirá daño y el jugador perderá el turno; si no consigue evitarlo, la vida del enemigo descenderá el valor del ataque del jugador. Si el usuario consigue derrotarlos a todos, la batalla terminará y el jugador recibirá los puntos de experiencia correspondientes a la cantidad y tipo de enemigos superados.

-Inventario: si el jugador selecciona el botón del inventario, se abrirá el menú correspondiente y en él podrá consumir una poción para recuperar vida, perdiendo el turno tras tomarla.

-Huir: por último, al pulsar en el botón de huir se calcula de nuevo quién tiene más velocidad de los dos bandos, pero añadiendo un valor aleatorio a la media de los enemigos. Si tras esta operación el jugador sigue siendo más rápido, podrá escapar y salir de la batalla, si es más lento, perderá el turno y deberá continuar en el combate.

Por otro lado, en el turno de los enemigos estos atacarán de forma individual al jugador y por cada uno de los ataques se calculará si éste esquivará o recibirá el golpe. Cuando todos los enemigos terminen de lanzar sus ataques, el jugador volverá a recibir el turno. Sin embargo, si alguno de éstos derrota al jugador en el proceso, la batalla terminará y el jugador volverá al nivel de exploración sin ninguna recompensa.



5.5.2.2. Alpha

Tras conseguir un videojuego sólido pero demasiado simple en la fase anterior, se implementaron el resto de mecánicas para conseguir la versión *Alpha* y tener un producto más disfrutable. En esta fase el trabajo de programación consistió en desarrollar en profundidad los dos sistemas principales: el sistema de exploración y el de batalla.

5.5.2.2.1 Sistema de exploración

Para poder implementar la historia principal del videojuego se analizó el documento donde Maite Real definió qué y cómo debería ocurrir en cada una de las misiones principales. Gracias a ello se identificó la estructura que debería tener interiormente el sistema: cada misión incluiría una serie de objetivos que el jugador debe cumplir para poder continuar con la historia. Se definieron varios tipos de objetivos con distintas características y variables personalizables (Fig. 79), para permitir implementar en el motor las misiones previamente diseñadas.



Fig. 79 Esquema con los tipos de objetivos y sus variables personalizables. Fuente propia.

Según el tipo de objetivo seleccionado, se crearán los elementos necesarios para cumplirlo (“Enemigo” hará que aparezcan enemigos en el mapa que el jugador debe vencer, “Localización” añadirá un área con la que el usuario debe interactuar, etc.). Dentro de cada misión se pueden definir todos los objetivos necesarios para cumplirla y cuando el usuario los complete pasará a la siguiente de forma automática. En todo momento se guarda en que misión principal se encuentra el jugador y cuantos objetivos de la misma ha cumplido, para que el usuario pueda dejar de jugar cuando prefiera y no pierda ningún progreso.

Un ejemplo de cómo se implementaría en el motor una de las misiones planteadas sería: en “¿Qué está pasando?” se comienza con una cinemática donde la protagonista, Tofi, se ve obligada a huir y en el proceso se encuentra con unos enemigos. Ésta debe derrotarlos y tras ello seguir huyendo hacia el pueblo más cercano. Esta misión dentro del motor se conseguiría con tres objetivos: una cinemática, una batalla automática y una localización. Nada más empezarla, se reproduciría una cinemática y en cuanto termine se abrirá el nivel de batalla con los enemigos ya definidos. Si el jugador consigue derrotarlos, deberá llegar a la zona del pueblo para completar la misión y poder continuar con la historia.

Una vez implementado el sistema, se simplificaron al máximo las variables presentes en el editor para que se pudiese modificar cualquier misión de la forma más rápida y cómoda posible (Fig. 80).

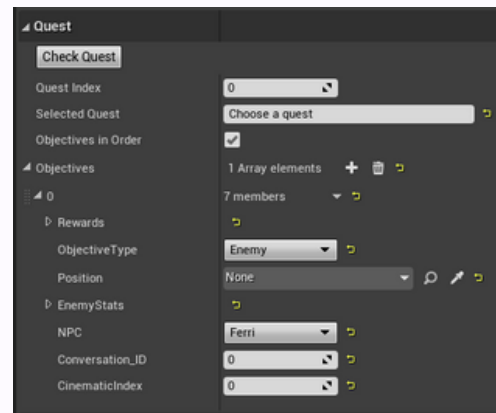


Fig. 80 Variables presentes en el editor de misiones.

Fuente propia.

Tras definir la estructura del sistema de misiones, éste se adaptó para las misiones secundarias. Éstas no ocurren de forma lineal, sino que el jugador puede activarlas al interactuar con *NPCs* si ya ha superado ciertas partes de la historia principal (Fig. 81) . Primero se crearon unos personajes que podrían estar visibles u ocultos según la misión principal que estuviese activa. Esta funcionalidad se añadió para poder dejar la historia de forma coherente: por ejemplo, en un momento un niño se pierde en la zona de las granjas y no tendría sentido que las granjeras que allí habitan estuviesen cerca y no le ayudasen, por lo que durante esa misión estos personajes están ocultos.

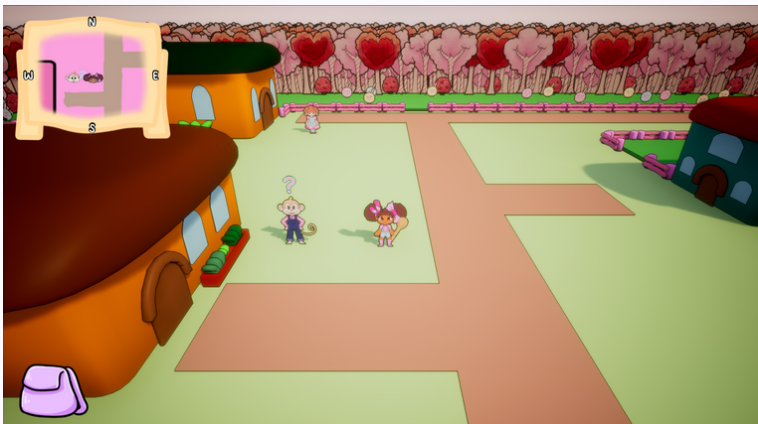


Fig. 81 Personaje con misión secundaria.
Fuente propia.

Luego se programaron estos personajes para que tuviesen distintas líneas de diálogo según el estado de sus misiones secundarias:

-Si el jugador aún no puede desbloquear la misión o ya la ha completado, el personaje dirá una frase genérica acorde a su personalidad.

-Si ya puede conseguirla, se activará el diálogo donde el *NPC* le cuenta qué problema tiene a la protagonista. Al terminar la conversación, se desbloqueará la misión y aparecerán los objetivos que el jugador debe completar para cumplirla.

-Si la misión ya está activa pero aún no se ha completado, el personaje le recordará al jugador qué tiene que hacer para ayudarle.

-Por último, en cuanto complete todos los objetivos (recolectar objetos, derrotar enemigos o hablar con otros personajes), el jugador podrá hablar de nuevo con el personaje para recibir su agradecimiento y una recompensa por sus acciones.

-Si el *NPC* en cuestión puede proporcionar más de una misión secundaria, cuando el jugador complete la primera misión podrá activar la segunda si ya cumple con todos los requisitos necesarios.



5.5.2.2 Sistema de batalla

Tras conseguir un sistema de misiones funcional, se crearon los demás elementos para poder conseguir unos combates completos y disfrutables, añadiendo mecánicas descritas en el *GDD* que aún no estaban implementadas. Primero se añadió un nuevo archivo de guardado para poder crear batallas según los valores definidos por la diseñadora de niveles. En éste se establecía la cantidad de enemigos, el tipo, el nivel y las recompensas que obtendría el jugador al derrotarlos. También se implementó la opción de crear batallas con tutoriales, donde las funcionalidades de ésta se adaptaban a lo que se precisase en el tutorial: bloquear ataques, impedir que el jugador muera, etc. (Fig. 82).

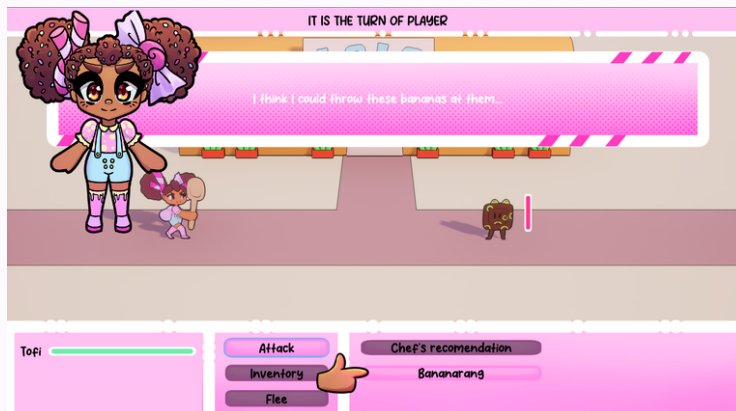


Fig. 82 Combate programado con un tutorial.
Fuente propia.

Por otro lado, se añadieron el resto de ataques y el funcionamiento de los minijuegos que los acompañan, siguiendo las descripciones del *GDD*. Todos se basan en una ventana emergente con un temporizador: si el jugador completa el minijuego antes de que se termine el tiempo, lanzará un ataque exitoso, si no lo consigue, no realizará daño y perderá el turno. Estos ataques tienen distintos porcentajes de daño y distintas áreas de efecto: el ataque “Batidor de merengue” y el “Horneado de pastel” realizan daño en área, es decir, reducirán la vida del enemigo seleccionado y de los que lo rodean; en cambio el resto de ataques solo dañarán a un enemigo. Hay cuatro tipos de ataques especiales con sus respectivos minijuegos:

-”Bananarang” (Fig. 83): para completar el minijuego con éxito, el jugador debe seleccionar el plátano visible en pantalla y arrastrarlo hacia la derecha rápidamente. Si lo mueve en otra dirección o no lo lleva hasta el final de la ventana, fallará el ataque.

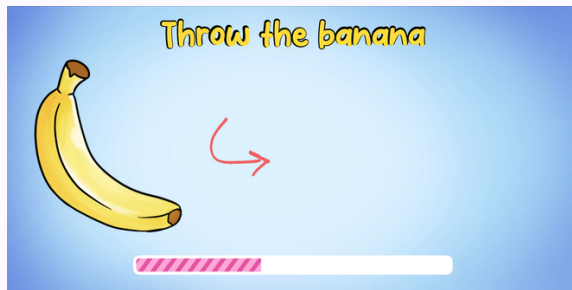


Fig. 83 Pantalla del minijuego “Bananarang”.
Fuente propia.

-”Batidor de merengue” (Fig. 84): en este minijuego se registran los movimientos del cursor del jugador y si éste realiza círculos amplios conseguirá ir añadiendo puntos a la barra de carga del ataque. Cuando esta se llene, se lanzará el ataque de forma eficaz.



Fig. 84 Pantalla del minijuego “Batidor de merengue”.
Fuente propia.

-"Horneado de pastel" (Fig. 85): este minijuego funciona gracias a una barra cuyo indicador se va moviendo cada *frame* y un botón que el jugador puede pulsar para detener dicho indicador. La barra se divide en tres zonas: rosada para indicar que el pastel saldrá a medio cocinar, verde para señalar que el pastel estará perfecto y roja para dejar claro que el pastel se habrá quemado. Estas zonas varían su posición de forma aleatoria cada vez que se cree el minijuego, para evitar que sea siempre igual y se vuelva demasiado sencillo superarlo. Al añadirse a pantalla el minijuego, el indicador de la barra empezará a moverse hacia la derecha y el jugador deberá pulsar el botón cuando éste se encuentre en la zona verde para conseguir lanzar un ataque perfecto. Si lo acciona antes de tiempo, lanzará un ataque *OK* que infligirá menos daño a los enemigos. Si pulsa el botón en la franja roja o el indicador llega al final de la barra, el jugador fallará el ataque y no realizará ningún daño.



Fig. 85 Pantalla del minijuego "Horneado de pastel".
Fuente propia.

-“Bate piruleta” (Fig. 86): este último ataque es el que más daño realiza pero también el más arriesgado. Al añadir el minijuego a pantalla se calcula un tiempo de espera aleatorio y, cuando éste termine, aparecerá un enemigo en pantalla durante un segundo. Si el jugador consigue clicar en pantalla mientras el enemigo esté visible, completará el minijuego y aturdirá al enemigo seleccionado previamente, consiguiendo que éste tenga una probabilidad muy alta de no atacar en los turnos posteriores. Si por el contrario no consigue clicar a tiempo, el jugador fallará el ataque y recibirá un porcentaje de daño.

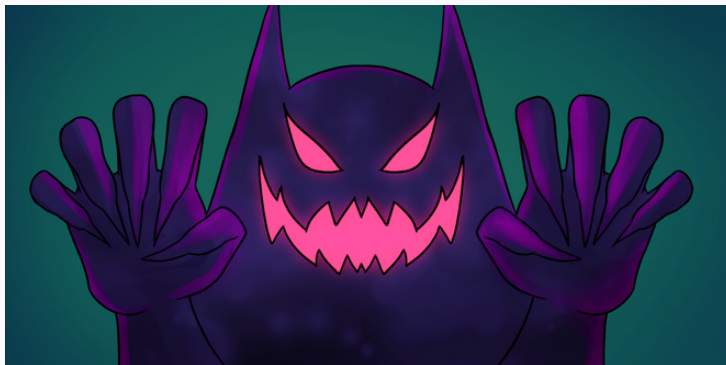
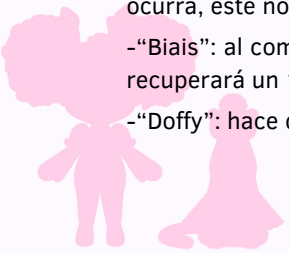


Fig. 86 Pantalla del minijuego “Bate piruleta”.
Fuente propia.

Los ataques especiales con minijuegos no se pueden usar hasta desbloquearlos en las misiones principales correspondientes. Cuando el jugador los consiga, tendrá un tutorial donde se explicará cómo funciona esta nueva mecánica. En éste el jugador solo podrá seleccionar el minijuego recién obtenido y, si se completa de forma satisfactoria, la batalla continuará de forma normal (pudiendo acceder al inventario o utilizar el resto de ataques conseguidos). Si lo falla o no lo realiza perfecto, se abrirá de nuevo la pantalla del minijuego correspondiente para que el jugador vuelva a intentarlo (sucediendo una y otra vez hasta que aprenda a utilizarlo y consiga completarlo de forma exitosa).

También se implementaron diversos efectos que afectaban el transcurso normal de la batalla, proporcionados por los consumibles no curativos y los diferentes compañeros desbloqueables a través de las misiones secundarias. Al consumir un “Azucarillo” el ataque del jugador aumenta un 30% durante una batalla (en cuanto ésta se termine, el ataque volverá a su valor original) y el “Subidón de azúcar” reduce el ataque de los enemigos un 30% para que el jugador reciba menos daño cuando estos le ataquen. Además, si el personaje tiene uno o más compañeros equipados, estos proporcionarán efectos que pueden llegar a cambiar drásticamente el resultado de un combate:

- “Caralleta”: si el jugador muere, este revivirá automáticamente con un 25% de su vida total. Una vez este efecto ocurra, éste no se volverá a activar hasta que pasen 15 turnos (pudiendo llegar a estar varios combates desactivado).
- “Biais”: al completar de forma exitosa los minijuegos de los ataques “Horneado de pastel” o “Bate piruleta”, el jugador recuperará un 10% de su vida total.
- “Doffy”: hace que el efecto de aturdido conseguido con el ataque “Bate piruleta” dure 2 turnos más.



Por último se implementó la batalla especial contra el rey Cucumis, el enemigo final, dividida en dos fases:

-La primera consiste en un combate normal contra tres enemigos fuertes (Fig. 87). Cuando el jugador consiga derrotarlos, comenzará la segunda fase.



Fig. 87 Primera fase de la batalla contra el rey.
Fuente propia.

5. Desarrollo del proyecto

Eva Quintela Baloira

-La segunda fase es la propia batalla contra el rey Cucumis (Fig. 88). En ella el jugador puede atacar al *Boss* y cuando su vida llegue al 75%, 50% y 25% de su total, saltará la fase especial. En ésta el enemigo se lanzará rodando hacia el jugador y éste deberá esquivarlo moviéndose arriba o abajo en tres carriles designados. El rey irá a por él en intervalos aleatorios de tiempo y siempre se moverá justo antes de empezar la animación al carril en el que se encuentre el jugador, para obligarle a estar atento y escapar de él. Si el rey entra en contacto con el jugador, éste morirá automáticamente y aparecerá en pantalla la opción de abandonar la batalla para probar de nuevo más tarde o volver a intentarla desde la fase en la que se encuentra.



Fig. 88 Segunda fase de la batalla contra el rey.
Fuente propia.

Cuando al *Boss* le quede menos del 5% de su vida total, aparecerá automáticamente en pantalla el minijuego del ataque "Horneado de pastel". Si el jugador completa el minijuego de forma perfecta, derrotará al rey y ganará la batalla. Si consigue un *OK* o lo falla, el enemigo recuperará un 20% de su vida total y el jugador deberá volver a atacarle hasta dejarlo de nuevo en menos del 5% de vida (repitiéndose el proceso hasta conseguir lanzar el ataque de forma exitosa).

5.5.2.3. *Golden*

En el último mes de desarrollo se añadieron los *sprites* y modelos 3D desarrollados por el equipo de arte y se publicó la versión *Beta*, para obtener opiniones externas al equipo. Gracias a ello se corrigieron errores no encontrados hasta el momento e implementaron la mayoría de las mejoras propuestas para poder conseguir una mejor versión final. Una de los problemas más importantes a la hora de jugar era que el usuario se perdía y no sabía que debía hacer, por lo que se añadieron diversos elementos para guiar mejor al jugador:

-Enemigos imposibles: hay algunas zonas del mapa a las que no debería poder accederse hasta cierto punto de la historia principal, por lo que para bloquear el camino de forma clara y que no sacase de la experiencia al jugador, se añadieron enemigos que al acercarse a cierta distancia parpadean en color rojo con una calavera sobre sus cabezas para indicar que son demasiado complicados. Si el jugador decide seguir acercándose, Tofi dirá una frase del estilo “Estos enemigos son demasiado fuertes, no creo que pueda derrotarlos”.

-Barreras invisibles: para evitar que el jugador se salte partes importantes de la historia yendo por caminos alternativos, estos se bloquearon con volúmenes que al interactuar con el jugador activan un diálogo del acompañante de la protagonista, Ferri. Éste le recuerda al jugador a que zona del mapa debería ir y tras ello se empuja al jugador en la dirección adecuada, impidiéndole seguir el otro camino.



-Casas en los pueblos: algunos *testers* al jugar no entendían en qué casas se podía entrar y cuáles estaban cerradas. Para solucionarlo se añadió un elemento visual en la entrada que cambia de color según se pueda entrar o no y si el jugador interactúa con una de las inaccesibles, Tofi informa al jugador (Fig. 89).

-*NPCs* de relleno: se crearon unos personajes que sueltan una frase genérica al interactuar con ellos (Fig. 90). Para añadir más contexto visual al mundo, se hizo que cambiasen de expresión y de frase cuando se encuentran en una casa ocupada por enemigos, pasando de estar felices a estar asustados y preocupados mientras el jugador no se deshaga de la amenaza. Luego al rescatarlos éstos agradecerán el gesto al jugador. Estos pequeños detalles ayudaron a que el usuario tuviese una experiencia más inmersiva.



Fig. 89 Aviso al intentar entrar en casas cerradas.
Fuente propia.



Fig. 90 Personajes sin importancia en la historia.
Fuente propia.

Por otro lado, varios de los jugadores nos informaron de que el sistema de selección de enemigos no era demasiado intuitivo y muchas veces incluso llegaba a ser incómodo, por lo que se decidió rediseñar completamente esa parte del sistema de batalla. En la primera versión debías seleccionar al enemigo que querías dañar y después el ataque que querías lanzar. En el momento de decidir qué ataque realizar no había ningún tipo de señalización de si el daño iba a afectar a un solo enemigo o a varios (solo podía saberse después de realizar el minijuego, ya que los enemigos afectados pasarían a tener la animación de recibir daño).

En la nueva versión el jugador debe seleccionar el ataque que considere más adecuado y al pasar el cursor por encima de los enemigos, éstos se remarcarán para señalar quienes recibirán daño (Fig. 91). Se puede cambiar en todo momento el ataque seleccionado, ya que este no se lanzará hasta que el jugador clique encima de uno de los enemigos. Además, para hacer la batalla más fluida, cuando el jugador luche contra un solo contrincante, los ataques se lanzarán automáticamente al pulsar en el botón correspondiente, sin necesidad de clicar en el enemigo.

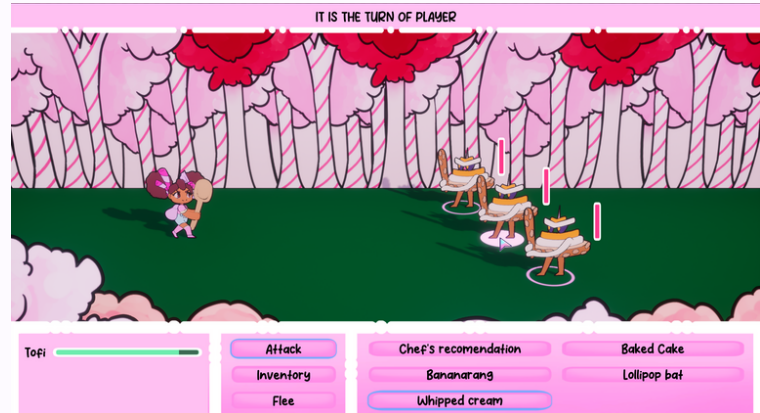


Fig. 91 Nuevo sistema de selección de enemigos.
Fuente propia.

También se nos comunicó que la batalla contra el enemigo final no se entendía claramente y parecía que las cosas sucedían de forma aleatoria. Para solucionarlo, se añadieron pequeños avisos al comenzar las partes especiales del combate:

-La primera vez que el enemigo entra en su fase especial, cuando el rey se lanza hacia el jugador el tiempo se ralentiza y Ferri aparece avisando a Tofi de que debe moverse para esquivar el ataque del enemigo. Cuando el usuario termine de leer el diálogo, el tiempo volverá a la normalidad.

-Cuando salta el ataque automático para rematar al *Boss*, Ferri vuelve a aparecer para avisar al jugador de que necesita completar el minijuego de forma exitosa para derrotarlo. Si el jugador lo falla, el animal informará de nuevo al usuario diciendo que el ataque no ha sido suficiente y el rey se ha cabreado de nuevo, aumentando su vida.

Otra de las críticas recibidas fue que los combates se volvían demasiado sencillos al desbloquear los ataques más fuertes porque se permitía lanzarlos una vez tras otra sin ningún tipo de desventaja. Se decidió optar por añadirles un tiempo de espera a los minijuegos tras usarlos, obligando al jugador a seleccionar cualquiera de los otros ataques que tenga disponibles mientras espera. Cuanto más daño haga el ataque, más turnos permanecerá bloqueado. Con esto los jugadores volvieron a utilizar el ataque básico (ataque que tras conseguir el primer minijuego dejaba de tener utilidad) y empezasen a planificar más estrategias de combate, dejando algunos ataques guardados para cuando más los necesitasen.



Para profundizar más en cualquiera de los contenidos expuestos a lo largo de esta parte de la memoria se recomienda ver el anexo 04. En él se encuentran explicadas en profundidad todas las mecánicas implementadas y se incluyen enlaces a las *blueprints* creadas.

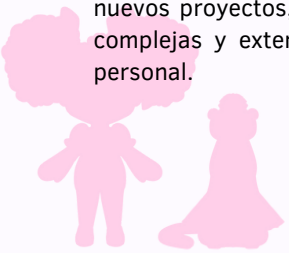
5.5.3. Conclusión individual

En el departamento de programación la meta principal fue conseguir un videojuego libre de errores y optimizado para cualquier ordenador. Además, más relacionado con el propio contenido del proyecto, se propusieron objetivos como conseguir un sistema de misiones completo y un sistema de batalla funcional y divertido. También se buscó facilitar el trabajo posterior de la diseñadora de niveles dejando en el editor solo variables básicas y fácilmente entendibles, aunque la programación interna de dichas funcionalidades fuese mucho más compleja. Todas estas metas consiguieron llevarse a cabo, tal y como puede observarse en el producto final, pero algunas otras no fue posible realizarlas. Se subestimaron los tiempos necesarios para implementar algunas de las mecánicas, como el sistema de misiones o el de batalla, y lo que se tardaría en corregir todos los errores que aparecieron al programar un videojuego tan complejo.



Uno de estos objetivos no realizados fue añadir código al proyecto de *Unreal Engine*, a la par que la programación mediante *blueprints*, ya que se podrían haber creado mecánicas aún más complejas y resolver problemas de forma más eficiente. Por otro lado, se propuso como meta conseguir un producto aprobado por un diverso grupo de *testers* antes de publicar la versión final, pero se comenzó el período de pruebas demasiado tarde. Cuando se publicó una de las versiones que se consideraba prácticamente una *Golden*, los *testers* se encontraron con gran cantidad de errores y nos propusieron diversas sugerencias de mejoras que se podrían haber implementado ya en versiones anteriores, pero al descubrirse tan tarde el equipo pasó por varios períodos de *crunch* en el último mes de desarrollo para poder conseguir la versión final. Esto se habría solucionado publicando versiones del proyecto de forma más recurrente y con mayor antelación, además de buscando un grupo de *testers* más amplio.

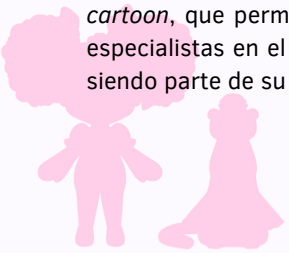
En conclusión, en lo personal, este proyecto me ayudó a mejorar como programadora, ya que me obligó a investigar nuevas formas de resolver problemas que nunca se me habrían siquiera planteado. También me sirvió para mejorar mi habilidad para trabajar en equipo, ya que me apoyé muchísimo en mis compañeras a la hora de tomar decisiones o encontrar soluciones, intentando no trabajar siempre de forma autosuficiente. Por otro lado, aprendí que a la hora de enfrentarme a nuevos proyectos, debo no subestimar los tiempos, añadiendo siempre períodos de descanso tras desarrollar mecánicas complejas y extenuantes a nivel mental. En definitiva, con este proyecto pude aprender muchísimo y mejorar a nivel personal.



En la etapa inicial del desarrollo, el equipo dedicó una gran cantidad de tiempo a estudiar el contenido final con el que contaría el proyecto. Se decidió realizar una reducción de elementos, dado que la magnitud del producto no era viable para su realización teniendo en cuenta el tamaño del equipo y el tiempo disponible. Dicha reducción conllevó a la eliminación de elementos cosméticos de baja prioridad, así como mecánicas de un solo uso que resultarían en un malgasto de recursos. A pesar de haber depurado considerablemente la cantidad de elementos, al realizar un análisis crítico del desarrollo, el equipo opina que esta reducción podría haber sido aún mayor, puesto que el proyecto continuaba siendo demasiado ambicioso en cuanto a contenido, lo que provocó que se tuviesen que realizar en ocasiones largas jornadas de trabajo.

La comunicación mantenida a lo largo del desarrollo fue clave para su correcta finalización, el equipo mantuvo una línea de comunicación abierta y sincera, que permitió un flujo de trabajo eficiente. Esta comunicación permitió a su vez minimizar los errores a la hora de trabajar conjuntamente en *Unreal Engine* empleando *GitHub*, ya que las integrantes del equipo debían avisar de los elementos en los que necesitaban trabajar, evitando así que se sobrescribiese el trabajo de otra persona.

Dado que el equipo está formado por personal técnico en su mayoría, se decidió optar por un estilo artístico sencillo y *cartoon*, que permitiese a los miembros técnicos del grupo apoyar al equipo de arte en las tareas necesarias aún sin ser especialistas en el departamento. Este acabado basado en colores vibrantes y sencillos favorece además al producto final, siendo parte de su personalidad y dejando que la jugabilidad sea el elemento destacable.



7. Anexos y apéndices

Flavor Savior

Anexo 01: Gameplay - Ángela Arbe Núñez, Eva Quintela Baloiira, Maite Real Mahía, Cristina Rodríguez Rodríguez

Anexo 02: Libro de arte - Ángela Arbe Núñez, Maite Real Mahía

Anexo 03: Libro de diseño - Maite Real Mahía

Anexo 04: Libro de programación - Eva Quintela Baloiira

Anexo 05: Libro de interfaces - Cristina Rodríguez Rodríguez

Anexo 06: Guion del juego multilinguaje - Maite Real Mahía

Anexo 07: Documento de mecánicas esenciales - Ángela Arbe Núñez, Eva Quintela Baloiira, Maite Real Mahía, Cristina Rodríguez Rodríguez

Anexo 08: Canal de *YouTube* en el que se encuentran vídeos sobre el desarrollo del proyecto ([Enlace](#)). - Ángela Arbe Núñez, Eva Quintela Baloiira, Maite Real Mahía, Cristina Rodríguez Rodríguez

Anexo 09: Organización del proyecto - Ángela Arbe Núñez, Eva Quintela Baloiira, Maite Real Mahía, Cristina Rodríguez Rodríguez

8.1 Bibliografía común

Rogers, S. (2014). *Level Up! The Guide to Great Video Game* (Second edition). John Wiley & Sons.



8.2 Bibliografía individual

8.2.1 Bibliografía utilizada por Ángela Arbe Núñez

Thomas, F., & Johnston, O. (1981). *Disney Animation: The Illusion of Life*. Abbeville Press.

Para diversas consultas relacionadas con la cinemática, se utilizó: Foro de *Blender* (<https://www.blender.org/>)

Cawthon, S. [Scott]. (2016). *Five Nights at Freddy's: Sister Location* (Windows version) [Videojuego]. Scott Cawthon.

Tanabe et al. (2012). *Paper Mario: Sticker Star* (Nintendo 3Ds version) [Videojuego]. Intelligent Systems.

Shota et al (2018). *Identity V* (Windows version) [Videojuego]. NetEase Games.

Lee et al (2021). *Cookie Run: Kingdom* (Android version) [Videojuego]. Devsisters.

Asano et al (2018). *Octopath Traveler* (Microsoft version) [Videojuego]. Square Enix y Acquire.

Bercaru et al (2022). *Bear and Breakfast* (Microsoft version) [Videojuego]. Gummy Cat.

TutsByKai. (n.d.). *Home [YouTube channel]*. *YouTube*. Recuperado el 22 de junio de 2023, desde <https://www.youtube.com/@TutsByKai>

kdsketch. (n.d.). *Home [YouTube channel]*. *YouTube*. Recuperado el 22 de junio de 2023, desde <https://www.youtube.com/@kdsketch>

8.2.1 Bibliografía utilizada por Eva Quintela Baloira

Para diversas consultas relacionadas con el motor, se utilizó: Foro de *Unreal Engine* (<https://dev.epicgames.com/community/unreal-engine>)

Para consultas sobre programación, se utilizó: *Stack Overflow* (<https://stackoverflow.com/questions>)

Ryan Laley. (n.d.). *Home [YouTube channel]*. YouTube. Recuperado el 22 de junio de 2023, desde <https://www.youtube.com/@RyanLaley>

underscore. (n.d.). *Home [YouTube channel]*. YouTube. Recuperado el 22 de junio de 2023, desde <https://www.youtube.com/@underscore00>

Beardgames. (n.d.). *Home [YouTube channel]*. YouTube. Recuperado el 22 de junio de 2023, desde <https://www.youtube.com/@Beardgames>

Leyendas & Videojuegos. (n.d.). *Home [YouTube channel]*. YouTube. Recuperado el 22 de junio de 2023, desde <https://www.youtube.com/@LeyendasyVideojuegos>

Izushi et al. (2001). *Pokemon Gold Version* (Game Boy Color version) [videojuego]. Game Freak.

Watanabe et al. (2007). *Final Fantasy 20th Anniversary Edition* (PSP version) [videojuego]. TOSE.

Takeshi et al. (2015). *Cooking Mama: Let's Cook!* (Android version) [videojuego]. Office Create.

Sakamoto et al. (2007). *Wario Ware: Smooth Moves* (Wii version) [videojuego]. Nintendo SPD; Intelligent Systems.

8.2.1 Bibliografía utilizada por Maite Real Mahía

Novak, J. (2011). *Game Developing Essentials: An introduction (Third edition)*. Cengage Learning.

Para la escritura del guion del juego en español, se recurrió a la web: Real academia española (<https://www.rae.es/>)

Para la escritura del guion del juego en gallego, se recurrió a la web: Real academia galega (<https://academia.gal/inicio>)

Para la escritura del guion del juego en inglés, se recurrió a la web: *WordReference* (<https://www.wordreference.com/>)

Para diversas consultas relacionadas con el motor, se utilizó: Foro de *Unreal Engine* (<https://dev.epicgames.com/community/unreal-engine>)

Eurogamespain. (n.d.). *Home [YouTube channel]*. YouTube. Recuperado el 22 de junio de 2023, desde <https://www.youtube.com/@eurogamerspain>

Polatrite. (2021, 27 de mayo). *Progression curves? No problem! Design your game's progression curves with this simple technique!* [Video]. YouTube. https://www.youtube.com/watch?v=OYWRd_EfeHY

Yamauchi et al. (2001). *Golden Sun* (Game Boy Advance version) [videojuego]. Camelot Software Planning.

Jiang et al. (2023). *Honkai: Star Rail* (Windows version) [videojuego]. HoYoverse.

Mayles et al. (2000). *Banjo-Tooie* (Nintendo 64 version) [videojuego]. Rare Limited.

8.2.1 Bibliografía utilizada por Cristina Rodríguez Rodríguez

Para la búsqueda de referencias para los diseños de interfaces, se utilizó: *Interface In Game* (<https://interfaceingame.com/>)

Para diversas consultas relacionadas con el motor, se utilizó: Foro de *Unreal Engine* (<https://dev.epicgames.com/community/unreal-engine>)

Unreal Geams. (n.d.). *Home [YouTube channel]*. *YouTube*. Recuperado el 22 de junio de 2023, desde <https://www.youtube.com/@UnrealGems>

Mathew Wadstein. (n.d.). *Home [YouTube channel]*. *YouTube*. Recuperado el 22 de junio de 2023, desde <https://www.youtube.com/@MathewWadsteinTutorials>

GomVoDeveloper. (n.d.). *Home [YouTube channel]*. *YouTube*. Recuperado el 22 de junio de 2023, desde <https://www.youtube.com/@GomVoTutoriales>

Tsaii et al. (2020). *Genshin Impact* (Mobile version). [videojuego]. HoYoverse.

Kaneko et al. (2020). *Pokemon Mystery Dungeon Rescue Team DX* (Nintendo Switch version) [videojuego]. Spike Chunsoft.

Rao et al. (2018). *Hades* (Windows version). [videojuego]. Supergiant Games.



UNIVERSIDADE DA CORUÑA



comunicación
audiovisual+3d+interactiva

