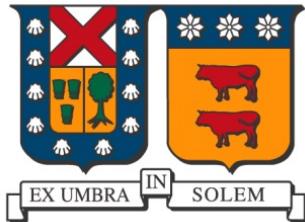


UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
SANTIAGO - CHILE



**"DESARROLLO DE HERRAMIENTAS DE APOYO PARA EL
DESARROLLO DE UN VIDEOJUEGO DE GÉNERO
CRAFTING"**

LUIS ALBERTO CORRALES PALMA

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN INFORMÁTICA

Profesor Guía: Sven von Brand
Profesor Correferente: Pedro Godoy

Julio - 2024

DEDICATORIA

Quiero dedicar este trabajo a mis padres, por su apoyo incondicional en cada paso de mi proceso educativo. Su amor y consideración me han permitido alcanzar este logro. A mi hermano, por sus bromas y por incentivar me a seguir adelante y terminar este proceso.

Al Yuyo y la Feña, por estar siempre, por su preocupación y ser un apoyo fundamental en los momentos más difíciles. Al Tiza, Vicho, Quality, Joaco, Agu, Tello, Jaz y todos los cabros, compañeros de la universidad y a todo aquel que en algún momento me ha brindado su compañía y apoyo.

A todos ustedes, gracias por ser parte de mi vida y por ayudarme a llegar hasta aquí.

AGRADECIMIENTOS

Quiero expresar mi más sincero agradecimiento a todas las personas que me brindaron su apoyo y contribuyeron al desarrollo de este trabajo.

A Sven, por darme la oportunidad de trabajar en Abstract Digital, realizar esta memoria y aprender tanto sobre el desarrollo de videojuegos. Agradecer a todo el equipo de Abstract Digital, por la experiencia enriquecedora y el privilegio de aportar a un proyecto tan hermoso como lo es “*Mix: the forgotten*”.

A Pedro, quien aceptó ser mi profesor correferente y siempre estuvo dispuesto a aclarar mis dudas durante nuestras reuniones.

Por supuesto, también quiero agradecer a cada persona que me brindó su apoyo incondicional: a mis padres, por su amor y respaldo constante; a mis hermanos, por su compañía y estímulo; y a todos mis amigos, quienes de una u otra manera estuvieron pendientes de mí y me ofrecieron su apoyo.

RESUMEN

Resumen— Este trabajo de titulación tiene como objetivo desarrollar herramientas que faciliten el desarrollo de videojuegos del género crafting en el motor de videojuegos Unity. El proyecto se centra en la creación de tres herramientas específicas: una para el diseño de niveles, otra para la importación de assets gráficos y una tercera para la creación de items crafteables. Estas herramientas buscan mejorar la eficiencia y la gestión del desarrollo de videojuegos, permitiendo a los desarrolladores centrarse más en la creatividad y la jugabilidad. Se utilizó una metodología ágil para el desarrollo iterativo y la validación de las herramientas, asegurando que cumplan con las necesidades de los desarrolladores y las tendencias actuales de la industria.

Palabras Clave— Videojuegos; Unity; Crafteo; Agilizar desarrollo; Herramientas de apoyo

ABSTRACT

Abstract— This degree project aims to develop tools that facilitate the development of crafting video games in the Unity game engine. The project focuses on the creation of three specific tools: one for level design, another for the import of graphical assets and a third for the creation of crafting items. These tools aim to improve the efficiency and management of video game development, allowing developers to focus more on creativity and gameplay. An agile methodology was used for iterative development and validation of the tools, ensuring that they meet the needs of developers and current industry trends.

Keywords— Video games; Unity; Crafting; Streamline development; Support tools

GLOSARIO

AR: Augmented Reality. Se traduce como Realidad aumentada.

ASMDEF: Abreviatura de Assembly Definition. Se refiere a un archivo de definición de ensamblado que permite organizar y administrar de manera más eficiente los scripts y recursos de un proyecto en Unity.

DI: Departamento de Informática.

FPS: Frames Per Second, se traduce como cuadros por segundo y se refiere a la medida de la velocidad a la que una imagen en movimiento se muestra en una pantalla.

GUI: Graphical User Interface. Se traduce como Interfaz gráfica de usuario.

IA: Inteligencia artificial. También puede encontrarse como AI (Artificial Intelligence).

IMGUI: Immediate Mode GUI. Se refiere a un enfoque de interfaz de usuario en Unity donde los elementos de la interfaz se dibujan y se actualizan directamente en cada frame de manera inmediata.

UI: User Interface. Traducido como Interfaz de Usuario.

UTFSM: Universidad Técnica Federico Santa María.

VR: Virtual Reality. Se traduce como Realidad virtual y puede encontrarse como RV.

XP: Tiene muchos significados dependiendo del contexto. Uno de esos es el acrónimo de Extreme Programming, que es una metodología ágil.

ÍNDICE DE CONTENIDOS

RESUMEN	IV
ABSTRACT	IV
GLOSARIO	V
ÍNDICE DE FIGURAS	IX
ÍNDICE DE TABLAS	XI
INTRODUCCIÓN	1
CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA	3
1.1 CONTEXTO	3
1.2 ÁRBOL DEL PROBLEMA	5
1.3 OBJETIVOS	6
1.3.1 OBJETIVO GENERAL	6
1.3.2 OBJETIVOS ESPECÍFICOS	6
1.4 ALCANCE DE LA SOLUCIÓN	7
CAPÍTULO 2: MARCO CONCEPTUAL	8
2.1 Videojuegos de género crafting	8
2.2 Desarrollo de videojuegos	9
2.3 Motor de videojuegos	9
2.4 Unity	9
2.4.1 Assets	10
2.4.2 Prefab	11
2.4.3 Editor Scripting	11
2.4.4 Scriptableobject	12
2.5 Tileset/Tilemap	13
2.5.1 Tileset	13
2.5.2 Tilemap	14
2.6 Metodologías de desarrollo de software	15
2.6.1 Metodologías tradicionales	15
2.6.2 Metodologías ágiles	16
CAPÍTULO 3: PROPUESTA DE SOLUCIÓN	17
3.1 Selección de la metodología de desarrollo	17
3.1.1 Scrum	17
3.1.2 Extreme Programming	18
3.1.3 Kanban	19

3.2 Planificación	20
3.2.1 Recolección de requisitos	20
3.2.2 Investigación-Estado del arte	20
3.2.3 Programación y desarrollo de las herramientas	21
3.3 Recolección de requisitos	21
3.4 Investigación-Estado del arte	22
3.4.1 Análisis de Soluciones Existentes	22
3.4.2 Estudio de Arquitecturas	23
3.5 Gestión de Tareas y Backlog del Proyecto	24
3.5.1 Sprint 1 - Herramienta Items crafteables	25
3.5.2 Sprint 2 - Herramienta Diseño de niveles	26
3.5.3 Sprint 3 - Herramienta importación de Assets gráficos	27
3.6 Documentación de las Herramientas propuestas en la solución	28
3.6.1 Herramienta de diseño de nivel	28
3.6.2 Herramienta importación de Assets/Nuevos ingredientes	32
3.6.3 Herramienta Items crafteables	37
CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN	42
4.1 Proceso de Validación mediante Testing de usuarios externos	42
4.1.1 Validación de la Herramienta de Diseño de Niveles	42
4.2 Resultados de las Encuestas de la Herramienta de Diseño de Niveles	43
4.3 Análisis y Discusión de la Herramienta de Diseño de Niveles	45
4.4 Conclusiones de la Validación de la Herramienta de Diseño de Niveles	46
4.5 Validación de las Herramientas de Importación de Assets Gráficos y Creación de Items Crafteables	47
4.5.1 Uso de la Solución en el Desarrollo de un Proyecto	47
4.5.2 Evaluación con experto	48
4.6 Resultados de la evaluación con el experto de las Herramientas de Importación de Assets Gráficos y Creación de Items Crafteables	49
4.6.1 Resultados de la Herramienta de Importación de Assets Gráficos	49
4.6.2 Resultados de la Herramienta de Creación de Items Crafteables	50
4.7 Análisis y Discusión de las Herramientas de Importación de Assets Gráficos y Creación de Items Crafteables	51
4.8 Conclusiones de la Validación de las Herramientas de Importación de Assets Gráficos y Creación de Items Crafteables	51
CAPÍTULO 5: CONCLUSIONES	53
5.1 Alcances y Limitaciones de la Propuesta de Solución	53
5.1.1 Alcances	53
5.1.2 Limitaciones	54
5.2 Resultados	54
5.2.1 Herramienta de Diseño de Niveles	55
5.2.2 Herramienta de Importación de Assets Gráficos	55
5.2.3 Herramienta de Creación de Objetos Crafteables	55

5.3 Validez de los Objetivos Formulados	55
5.4 Contribuciones y Aplicaciones	56
5.5 Impacto y Aporte	56
5.6 Recomendaciones y Futuras Ampliaciones	56
 ANEXOS	58
5.1 Imágenes de las pruebas realizadas con usuarios	58
5.1.1 Herramienta de diseño de niveles	58
5.1.2 Herramienta de importación de assets y creación de ingredientes	59
5.1.3 Herramienta de creación de nuevos objetos crafteables	60
5.2 Resultados de las encuestas - Herramienta de diseño de niveles	63
5.2.1 Gráficos	63
5.2.2 Dificultades del usuario al realizar tareas	67
5.2.3 Retroalimentación abierta	68
5.3 Detalles de las Mediciones de Tiempos en el uso de las herramientas	70
5.3.1 Herramienta de Importación de Assets	70
5.3.2 Herramienta de Creación de Objetos Crafteables	70
5.4 Enlaces	70
 REFERENCIAS BIBLIOGRÁFICAS	72

ÍNDICE DE FIGURAS

1	Softwares y motores más utilizados para el desarrollo de videojuegos.	4
2	Mercado de los videojuegos, ingresos por país.	5
3	Árbol del Problema.	6
4	Ejemplo de un sistema de crafteo.	8
5	Editor de Unity (motor de videojuegos).	10
6	Assets.	11
7	Ejemplo de un <i>ScriptableObject</i> usado para guardar el <i>item</i> de un videojuego. .	12
8	Ejemplo de Tileset.	14
9	Ejemplos de Tilemaps.	15
10	Metodología Kanban.	19
11	Bosquejo/idea de herramienta requerida.	22
12	ProBuilder (Unity).	23
13	Documentación en Notion	25
14	StyleManager - Prefab auxiliar de la herramienta.	29
15	StyleManager - Campos para definir opciones de terreno.	30
16	Interfaz inicial de la herramienta de diseño de niveles.	30
17	Interfaz inicial de la herramienta de diseño de niveles.	31
18	Menú de cargado/guardado del escenario - Herramienta de diseño de niveles. .	31
19	Menú de la grilla - Herramienta de diseño de niveles.	32
20	Menú de otros prefabs - Herramienta de diseño de niveles.	33
21	Escena de Unity y Herramienta de diseño de niveles	34
22	Menú inicial - Herramienta de importación de Assets gráficos y diseño de ingredientes	35

23	Modelo referencia instanciado - Herramienta de importación de Assets gráficos y diseño de ingredientes	35
24	Menú de ajustes - Herramienta de importación de Assets gráficos y diseño de ingredientes	36
25	Item component - Herramienta de importación de Assets gráficos y diseño de ingredientes	36
26	Transform component - Herramienta de importación de Assets gráficos y diseño de ingredientes	37
27	Menú creación de receta - Herramienta de Items crafteables	38
28	ScriptableRecepyp - Herramienta de Items crafteables	39
29	Ejemplo prefab Item crafteable - Herramienta de Items crafteables	40
30	Creación prefab del objeto crafteable - Herramienta de Items crafteables	41
31	Ejemplo de Prefab final del objeto crafteable - Herramienta de Items crafteables	41
32	Experiencia general de los usuarios testeados al usar la herramienta	44
33	Que tan intuitiva encontraron la herramienta los usuarios testeados	45
34	Comparación eficiencia usando la herramienta	45
35	Tarea 1 - Prueba herramienta diseño de niveles.	58
36	Tarea 2 - Prueba herramienta diseño de niveles.	58
37	Tarea 3 - Prueba herramienta diseño de niveles.	59
38	Tarea 1 - Prueba de la herramienta importación de assets gráficos y creación de nuevos ingredientes.	59
39	Tarea 2 - Prueba de la herramienta importación de assets gráficos y creación de nuevos ingredientes.	60
40	Tarea 3 - Prueba de la herramienta importación de assets gráficos y creación de nuevos ingredientes.	60
41	Tarea 3 - Prueba de la herramienta importación de assets gráficos y creación de nuevos ingredientes.	61

42 Tarea 3 - Prueba de la herramienta importación de assets gráficos y creación de nuevos ingredientes.	62
43 Tarea 3 - Prueba de la herramienta importación de assets gráficos y creación de nuevos ingredientes.	62
44 Edad de los usuarios testeados	63
45 Ocupación de los usuarios testeados	63
46 Experiencia desarrollando videojuegos con Unity	64
47 Experiencia general de los usuarios testeados al usar la herramienta	64
48 Facilidad al momento de aprender a usar la herramienta según los usuarios testeados	65
49 Usabilidad de la interfaz de la herramienta según los usuarios testeados	65
50 Problemas en el testeo por parte de los usuarios testeados	66
51 Que tan intuitiva encontraron la herramienta los usuarios testeados	66
52 Comparación eficiencia usando la herramienta	67
53 Recomienda el usuario la herramienta a otros desarrolladores de videojuegos . .	67

ÍNDICE DE TABLAS

1 Porcentaje de Éxito y Tiempos Promedio en la Realización de Tareas para la Herramienta de Diseño de Niveles	44
2 Tiempos de Realización de Tareas con y sin el Uso de las Herramientas	48
3 Porcentaje de Mejora en los Tiempos de Realización de Tareas al Usar las Herramientas	48
4 Porcentaje de Éxito y Tiempos en la Realización de Tareas para la Herramienta de Importación de Assets Gráficos	50
5 Porcentaje de Éxito y Tiempos en la Realización de Tareas para la Herramienta de Creación de Items Crafteables	50

INTRODUCCIÓN

El desarrollo de videojuegos ha evolucionado considerablemente en las últimas décadas, impulsado por avances tecnológicos y la creciente demanda de experiencias de entretenimiento interactivas y envolventes. En este contexto, Unity se ha consolidado como una de las plataformas más populares y versátiles para la creación de videojuegos, gracias a su accesibilidad, amplio conjunto de herramientas y comunidad de desarrolladores. Sin embargo, a pesar de sus muchas ventajas, los desarrolladores enfrentan desafíos significativos en términos de eficiencia y creatividad en el proceso de diseño y desarrollo de juegos.

A medida que los videojuegos se vuelven más complejos y detallados, los desarrolladores enfrentan dificultades para gestionar y optimizar sus flujos de trabajo. La necesidad de herramientas que faciliten el diseño de niveles, la importación de assets gráficos y la creación de items crafteables se vuelve crucial para mantener un desarrollo eficiente y enfocado en la creatividad.

El presente trabajo aborda esta problemática mediante el desarrollo de tres herramientas innovadoras para Unity, destinadas a facilitar y optimizar el flujo de trabajo en proyectos de videojuegos. Estas herramientas incluyen una para el diseño de niveles, otra para la importación de assets gráficos y una última para la creación de items crafteables. La herramienta de diseño de niveles permite a los desarrolladores crear y manipular mapas 3D de manera eficiente. La herramienta de importación de assets gráficos facilita la integración de elementos gráficos en el sistema de crafeo del juego. Finalmente, la herramienta de creación de items crafteables permite la creación de nuevos objetos crafteables y la generación automática de sus elementos fundamentales.

Para abordar el desarrollo de estas herramientas, se empleó una metodología ágil, permitiendo flexibilidad y mejora continua a lo largo del proyecto. Para la herramienta de diseño de niveles, se realizaron pruebas de usabilidad con usuarios diversos para validar su funcionalidad y facilidad de uso. Las otras dos herramientas fueron evaluadas por desarrolladores profesionales de videojuegos para asegurar su eficacia y relevancia en entornos de desarrollo reales. Además, se tuvo en cuenta el uso de estas herramientas por desarrolladores de videojuegos en un proyecto concreto, lo que proporcionó valiosas métricas y feedback del equipo de desarrollo.

Inicialmente, se introduce el problema y se contextualiza la necesidad de desarrollar estas herramientas. Luego, se presenta la revisión del estado del arte y la investigación realizada sobre herramientas similares y metodologías relevantes. Posteriormente, se describe la metodología empleada para el desarrollo de las herramientas, seguida del diseño e implementación de cada una de las herramientas desarrolladas. A continuación, se discuten los resultados obtenidos y se realiza un análisis de las pruebas de usabilidad. Finalmente, se exponen las conclusiones y recomendaciones para futuras investigaciones y desarrollos.

A través de este trabajo, se busca no solo proporcionar herramientas prácticas para el desa-

rrollo de videojuegos en Unity, sino también contribuir al conocimiento y documentación en el área de desarrollo de herramientas para esta plataforma, estableciendo una base sólida para futuras mejoras y expansiones.

CAPÍTULO 1

DEFINICIÓN DEL PROBLEMA

1.1. CONTEXTO

La industria de los videojuegos está en constante crecimiento, y se ha transformado en una de las que más público y dinero mueve. En los últimos años, ha alcanzado a las principales industrias del entretenimiento como la música, los deportes y el cine. Por lo anterior, el desarrollo de los videojuegos ha tomado una gran importancia. Este auge se ha traducido en un crecimiento de la industria dedicada al desarrollo de videojuegos, de forma que cada vez aparecen más empresas que se dedican a este ámbito. Sin embargo, insertarse en el mercado de los videojuegos es bastante difícil y está lejos de ser una simple mina de oro. Es una realidad que la gran mayoría de los videojuegos no terminan prosperando.

El desarrollo de videojuegos se considera un proceso complicado debido a su alta curva de aprendizaje, ya que aprender implica invertir mucho tiempo [ComputerCareers, 2022], incluso para aquellas personas dedicadas a la programación. Además, no se reduce solamente a programar o diseñar el videojuego, más bien es un proceso complejo y que requiere de mucho tiempo. Los equipos de trabajo que están detrás del desarrollo de un videojuego la mayoría de las veces están conformados por un diverso grupo que no solo comprende programadores y diseñadores de videojuegos, sino que suele estar conformado por Artistas gráficos, músicos, productores y demás personas, por lo que surgen diversas problemáticas, ya que muchas veces personas que no tienen un dominio de las herramientas utilizadas en el desarrollo del videojuego se ven involucrados de alguna forma en el desarrollo del videojuego. Todo esto termina generando diversas dificultades en el equipo de trabajo involucrado en el desarrollo de videojuegos, debido al poco entendimiento entre las tareas realizadas por el resto del equipo, además de los errores y trabas que se presentan al momento de integrar el trabajo realizado por los distintos integrantes que se desempeñan en el desarrollo del videojuego.

Existen diversas herramientas y motores para el desarrollo de videojuegos, siendo *Unity* el más popular con una amplia diferencia por sobre las demás herramientas (figura 1). Si bien es el software más popular, sigue siendo una herramienta que requiere de conocimientos de desarrollo de videojuegos, programación y demás experiencia. *Unity* está basado en la programación en C#, es decir, programación orientada a objetos, por lo que la utilización de *Unity* implica estar dedicado a programar scripts constantemente.

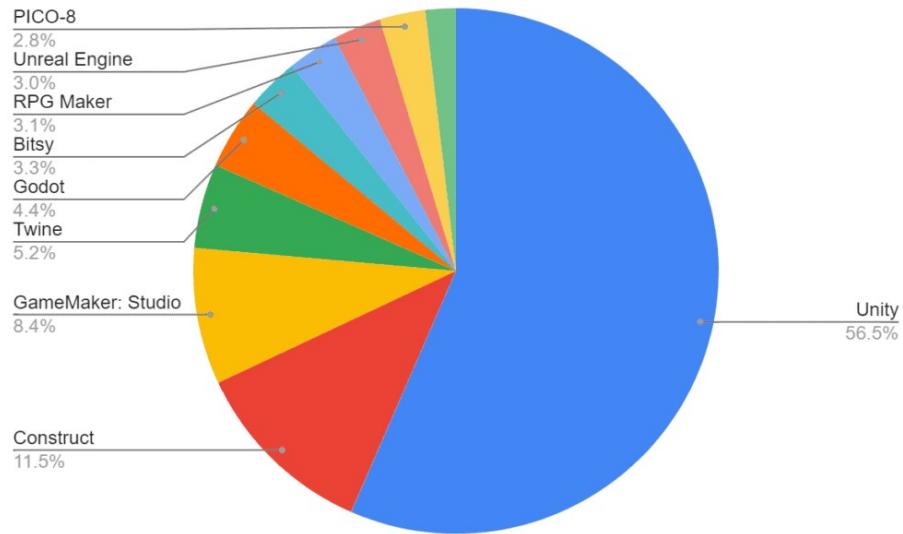


Figura 1: Softwares y motores más utilizados para el desarrollo de videojuegos.

Fuente: [Skoczyłas, 2022].

Por ello, siempre es bienvenida la implementación de nuevos mecanismos, prácticas o herramientas que faciliten el desarrollo de videojuegos, ya sea mediante herramientas de programación visual, reutilización de recursos, etc. El desarrollo de videojuegos puede resultar algo complejo, por lo que agilizar y acelerar el flujo de trabajo de los desarrolladores puede ayudar a reducir costos, errores y tiempo empleado por los trabajadores, lo que es muy valioso en una industria como la de los videojuegos. Por lo tanto, se propone la implementación de nuevos mecanismos y herramientas de *Unity* (el motor de videojuegos más popular) que sean fáciles de utilizar y proporcionen facilidades a aquellos involucrados en el desarrollo de un videojuego, de forma que sean un verdadero apoyo y puedan acelerar el flujo de trabajo del equipo de desarrollo.

La gran mayoría de las empresas dedicadas al desarrollo de videojuegos no generan grandes ganancias o directamente no recuperan la inversión inicial del proyecto. En Chile y el resto de Latinoamérica, esta es una industria deficiente, poco desarrollada y en la que no existen demasiados recursos (en la figura 2 se puede ver que Latinoamérica se encuentra muy por detrás con respecto al mercado global de videojuegos). La mayoría de las empresas tienen la necesidad de implementar nuevas tecnologías, herramientas y prácticas que les permitan automatizar u optimizar su trabajo.



Figura 2: Mercado de los videojuegos, ingresos por país.
Fuente: Statista Digital Market Outlook.

1.2. ÁRBOL DEL PROBLEMA

A continuación, se presenta el árbol del problema (figura 3), donde se observa más gráficamente el problema central abordado, junto a las causas y los efectos de este:

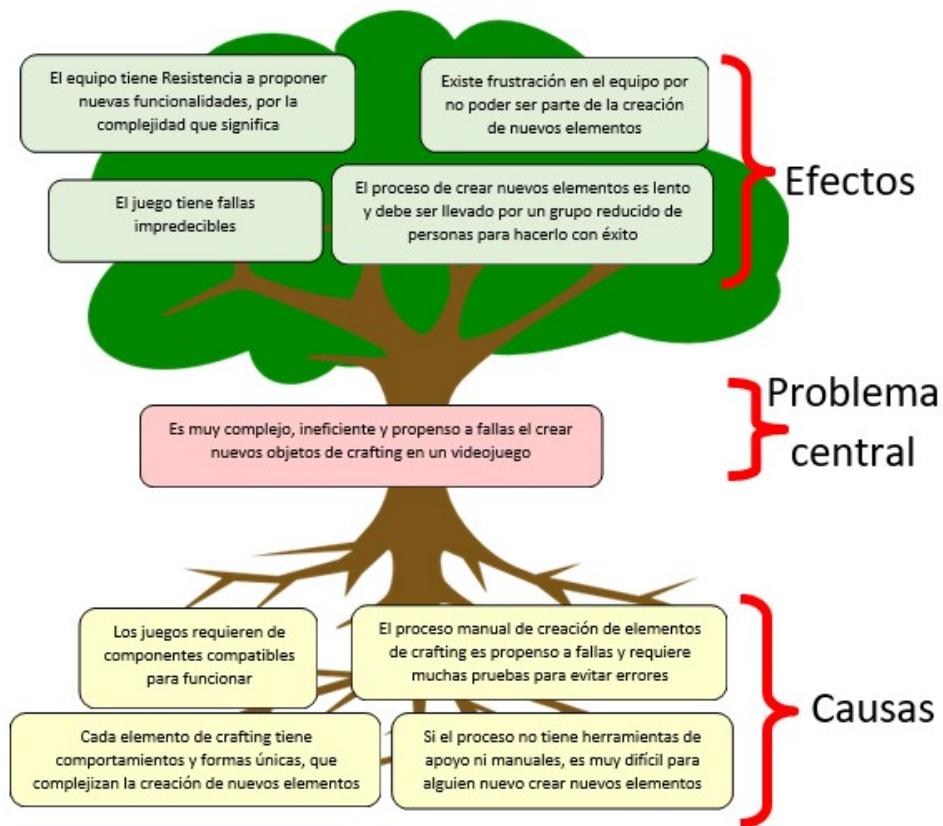


Figura 3: Árbol del Problema.
Fuente: Elaboración propia.

1.3. OBJETIVOS

1.3.1. OBJETIVO GENERAL

Crear herramientas de apoyo para la creación de assets gráficos y funcionales para juegos del género *crafting* en Unity.

1.3.2. OBJETIVOS ESPECÍFICOS

Para cumplir con el objetivo general planteado, es necesario realizar los siguientes objetivos específicos:

1. Crear un proceso que no requiera programación ni interacciones complejas con Unity para verificar la correcta creación de assets gráficos.
2. Desarrollar herramienta de construcción de niveles/mapas de un juego.

3. Desarrollar herramienta de asistencia de combinación de elementos para creación de herramientas en un juego de género *crafting*.
4. Validar la solución en base al desempeño y la satisfacción de los usuarios trabajando con la propuesta.

1.4. ALCANCE DE LA SOLUCIÓN

La propuesta de solución para las problemáticas planteadas, en un principio se realizará enfocada en el videojuego “Mix the forgotten”, el cual se encuentra en desarrollo por parte de la empresa Abstract Digital, de forma que esta solución signifique un apoyo para agilizar el trabajo del equipo de desarrollo en la elaboración del juego, facilitando así la reducción de los tiempos de integración y la cantidad de errores que se generan. De este modo se tiene un usuario definido para luego realizar la validación de la propuesta en base a la experiencia de los usuarios con esta nueva solución.

Sin embargo, tal como se puede deducir del título de esta memoria, se busca llegar a una solución reutilizable en el desarrollo de cualquier videojuego de género *crafting*. Por lo que el desarrollo de esta memoria es del interés de cualquier persona o empresa ligada al desarrollo de videojuegos, especialmente de género *crafting*. Un aspecto clave considerado al momento de proponer una solución, fue el alcance de esta, por lo que se optó por el desarrollo de herramientas en *Unity*, que como se mencionó previamente es el motor de videojuegos más popular en el desarrollo de estos. Por esto, sería bueno también un proceso de validación con usuarios externos a Abstract Digital.

CAPÍTULO 2

MARCO CONCEPTUAL

2.1. Videojuegos de género crafting

El género *crafting* refiere a aquellos videojuegos que contemplan un sistema de *crafteo*¹, que trata la acción de combinar distintos *items* o recursos para hacer un nuevo objeto.

El *crafting* empezó como una mecánica que se veía en algunos juegos de rol, pero a día de hoy se ha expandido hasta el punto en que está presente en la mayoría de los juegos modernos. Actualmente abundan ejemplos de juegos populares que poseen un sistema de *crafteo*, sobre todo en aquellos juegos con modalidad de supervivencia² como en el *Minecraft*, *Rust*, *The Forest*, etc.

Uno de los videojuegos con sistema de *crafteo* más icónicos es el ya mencionado *Minecraft*. A continuación se tiene una imagen de ejemplo, donde se puede ver que con la combinación de ciertos objetos se genera un nuevo objeto (Figura 4).



Figura 4: Ejemplo de un sistema de crafteo.

Fuente: Minecraft.

¹Con frecuencia se habla de *craftear* o *crafteo*, siendo esta una españolización de la palabra de origen inglés *crafting*.

²Los juegos de supervivencia son un subgénero de videojuegos de acción, en los que el jugador empieza con un equipamiento mínimo y para sobrevivir tiene que craftear herramientas y recolectar recursos.

Si bien no existe una clasificación estándar de estos sistemas de crafteo, ya que cada juego lo implementa acorde a la arquitectura o características del videojuego, se pueden reconocer ciertos enfoques [King, 2015] que suelen ir acorde al género del videojuego.

2.2. Desarrollo de videojuegos

El desarrollo de videojuegos trata del proceso de creación de un videojuego por un desarrollador y su equipo. Comprende desde un concepto o idea inicial hasta el videojuego ya finalizado. Es un proceso largo y complejo, que involucra profesionales de diversas áreas. Un equipo de desarrollo de videojuegos suele comprender un variado grupo de profesionales que van desde programadores, diseñadores gráficos, ingenieros de sonido, etc.

2.3. Motor de videojuegos

Un motor de videojuegos (en inglés *Game engine*) hace referencia a una serie de librerías y rutinas de programación que permiten el diseño, creación y funcionamiento de videojuegos [Carrasco, 2018]. Los motores de videojuegos buscan hacer más fácil y rápido el desarrollo de un videojuego, usando una interfaz gráfica de usuario (GUI)³ (figura 5) que ayude con el desarrollo gráfico del videojuego [freeCodeCamp, 2019].

Los motores de videojuegos proporcionan herramientas de desarrollo visual y permiten a los desarrolladores reutilizar *software* y funcionalidades que pueden ir desde el *renderizado*⁴ de gráficos 2D y 3D, un motor físico que simula las leyes de la física (usualmente para la detección de colisiones), animación, *scripting*, sonidos, inteligencia artificial (IA) y mucho más.

2.4. Unity

Existen diversos motores de videojuegos, unos más intuitivos que otros, unos más complejos, unos requieren más conocimientos de programación que otros, otros ofrecen mayor soporte gráfico, y así. Pero para el desarrollo de esta memoria se centrará en *Unity*, el que es con diferencia el motor de videojuegos más popular (Figura 1). *Unity* es un motor de videojuegos fácilmente accesible y de amplio soporte, ya que permite el desarrollo de juegos 2D, 3D, realidad aumentada (AR) y realidad virtual (VR), además de diversas funcionalidades y características necesarias para las distintas categorías de videojuego.

³GUI: *Graphic user interface*, traducido al español como Interfaz gráfica de usuario.

⁴Anglicismo que viene del término *rendering* que se refiere al proceso en que se genera una imagen partir de un modelo 2D o 3D, por medio de la utilización de programas informáticos.

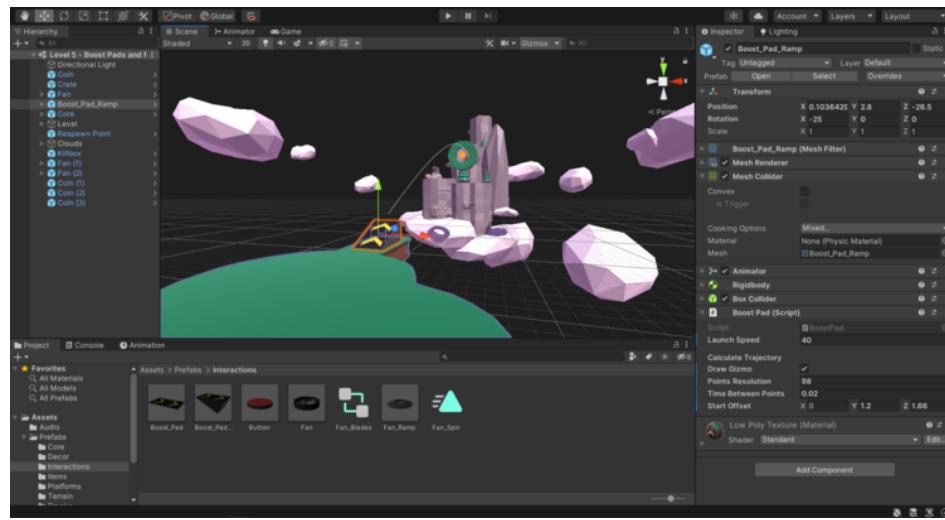


Figura 5: Editor de Unity (motor de videojuegos).

Fuente: Mastery coding.

Unity es un motor de videojuegos muy completo, que ofrece múltiples facilidades y elementos, a continuación se tienen algunos de estos elementos, objetos o herramientas, los cuales son esenciales para el desarrollo de esta memoria y la solución del problema.

2.4.1. Assets

Un Asset es cualquier recurso que se puede usar en el juego o proyecto. Los Assets van desde arte gráfico (como modelos 2D y 3D), hasta música, efectos de sonido y cualquier elemento que se importe al proyecto para ser utilizado en el editor de Unity para el desarrollo del videojuego (figura 6).

Los Assets son un recurso fundamental en el desarrollo de videojuegos. Los modelos gráficos, soundtrack⁵, y demás elementos que forman parte del desarrollo del videojuego son clave para el éxito de un videojuego. El manejo de Assets conecta el trabajo de los diseñadores gráficos, los músicos y demás artistas con el proyecto, la programación y diseño del videojuego.

⁵El concepto de soundtrack o banda sonora hace referencia a la parte sonora del proyecto (en este caso un videojuego), y puede estar compuesta por efectos de sonido, música, diálogos, etc.

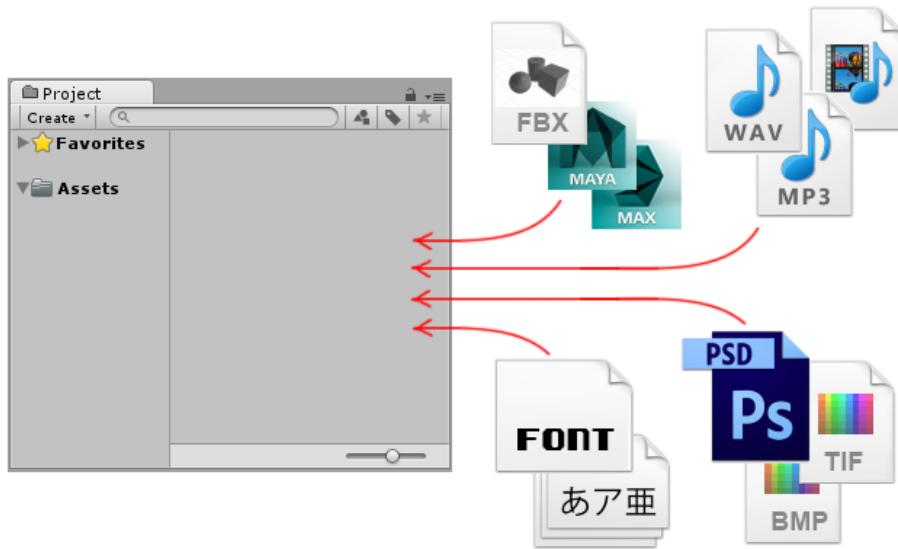


Figura 6: Assets.
Fuente: Unity3d.

2.4.2. Prefab

Unity ofrece la posibilidad de crear objetos preconfigurados reutilizables, conocidos como *Prefabs*, que actúan como plantillas para crear nuevas instancias del objeto en la escena del videojuego. Estos *prefabs* proveen una mayor flexibilidad y eficiencia del flujo de trabajo, ya que permiten reutilizar dichos objetos previamente creados [Hussain et al., 2020], evitando así tener que crear desde cero el mismo tipo de objeto. De este modo, uno puede almacenar objetos creados dentro de la escena, junto a sus componentes y propiedades, creando un *prefab* que viene siendo un tipo de *asset*. La mayoría de los paquetes de *assets* que se encuentran disponibles en el catálogo de *Unity* contienen *prefabs*.

2.4.3. Editor Scripting

En *Unity* existe la posibilidad de crear y publicar herramientas o extensiones para el editor, las cuales suelen facilitar la programación dentro del editor de *Unity* [KinematicSoup, 2016], por lo que existe un inmenso abanico de herramientas auxiliares a disposición. La programación de estas herramientas tiene como nombre *Editor Scripting*. Si bien existe un extenso catalogo de herramientas disponibles para importar en el editor de *Unity*, lamentablemente no existe demasiada documentación sobre el desarrollo de herramientas por parte de *Unity*.

En la *Assetstore*⁶ se encuentran herramientas de *Unity* de todo tipo (en su mayoría herra-

⁶Catálogo de *Unity* en el que se pueden obtener los distintos paquetes de *Assets*.

mientas de pago) que sirven para facilitar el desarrollo de los videojuegos dentro del editor, facilitando la implementación de múltiples funcionalidades para creación de niveles, importación de modelos gráficos, etc. En el caso de esta memoria, se busca desarrollar herramientas que permitan agilizar procesos dentro del desarrollo de un videojuego de género *crafting*, para lo cual será clave el *Editor Scripting*.

2.4.4. ScriptableObject

Los *ScriptableObject* son una clase en *Unity* que permite almacenar grandes cantidades de datos compartidos y que son independiente de los *scripts* instanciados [Hipple, 2020]. Son parte importante para el desarrollo de un videojuego, dado que ofrece ciertas facilidades para el manejo de la memoria. El uso de esta clase hace el juego más dinámico y son una forma eficiente de almacenar datos [Kunal, 2018], ya que funcionan como un contenedor de datos al que se accede por medio de referencias, evitando así el almacenar datos duplicados [Jin, 2020] y reducir el uso de la memoria.

Estos son usualmente usados para almacenar configuraciones, estadísticas de perfiles e *items* de inventario [French, 2022], lo que se relaciona fuertemente con los videojuegos de género *crafting* que se basan en la creación y fabricación de herramientas, objetos o *items* en general. Figura 7. Los sistemas de *crafeo* que poseen los videojuegos están basados en arquitecturas que implementan estos objetos bajo la utilización de los propios *ScriptableObjects* [Serpa, 2022].

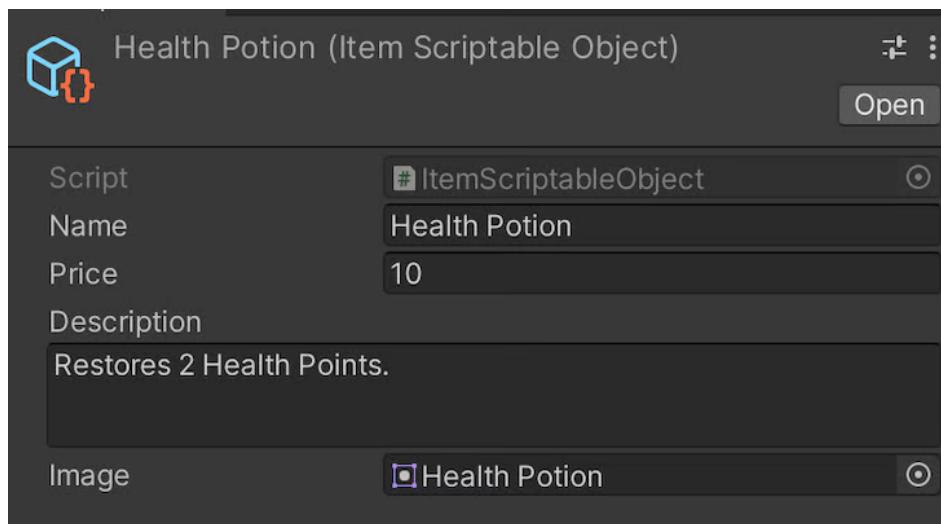


Figura 7: Ejemplo de un *ScriptableObject* usado para guardar el *item* de un videojuego.
Fuente: LogRocket.

2.5. Tileset/Tilemap

Técnicas utilizadas en el desarrollo de videojuegos, especialmente en juegos 2D, comúnmente basados en el pixel art. Ambos son conceptos o técnicas que ayudan a construir escenarios mediante un proceso sencillo, ágil y capaz de hacerle la vida más fácil a los diseñadores de videojuegos. Esta forma de diseñar escenarios o niveles en los videojuegos es una de las técnicas más usadas hasta el día de hoy, de hecho se ha convertido en la base fundamental de algunos de los más grandes clásicos videojuegos como Super Mario Bros, Stracraft, Pacman, Pokemon, etc.

2.5.1. Tileset

Un *tileset*, también conocido como set de teselas, consiste en un conjunto de imágenes tex- turizadas dispuestas en una sola composición visual. Estas imágenes texturizadas son las unidades gráficas que conforman el entorno visual de un videojuego, abarcando elementos como suelos, paredes, escaleras, techos y más8.

La creación de este conjunto implica la organización de las imágenes en una cuadrícula compuesta por cuadrados de dimensiones uniformes, en este caso, 16x16 píxeles. Cada uno de estos cuadrados en la cuadrícula recibe el nombre de *tile* o *tesela*, representando la unidad visual más pequeña dentro de la composición.

Esta colección de componentes (o texturas) debe ser organizada de manera que permita distinguir objetos como una pared, un toldo, múltiples puertas, una sección de suelo y un par de áreas con colores sólidos, por ejemplo. Estos elementos servirán como base para construir el entorno visual.

El tileset desempeña un papel sumamente significativo en el desarrollo de un videojuego, ya que es el archivo que actúa como guía a lo largo de todo el proceso de creación, facilitando la tarea al permitir la construcción modular de los escenarios. En consecuencia, contribuye a reducir el tiempo necesario para el desarrollo.

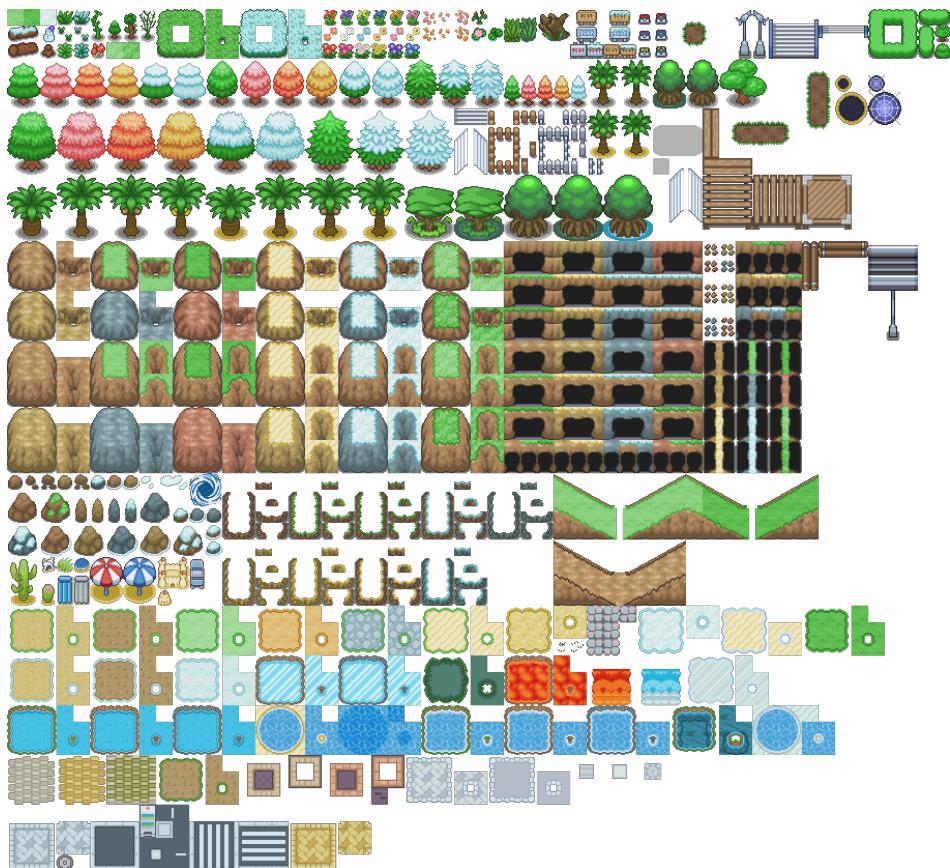


Figura 8: Ejemplo de Tileset.
Fuente: Deviant Art, @Flurmimon.

2.5.2. Tilemap

El *tilemap*, también conocido como mapa de teselas o mosaico, representa el lienzo en el cual se despliegan las teselas gráficas. En la terminología de los videojuegos, se prefiere el término *mapa* en lugar de *escenario*, ya que es más común y preciso en este contexto.

Es esencial comprender que, empleando un único conjunto de teselas gráficas (tileset), es posible generar una variedad infinita de mapas únicos⁹. La efectividad del diseño del tileset es crucial, ya que incide directamente en la amplitud de las opciones creativas que se pueden explorar.

Algunas ventajas prácticas de los tilesets y los tilemaps en el proceso de desarrollo de juegos:

- Aceleran la creación de niveles al utilizar un conjunto limitado de imágenes para construir diversas áreas.
- Optimizan el manejo de archivos al reducir la cantidad de imágenes individuales que se deben cargar en la memoria del juego.

- Simplifican la tarea de crear una estética visual cohesiva y elegante al requerir menos activos gráficos para lograr un resultado visualmente atractivo.

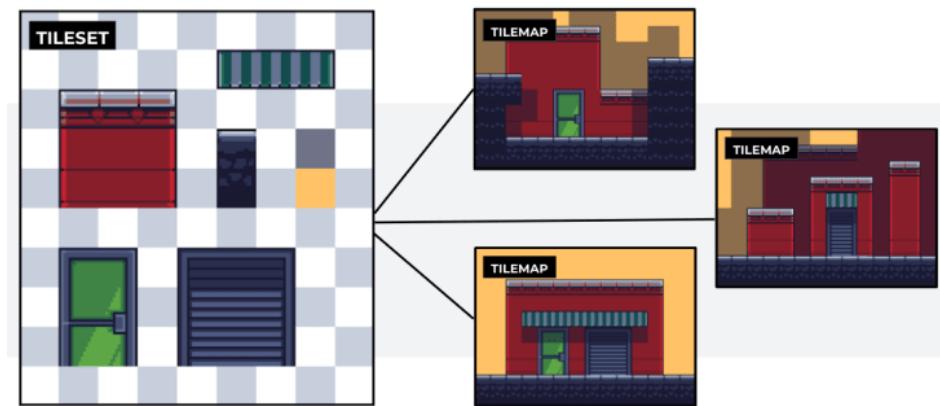


Figura 9: Ejemplos de Tilemaps.

Fuente: [Cruscuolo, 2021].

2.6. Metodologías de desarrollo de software

El desarrollo de esta memoria involucra de pleno el desarrollo de software para llevar adelante la propuesta de solución que se busca elaborar. En el desarrollo de software, es fundamental adoptar una metodología que proporcione un marco estructurado para el proceso de creación de aplicaciones y sistemas. Estas metodologías permiten establecer prácticas y enfoques específicos para gestionar el desarrollo de software de manera efectiva, teniendo en cuenta factores como los requisitos del proyecto, el tiempo, los recursos y la colaboración entre los miembros del equipo.

Existen diferentes enfoques para el desarrollo de software, que se pueden clasificar en dos categorías principales: metodologías tradicionales y metodologías ágiles.

2.6.1. Metodologías tradicionales

Las metodologías tradicionales, también conocidas como enfoques de desarrollo en cascada o en fase secuencial, se basan en un modelo de desarrollo lineal y planificado. Estas metodologías enfatizan la planificación exhaustiva y la documentación detallada antes de iniciar la implementación. Algunos ejemplos de metodologías tradicionales son el Modelo en Cascada y el Modelo en V.

En el modelo en cascada, el desarrollo se divide en fases secuenciales y lineales, como el análisis de requisitos, el diseño, la implementación, las pruebas y el mantenimiento. Cada

fase debe completarse antes de pasar a la siguiente, y los cambios posteriores pueden ser costosos y complicados de manejar.

2.6.2. Metodologías ágiles

Las metodologías ágiles, por otro lado, surgieron como respuesta a las limitaciones de los enfoques tradicionales. Estas metodologías se centran en la adaptabilidad, la colaboración y la entrega continua de incrementos funcionales de software. Los equipos ágiles valoran la comunicación directa, la retroalimentación temprana y la flexibilidad para ajustar los requisitos y las funcionalidades a medida que avanza el proyecto.

- **Scrum** es una de las metodologías ágiles más populares y ampliamente utilizadas. Se basa en ciclos de desarrollo iterativos y repetitivos llamados *sprints*, que generalmente tienen una duración de 1 a 4 semanas. Durante cada sprint, se planifican, desarrollan, prueban y entregan incrementos de software funcionales. Scrum se caracteriza por su enfoque en roles específicos, como el Scrum Master, el Product Owner y el Equipo de Desarrollo, y por sus eventos regulares, como las reuniones diarias de seguimiento (daily stand-ups), las revisiones de sprint y las retrospectivas.
- **Kanban** es otra metodología ágil que se centra en la visualización del flujo de trabajo y la optimización del mismo. Se utiliza un tablero Kanban para representar las tareas y su estado actual, lo que permite a los miembros del equipo tener una visión clara de las actividades en curso, las tareas pendientes y las completadas. Kanban se basa en principios como limitar el trabajo en progreso, maximizar el flujo y mejorar continuamente el proceso. A diferencia de Scrum, Kanban no se basa en sprints y permite un flujo de trabajo más continuo y flexible.
- **Extreme Programming (XP)** es una metodología ágil que se centra en la entrega rápida y continua de software de alta calidad. XP se basa en una serie de prácticas y valores clave que promueven la colaboración, la adaptabilidad y la retroalimentación constante.

CAPÍTULO 3

PROPUESTA DE SOLUCIÓN

En esta memoria para desarrollar una solución se trabajará en la elaboración de herramientas de apoyo para el desarrollo de videojuegos de género crafting, y para ello se tendrán en cuenta las necesidades reconocidas anteriormente en la definición del problema. Para recoger necesidades reales de la gente que se ve involucrada y afectada por el problema identificado será de utilidad trabajar visualizando los requerimientos que se pueden recoger de un equipo de desarrollo de videojuegos que trabaja en un videojuego de género crafting, como lo es Abstract Digital, quienes están desarrollando el videojuego “Mix the forgotten”.

3.1. Selección de la metodología de desarrollo

Para el desarrollo de la solución propuesta en esta memoria se optará por elegir una metodología de desarrollo iterativo e incremental, así poder ir desarrollando el software solución en ciclos iterativos y entregas incrementales, lo que permite desarrollar las herramientas de apoyo de forma flexible adaptándose a *feedback*, nuevas necesidades que se reconozcan y mejoras continuas.

Se piensa adoptar una mezcla de la metodología Scrum, Extreme Programming y Kanban, utilizando Scrum para la planificación y la entrega iterativa de funcionalidades, pero adoptando aspectos fundamentales de XP como la adaptabilidad y la retroalimentación constante, mientras que Kanban puede ser útil para visualizar el flujo de trabajo y gestionar eficientemente las tareas en cada etapa del desarrollo. Aunque sea típicamente utilizado en entornos de desarrollo de software en equipos, los principios y prácticas de Scrum se pueden adaptar y aplicar a proyectos individuales.

3.1.1. Scrum

Para aplicar la metodología Scrum al desarrollo del software solución que se propone en esta memoria, en primer lugar se establece que quien desarrolla la memoria asume los diferentes roles de Scrum por sí mismo, actuando así como Product Owner, Scrum Master y Equipo de Desarrollo, encargándose de definir los objetivos y metas de la tesis, identificando los requisitos y necesidades que deben ser abordados. Se debe asegurar de seguir los principios y prácticas de Scrum, adaptándolos al contexto individual de esta memoria.

Aunque no se gestione un Product Backlog extenso, se aplicaría la idea de tener una lista de tareas o requisitos prioritarios para el software solución de la presente memoria. Esto ayuda a mantener un seguimiento de las actividades que se necesitan realizar y a establecer prio-

ridades. Los elementos más críticos se identifican y organizan en función de su importancia y dependencias. A partir de esta lista, se planifican los sprints, períodos de tiempo definidos en los cuales se focalizaría en un conjunto específico de tareas.

Durante cada sprint, se trabaja de forma iterativa y se logran entregas incrementales del software o herramientas de apoyo propuestas como solución. Mediante reuniones de seguimiento periódicas, se revisa el progreso realizado y se realizan adaptaciones en función del feedback. Estos encuentros permiten evaluar el avance, identificar obstáculos y cambios necesarios, y establecer nuevos objetivos para el siguiente sprint. En el caso de esta memoria se propone el desarrollar tres herramientas de apoyo, y se planea separar el desarrollo de cada herramienta a un sprint distinto, además de que en cada uno de estos sprint se desarrolle de manera incremental, implementando nuevas funcionalidades, además de poder realizar entregas incrementales y recibir feedback regularmente para mejorar el trabajo realizado.

Al concluir cada sprint, se lleva a cabo una revisión, en la cual se evaluaron los resultados obtenidos y se reflexiona acerca de los logros alcanzados y las lecciones aprendidas. Asimismo, se realiza una retrospectiva para identificar oportunidades de mejora en el enfoque utilizado y ajustar las prácticas para futuros sprints.

3.1.2. Extreme Programming

En esta memoria se llevó a cabo lo que se conoce como desarrollo iterativo e incremental, ya que se busca desarrollar una herramienta a la vez, siguiendo un enfoque incremental y los principios del XP.

Para el desarrollo de software que se contempla en esta memoria como se ha aclarado con anterioridad, será fundamental la retroalimentación y la comunicación con personal involucrado en el desarrollo de videojuegos al momento de identificar necesidades y recoger requisitos. Esto va acorde a los principios de Extreme Programming (XP), ya que XP promueve una comunicación constante y abierta entre el equipo de desarrollo y el cliente, solicitando feedback y realizando ajustes en función de las necesidades.

XP pone énfasis en la realización de pruebas unitarias para garantizar la calidad del código, y en este caso se incorporarán *pruebas unitarias*⁷ en el desarrollo de las herramientas y se busca realizar *refactoring*⁸ para mejorar la calidad del código.

XP fomenta la integración continua, es decir, la integración frecuente y regular del código desarrollado, y en esta memoria se busca seguir un enfoque de integración continua al incorporar las nuevas funcionalidades que se vayan integrando en cada herramienta.

⁷Las pruebas unitarias son un tipo de prueba en el desarrollo de software que se centran en verificar el comportamiento y la funcionalidad de unidades individuales de código, como funciones, métodos o clases.

⁸El refactoring o refactorización de código tiene como objetivo mejorar la facilidad de comprensión del código o cambiar su estructura y diseño y eliminar código muerto, para facilitar el mantenimiento en el futuro.

3.1.3. Kanban

Se planea implementar la metodología Kanban para el desarrollo de las herramientas de apoyo ideadas como la propuesta solución. Kanban se centra en visualizar y optimizar el flujo de trabajo, limitando en progreso y priorizando las tareas de manera eficiente.

Para el desarrollo de esta memoria se planea desarrollar una herramienta a la vez y luego dejar a disposición de ser testeadas. Este enfoque de desarrollo incremental y por etapas va acorde a los principios de Kanban. Al completar una herramienta, esta pasa a un estado de *Hecho* o *Completado*, mientras que las herramientas que aun se encuentran en desarrollo o las que están siendo testeadas se encuentran en un estado de *En progreso*. La idea es que a medida que se termine un sprint, la herramienta asociada a dicho sprint ahora se deje a disposición para que esta pueda ser testeada, permitiendo así la retroalimentación y la mejora continua.



Figura 10: Metodología Kanban.

Fuente: Tecno-Soluciones.

3.2. Planificación

Para llevar a cabo el desarrollo de esta memoria se idea una planificación, de forma que se tenga un plan claro para el desarrollo de la solución, reconociendo distintas etapas y procesos involucrados en el desarrollo de esta memoria, junto a un tiempo estimado para llevar a cabo cada una de estas etapas.

3.2.1. Recolección de requisitos

En un principio es necesario reconocer las necesidades y recolectar requisitos, para lo cual se recurre a las personas interesadas en el software que se busca desarrollar, es decir, el posible público interesado en el desarrollo de esta memoria. Para lo anterior es que en el desarrollo de esta memoria se ha tenido en cuenta las necesidades que plantean en el equipo de trabajo de Abstract Digital, quienes se encuentran desarrollando un videojuego de género *crafting*.

El proceso de recolección de requisitos/requerimientos es fundamental para el desarrollo de software y permite tener una idea de lo que se quiere desarrollar y poder empezar a idear el software que se desea desarrollar, identificando posibles funcionalidades y diseño.

En esta etapa, se trabaja en la comprensión de los requisitos del software y las necesidades de los usuarios. Para esto se pueden llevar a cabo entrevistas, reuniones y encuestas con los interesados para recopilar información sobre las características, funcionalidades y objetivos del software. En el caso de esta memoria se me permitió ser partícipe de las reuniones tipo *daily* (metodología Scrum) del equipo de trabajo de Abstract Digital empleado en el desarrollo de videojuegos, lo cual fue una instancia para recoger ideas y requerimientos directamente de los implicados e interesados en el desarrollo de esta memoria.

3.2.2. Investigación-Estado del arte

Se realiza una investigación acerca de software, herramientas o soluciones relacionadas al problema identificado en el desarrollo de esta memoria, es decir, revisar el estado del arte de este problema para conocer las soluciones, sus arquitecturas y algoritmos que se utilizan e implementan las soluciones existentes. La investigación del estado del arte constará de:

- Análisis de Soluciones Existentes
- Estudio de Arquitecturas

3.2.3. Programación y desarrollo de las herramientas

Basándose en los requisitos y en la investigación realizada, se idea y diseñan las herramientas. Esto abarca la definición de la arquitectura, la interfaz de usuario y la lógica de funcionamiento. Básicamente en esta etapa se traducen las ideas en el resultado final, desarrollando la interfaz de usuario, los algoritmos y la funcionalidad.

3.3. Recolección de requisitos

Para recolección de requisitos de la propuesta de solución se utilizaron distintas formas: en un principio se realizaron entrevistas con los afectados por el problema que se busca solucionar, además de reuniones previamente y durante el desarrollo de la solución que se propone. De esta forma se reconocen objetivos para el desarrollo de la solución, los cuales detallan requisitos/funcionalidades que se buscan implementar. Todo esto se encuentra en la aplicación multifuncional Notion que se decidió utilizar para organizar el desarrollo de esta memoria.

El primer acercamiento e intención de recolectar requisitos fue con entrevistas a desarrolladores de videojuegos que se encuentran desarrollando un videojuego de género crafting. Estas entrevistas permitieron encontrar formas de agilizar procesos en Unity y reconocer posibles funcionalidades que se quieran implementar en las herramientas que se buscan desarrollar para facilitar el desarrollo de un videojuego de género crafting en Unity.

En principio se realizaron varias reuniones con el usuario objetivo con el fin de recolectar ideas y necesidades para el posterior desarrollo de las herramientas de apoyo que se proponen en la solución. Como resultado de estas reuniones se identificaron requisitos y se bosquejaron ideas para las herramientas y su interfaz (Figura 11).

No solo se obtuvieron requisitos e historias de usuario en estas reuniones iniciales, sino que constantemente se realizaron reuniones con el equipo de desarrollo del videojuego “Mix: the forgotten” de modo que continuamente podían aparecer nuevos requisitos de acuerdo a las nuevas necesidades que se vayan identificando mientras se desarrollaban las herramientas. Por lo anterior es que es fundamental el implementar y diseñar herramientas totalmente flexibles y con una arquitectura que permita el añadir nuevas funcionalidades de forma fácil.

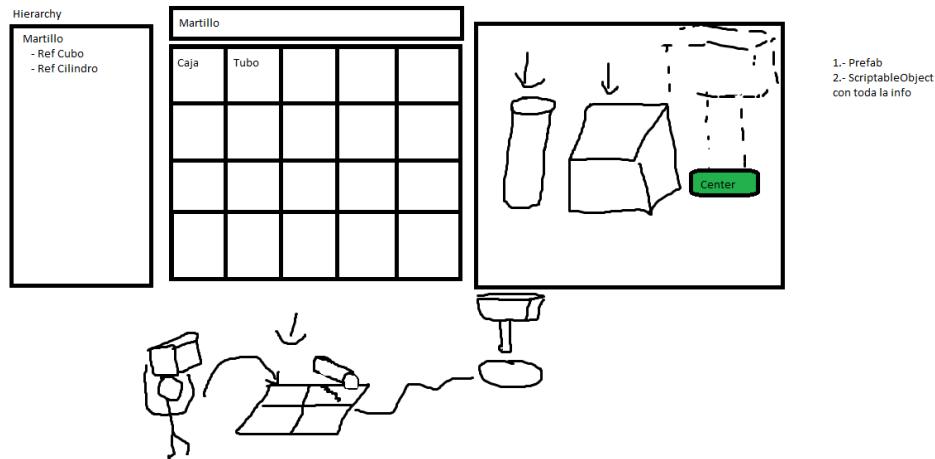


Figura 11: Bosquejo/idea de herramienta requerida.

Fuente: Elaboración propia.

3.4. Investigación-Estado del arte

3.4.1. Análisis de Soluciones Existentes

En el contexto de la investigación del estado del arte sobre el desarrollo de herramientas y extensiones para Unity, se observa una notable escasez de información y recursos disponibles. A diferencia de otras áreas de desarrollo de software, donde existen numerosos estudios, tutoriales y herramientas open source⁹, el ámbito específico del desarrollo de herramientas y extensiones para Unity presenta un panorama menos documentado y explorado.

La literatura y los recursos disponibles en línea sobre este tema son limitados, lo que sugiere una necesidad de investigación y documentación más amplia en este campo. La falta de información detallada y accesible puede plantear desafíos adicionales para los desarrolladores que buscan crear herramientas personalizadas o ampliar las capacidades de Unity a través de extensiones.

Durante la investigación del estado del arte, se indagó sobre extensiones de Unity y recursos relevantes que abordan problemas similares al que se aborda en este proyecto. Entre las herramientas más destacadas se encuentra **ProBuilder**, la cual es una herramienta dentro de Unity que combina funciones de modelado 3D y diseño de niveles, permitiendo a los desarrolladores crear geometría de forma rápida y eficiente para prototipos o proyectos finales en el entorno de Unity. ProBuilder brinda la capacidad de crear desde estructuras básicas hasta estructuras complejas, así como terrenos detallados, vehículos, armas y otros elementos visuales dentro del entorno de edición de Unity. Esto simplifica significativamente el desarrollo de juegos y aplicaciones interactivas al ofrecer una amplia gama de posibilidades de

⁹Open source: Modelo de desarrollo de software donde el código fuente es accesible al público para su visualización, modificación y distribución.

diseño y modelado.

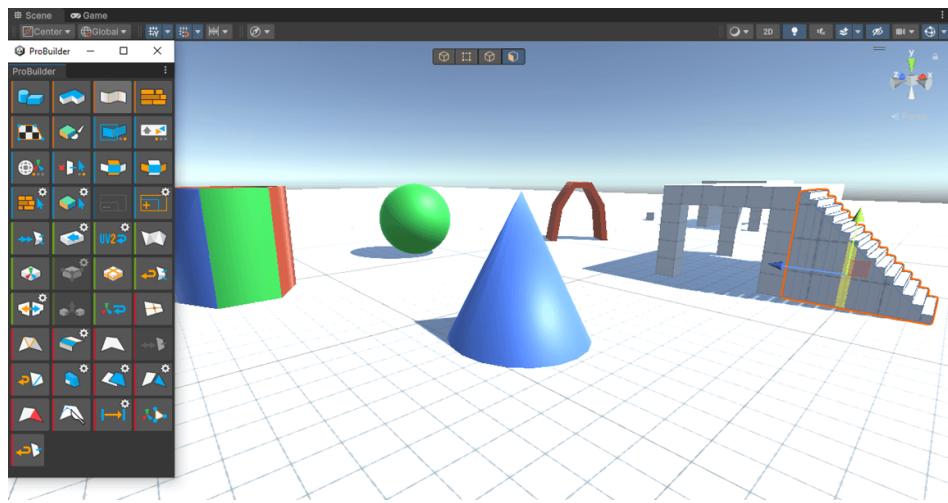


Figura 12: ProBuilder (Unity).

Fuente: nobreakpoints Blog.

Unity posee un mercado de extensiones disponibles para agregar al editor de Unity, donde se encuentran gran cantidad de herramientas aparte de ProBuilder. Sin embargo, no se encontró registro de alguna herramienta que permita implementar modelos predefinidos dentro del mismo proyecto, lo cual puede ser interesante de cara a la implementación de una nueva solución.

Otro Asset tomado en cuenta fue el *3D Tilemap - Unity Asset Store*. Este Asset disponible en la tienda de Unity ofrece una solución para la creación de tilemaps en entornos 3D, lo cual va relacionado con la solución que buscamos proponer. Sin embargo, algunas reseñas mencionan que al crear mapas de grandes dimensiones, puede haber problemas de rendimiento, especialmente en términos de FPS¹⁰. Esta información es relevante para el desarrollo de una nueva solución, ya que se ha optado por trabajar con una grilla 2D en lugar de una escena 3D para evitar posibles problemas de rendimiento.

3.4.2. Estudio de Arquitecturas

En cuanto al estudio de arquitecturas se halló información potencialmente útil en algunos artículos explicativos acerca de formas de implementar una herramienta de diseño de niveles, dentro de los cuales se rescató el artículo "*Building a Level Editor in Unity*" disponible en el sitio Red-Gate[Talbert, 2019], articuló que proporciona una visión general sobre cómo

¹⁰FPS: abreviatura de “Frames Per Second” (cuadros por segundo) en inglés, y se refiere a la medida de la velocidad a la que una imagen en movimiento se muestra en una pantalla. En el contexto de los videojuegos y la visualización en computadoras, FPS indica la cantidad de cuadros individuales que se muestran cada segundo. Un FPS más alto generalmente se percibe como una experiencia más fluida y suave en términos visuales.

construir un editor de niveles en Unity. Explora conceptos como la manipulación de objetos en la escena, la creación de herramientas de edición y la integración con scripts de control de juego. Si bien se centra en la construcción de un editor de niveles genérico, ofrece ideas y técnicas que pueden ser aplicables al desarrollo de herramientas específicas para este proyecto.

También se consultó el libro *Beginning Unity Editor Scripting*[Kok, 2021]. Este libro proporcionó una valiosa visión general sobre las mejores prácticas y enfoques para el desarrollo de herramientas en el entorno de Unity, incluyendo el scripting de editor. A través de este libro, se aprendió sobre diferentes métodos posibles de personalización del editor para adaptar el flujo de trabajo del juego personalizado.

El libro cubre temas como:

- Comenzar con el scripting de editor en Unity.
- Trabajar con temas avanzados de editor como EditorWindows personalizados y Editor-Tool.
- Estructurar el código C# con espacios de nombres y asmdef¹¹.
- Usar IMGUI¹² y UI Toolkit¹³ para crear interfaces gráficas de editor.
- Establecer un mejor flujo de trabajo para la publicación de activos, el mantenimiento y las actualizaciones iterativas.

3.5. Gestión de Tareas y Backlog del Proyecto

La solución que se propone para el problema identificado en el desarrollo de esta memoria contempla el desarrollo de 3 herramientas (software) de apoyo distintas, donde cada una tienen el objetivo de facilitar el trabajo y agilizar tareas involucradas en el desarrollo y diseño del videojuego. De esta forma podemos separar el desarrollo de la solución propuesta en sprints¹⁴ asociados a cada una de las herramientas.

¹¹asmdef: abreviatura de “Assembly Definition” en Unity. Se refiere a un archivo de definición de ensamblado que permite organizar y administrar de manera más eficiente los scripts y recursos de tu proyecto en Unity. Con asmdef, puedes agrupar scripts relacionados en ensamblados separados dentro de tu proyecto, lo que facilita la modularidad, la reutilización y el mantenimiento del código.

¹²IMGUI: significa “Immediate Mode Graphical User Interface” en Unity. Es un enfoque de interfaz de usuario que opera en el mismo ciclo de actualización de Unity, lo que permite una interacción más directa y rápida con los elementos de la interfaz.

¹³UI Toolkit: es un sistema de interfaz de usuario (UI) desarrollado por Unity para crear y gestionar interfaces de usuario en sus proyectos. Proporciona herramientas y componentes para diseñar y personalizar la apariencia y la interactividad de la interfaz de usuario de manera eficiente y flexible.

¹⁴Se define Sprint a un ciclo de trabajo en la metodología Scrum, en el que se planifican y completan ciertas actividades o tareas establecidas.

Para llevar un Backlog de las tareas en el desarrollo de la solución se optó por utilizar *Notion*¹⁵, ya que el equipo de trabajo de Abstract Digital (interesados en el desarrollo de esta memoria y apoyo de cara a la identificación de requisitos, la validación de la propuesta, etc.) utilizan *Notion* para la organización de sus proyectos, dentro de los cuales está el desarrollo del videojuego de género crafting “Mix the forgotten” con el que se ha trabajado. Además *Notion* es una aplicación multifuncional, la cual no solo servirá para llevar un seguimiento de lo que sería el Backlog, sino que será útil para llevar una documentación de las herramientas desarrolladas.



Herramienta Items crafteables



Figura 13: Documentación en Notion

Fuente: Elaboración propia.

3.5.1. Sprint 1 - Herramienta Items crafteables

En este sprint se identifica como objetivo general el facilitar la creación de nuevos objetos crafteables y recetas para el sistema de crafteo de un videojuego desarrollado en Unity. Para cumplir con lo anterior se plantea desarrollar una herramienta de apoyo que simplifique y agilice el desarrollo y diseño de los 2 componentes fundamentales de un objeto crafteable en un juego desarrollado en Unity, que son la receta del objeto y el objeto de por sí.

De lo anterior es que surgen dos tareas generales para el desarrollo de este sprint, las cuales luego contemplan una serie de subtareas que llevan al desarrollo de las funcionalidades y requisitos identificados anteriormente:

1. Construir herramienta en la que se puedan crear las recetas más gráficamente.
 - Seleccionar forma para llenar en las casillas de la grilla
 - Elegir material del ingrediente requerido

¹⁵Aplicación organizadora de tareas que permite gestionar la planificación en cualquier ámbito

- Elegir función del ingrediente requerido
 - Pintar/Borrar casillas de la grilla con el ingrediente seleccionado
 - Generar el *ScriptableRecepy* de la receta
 - Verificar que no exista una receta igual (también verifica la matriz transpuesta) evitando duplicados y otro tipo de errores
2. Generar *prefab* asociado a la receta creada (el objeto de por sí)
- Vista de los ingredientes de la receta
 - Implementar sistema de *drag and drop*¹⁶ para ordenar los ingredientes de la receta
 - Generar el prefab resultante con los ingredientes (listo para modificar el diseño gráfico del objeto)
 - Añadirle *scripts*¹⁷ necesarios al *prefab* para el correcto funcionamiento del objeto.

3.5.2. Sprint 2 - Herramienta Diseño de niveles

En este sprint se reconoce como objetivo general el facilitar el diseño de niveles para un videojuego desarrollado en Unity3D. Para lo anterior es que se plantea desarrollar una herramienta de diseño de niveles que permita agilizar de forma significativa el tedioso proceso de diseñar el nivel de un videojuego 3D. Para esto es que se idea una herramienta de Unity, que por medio de una grilla visual se puedan instanciar bloques/terreno/particiones del mapa.

Para desarrollar la idea de herramienta es que se reconocen las siguientes tareas para el desarrollo del sprint, las cuales van acorde a las necesidades y requisitos reconocidos o recogidos con anterioridad:

- Seleccionar el prefab para llenar un *tile*¹⁸ del terreno
- Pintar con un pincel sobre la grilla del terreno con el prefab seleccionado
- Opción de guardar como prefab
- Opción de borrar todo (reseteo del mapa)

¹⁶Sistema drag and drop (traducido al español arrastrar y soltar) es un sistema que se utiliza en las interfaces gráficas y permite tomar un elemento de la pantalla y desplazarlo hacia otro lugar utilizando el mouse.

¹⁷El motor de videojuegos Unity se basa en la programación en lenguaje C#, de forma que cada objeto suele ir con scripts asociados y que definen los atributos, comportamiento y demás aspectos del objeto

¹⁸El *tile* es la parte gráfica de cada videojuego que puede ser utilizada para completar partes de un fondo por medio de un *tileset* (set de *tiles*). En el caso de esta herramienta corresponde a una casilla y representa un bloque que conforma el terreno del nivel diseñado

- Opción de cargar mapa creado con la herramienta
- Ajustar dimensiones de la grilla sobre la cual pintar
 - Elegir las dimensiones antes de generar la grilla
 - Reajustar las dimensiones de la grilla del mapa actual
- Ajustar altura sobre la cual instanciar el prefab
- Poder colocar otros prefab a distintas alturas en una misma posición de la grilla
 - Colocar otros prefab encima del suelo
 - Distintas vistas de grilla según la capa o altura (GUI)
- Botón para deshacer el ultimo prefab instanciado
- Agregar y ajustar para que puedan ser instanciados con la herramienta otros prefabs que hayan además del terreno, como enemigos e items del videojuego

3.5.3. Sprint 3 - Herramienta importación de Assets gráficos

En este sprint se tiene como objetivo general facilitar la integración de nuevos Assets gráficos como nuevos ingredientes para el sistema de crafteo del videojuego. Para esto se propone desarrollar una herramienta en Unity que permita elegir Assets gráficos (modelos 3D), ajustar el tamaño y dimensiones del modelo 3D acorde a las dimensiones del videojuego y los demás ingredientes ya integrados, para luego eligiendo el material, forma y función del ingrediente se le asignen de forma automática los scripts y demás componentes que permiten el correcto funcionamiento del objeto como un ingrediente para el sistema de crafteo del videojuego.

Para lograr lo anterior se reconocen las siguientes tareas/funcionalidades a desarrollar a lo largo de este tercer sprint:

- Elegir el Asset gráfico que se quiere integrar
- Elegir la forma del ingrediente a generar (cilindro, caja, rueda, etc.)
- Ajustes y configuración del ingrediente a generar
 - Definir propiedades básicas del ingrediente (nombre, tag)
 - Ajustar el script de item como componente
 - Elegir material principal del ingrediente
 - Definir lista de materiales del ingrediente
 - Definir lista de funciones del ingrediente
 - Definir la escala/dimensiones del Asset gráfico acorde a lo que quiere

- Instanciar un modelo de referencia para comparar
- Modificar el *transform*¹⁹ del Asset gráfico (escala, rotaciones, posición)

3.6. Documentación de las Herramientas propuestas en la solución

Las 3 herramientas se desarrollaron teniendo en cuenta aspectos como flexibilidad, adaptación y escalabilidad, como bien se explico en los métodos de desarrollo de software aplicados. Para cumplir con lo anterior es que se optó por diseñar un prefab modificable que cumpla la función de darle flexibilidad y escalabilidad a las herramientas, teniendo la opción de agregar nuevos elementos.

Este prefab es utilizado para definir los estilos de los botones, como los prefab asociados y que después son instanciados para diseñar el nivel. Este prefab permite ir agregándole más opciones a la herramienta evitando meterse de lleno con el código

3.6.1. Herramienta de diseño de nivel

Esta herramienta utiliza un prefab llamado StyleManager14 que contiene un script StyleManager.cs y que es utilizado en la Herramienta de items crafteables, la Herramienta de importación de Assets y generación de ingredientes, y en esta Herramienta diseño de niveles. Este prefab es utilizado para definir los estilos de los botones, como los prefabs asociados y que después son instanciados para diseñar el nivel. Este prefab permite ir agregándole más opciones a la herramienta evitando meterse de lleno con el código, otorgándole flexibilidad y facilitándole a los usuarios escalar la herramienta agregándoles los elementos que puedan ir diseñando y se implementen al proyecto en un futuro. Otro aspecto importante de esta implementación es la gran ventaja que supone el poder definir los prefab de los bloques de terreno dentro del proyecto¹⁵, a diferencia de cualquier otra herramienta disponible en la Unity Asset Store²⁰ que te ofrecen modelos genéricos que después se deben modificar.

Como bien se ha explicado con anterioridad, esta herramienta lo que hace es dar la posibilidad de diseñar un escenario o nivel del videojuego de manera más ágil por medio de una interfaz que consta de distintas secciones o menús, los cuales se detallarán a continuación:

Al abrir la herramienta en la escena del editor de Unity se verifica automáticamente si ya hay un escenario sobre el cual se está usando la herramienta para diseñar niveles. Primero hay dos secciones base de la herramienta, que aparecerán siempre a disposición del usuario,

¹⁹Todos los objetos dentro de Unity contienen un componente llamado *transform*, el cual detalla la posición, rotación y escala del objeto.

²⁰El Asset Store de Unity es el hogar de una creciente biblioteca de assets comerciales y gratuitos creados por Unity Technologies y miembros de la comunidad. Hay una gran cantidad de assets disponibles, desde texturas, modelos y animaciones hasta ejemplos de proyectos completos, tutoriales y extensiones del editor.

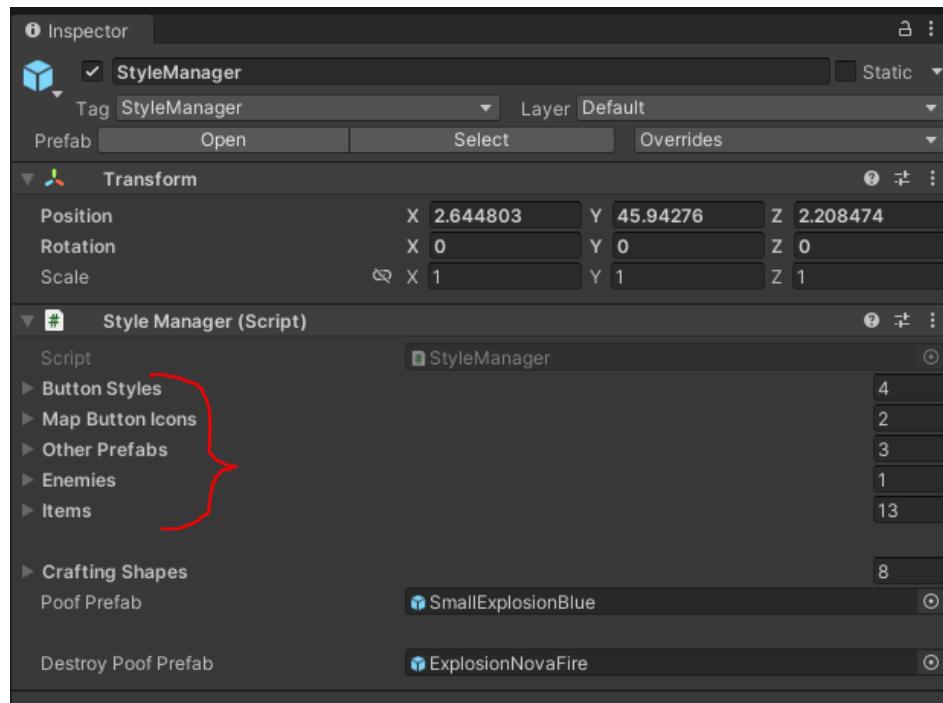


Figura 14: StyleManager - Prefab auxiliar de la herramienta.

Fuente: Elaboración propia.

los cuales tienen el fin de empezar a crear un nuevo escenario o cargar uno para continuar diseñando:

La primera sección de la interfaz del usuario ofrece la posibilidad de definir unas dimensiones iniciales para el diseño del escenario. Luego se generará una grilla base sobre la cual empezar a diseñar el escenario (luego estas dimensiones iniciales se pueden ir reajustando dinámicamente mientras se diseña el escenario).

La interfaz (figura 16) de este pequeño menú inicial es muy simple, consta con 2 campos para llenar y especificar la cantidad de filas y columnas que se requieren para generar la grilla inicial, además del botón para generar la grilla inicial, desbloqueando el resto de menus y poder proceder al diseño del nivel.

Esta otra parte de la interfaz (figura 17) ofrece la posibilidad de cargar un escenario diseñado con esta herramienta con anterioridad y que se encuentre en los archivos del proyecto, para continuar o reanudar el diseño de este escenario.

Para esto se tiene la opción de elegir el directorio en el que se encuentren guardado los escenarios diseñados con esta herramienta, y así poder cargar el escenario que se deseé. Al momento de cargar uno de estos escenarios se desbloquea el resto de secciones para poder proceder y continuar diseñando el escenario que se ha cargado.

Cuando se detecta el escenario dentro de la de escena de Unity, la herramienta muestra otras

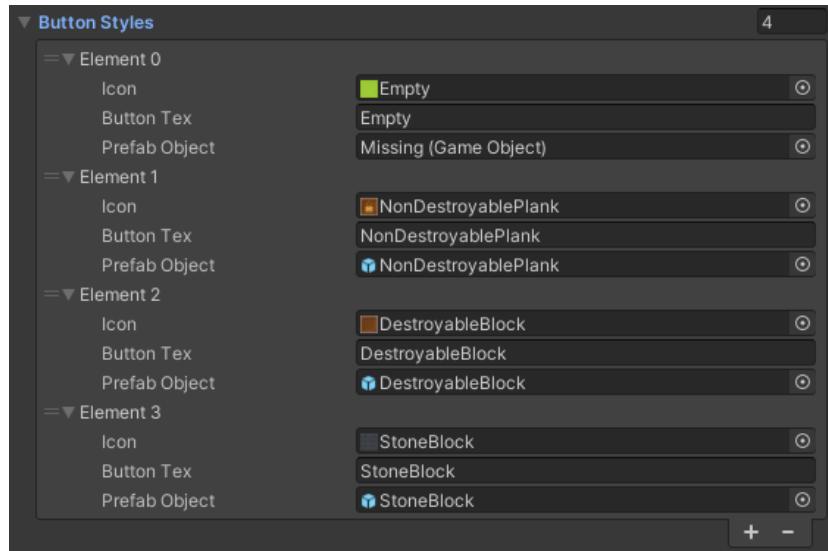


Figura 15: StyleManager - Campos para definir opciones de terreno.

Fuente: Elaboración propia.

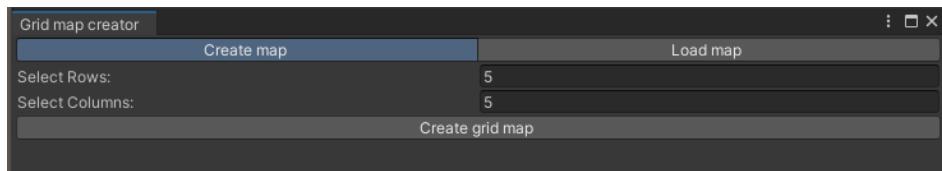


Figura 16: Interfaz inicial de la herramienta de diseño de niveles.

Fuente: Elaboración Propia

dos secciones que le dan la posibilidad al usuario de diseñar dinámicamente el escenario dentro de la escena de Unity, además de que el menú de carga se convierte en un menú de Carga y Guardado, por lo que aparece la opción de guardar el escenario que se encuentran diseñando (figura 18):

La primera sección y probablemente la más fundamental de la interfaz gráfica de esta herramienta es el Menú de la grilla o *GridMap*:

Esta sección de la interfaz contiene una vista del escenario que se está diseñando en forma de una grilla de dos dimensiones. Los cuadrados que componen esta grilla son botones que cumplen la función de diseñar el terreno del escenario.

Cuando se habla de diseñar dinámicamente el escenario, es porque en esta grilla se rellena/pinta con los bloques base del terreno o diseño del videojuego en cuestión, de forma que al ir pintando sobre la grilla 2D de forma simultanea se van instanciando los bloques y elementos en la escena del editor de Unity.

En este menú (figura 19) se encuentran distintas opciones y botones que buscan ayudar a

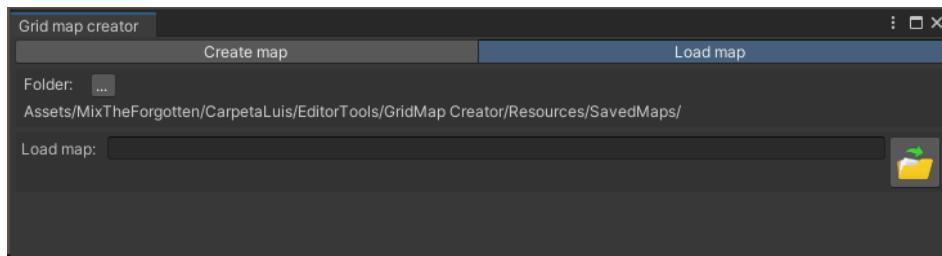


Figura 17: Interfaz inicial de la herramienta de diseño de niveles.

Fuente: Elaboración Propia

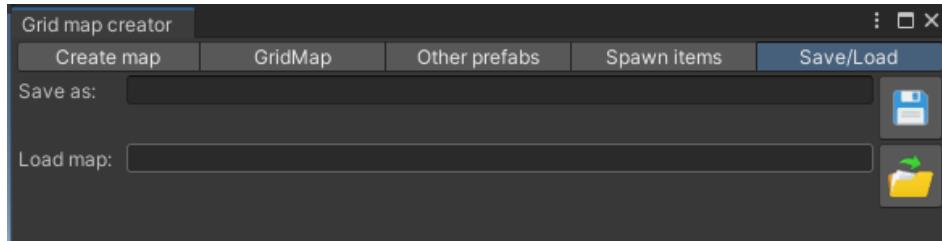


Figura 18: Menú de cargado/guardado del escenario - Herramienta de diseño de niveles.

Fuente: Elaboración Propia

diseñar el escenario:

- **Añadir/Remover columnas:** Permite reajustar de forma dinámica las dimensiones de la grilla.
- **Añadir/Remover filas:** Permite reajustar de forma dinámica las dimensiones de la grilla.
- **Clear:** Botón que remueve todo de la grilla y la escena en Unity, funcionando como un reseteo para empezar desde 0.
- **Botones con forma de bloque:** Botones conformados por imágenes de previsualización de los bloques base disponibles para el terreno y diseño del videojuego, al presionar uno de estos botones se elige el bloque con el cual se pinte el cuadro de la grilla que se pinte a continuación (estos bloques son definidos en el prefab StyleManager instanciado en la escena de Unity, de forma que se pueden agregar los bloques y redefinir de acuerdo al proyecto).
- **Height o Capa de altura de la grilla:** Como esta es una grilla 2D y el escenario es 3D, se implementa un sistema de capas para modificar a distintas alturas y que la grilla que se muestra en la interfaz va acorde a la capa seleccionada. Además de los botones + y - que permiten elegir la capa de altura.

Otro menú que aparece es el de *Other prefabs* (Otros prefabs), el cual permite instanciar otros prefabs distintos a los bloques de terreno base en el escenario. Estos prefabs que se

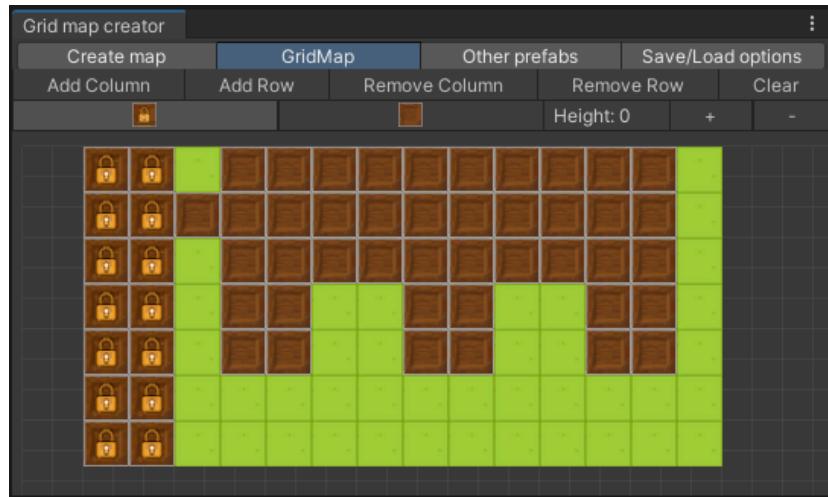


Figura 19: Menú de la grilla - Herramienta de diseño de niveles.

Fuente: Elaboración Propia

pueden instanciar pueden ser NPCs, herramientas, armas, ingredientes u objetos, y básicamente cualquier otro prefab necesario que no sea un bloque base del terreno. Para esto en el prefab StyleManager explicado con anterioridad se tiene la opción de definir los prefabs necesarios de acuerdo al proyecto.

En el caso del proyecto “Mix: the forgotten” que está siendo ejemplo para implementar estas herramientas, se ha decidido incluir 3 tipos de prefabs fundamentales (además del terreno): Mixers (plataformas para craftear objetos arrojándole items), Enemigos e Items. De esta forma, en la interfaz aparece una pestaña para cada tipo de prefab, además 1 botón para cada uno de los prefabs incluidos en cada tipo, de forma que podamos seleccionar el prefab a instanciar en el escenario, además de un botón *Undo* (Deshacer) para borrar el último prefab que se instancie en el escenario.

A continuación se puede ver como es que se instancian en la escena dinámicamente mientras diseñamos el escenario por medio de la interfaz gráfica de la herramienta de apoyo (figura 21).

3.6.2. Herramienta importación de Assets/Nuevos ingredientes

Esta herramienta como bien indica su nombre busca ser un apoyo de cara a la importación de Assets gráficos y diseñar nuevos ingredientes para el sistema de crafteo del videojuego. Esta herramienta ofrece agilizar el proceso de integrar Assets gráficos al proyecto, dando la posibilidad de ajustar las características del ingrediente por medio de una interfaz gráfica bien simple, ajustando por detrás los scripts asociados al ingrediente. También esta herramienta ayuda a modificar las dimensiones del Asset gráfico para que sea acorde a las dimensiones de los objetos dentro del proyecto.

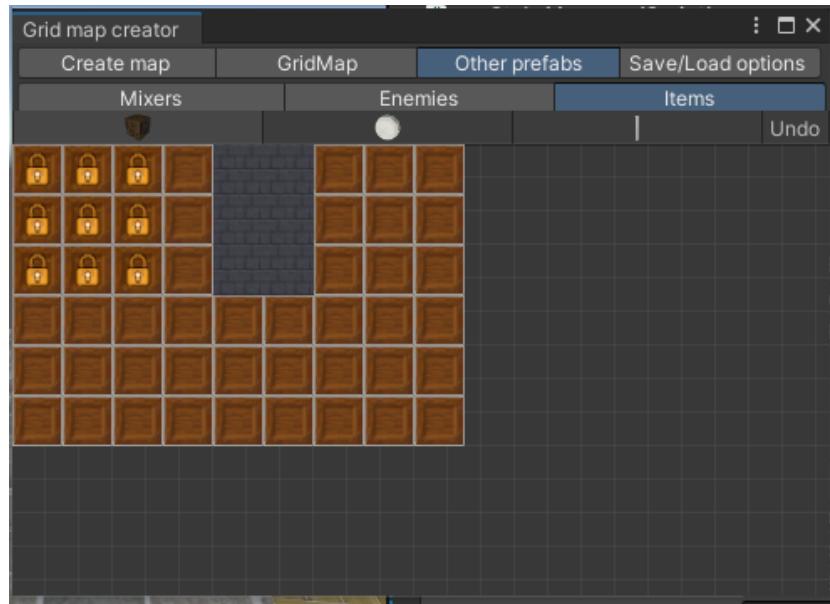


Figura 20: Menú de otros prefabs - Herramienta de diseño de niveles.

Fuente: Elaboración Propia

Esta herramienta se separa en dos menús principales, de forma que primero al abrir la herramienta se accede a un menú inicial (figura 22) que tiene que ver con el cargado del Asset a importar, seleccionar la forma del ingrediente y una opción de seleccionar la carpeta en que se guardará el nuevo ingrediente.

- Lo primero es un campo en el cual se debe seleccionar el Asset gráfico que se desea integrar y usar como modelo 3D para el nuevo ingrediente a generar (por ejemplo un modelo 3D de un lápiz).
- Botón para redefinir la carpeta dentro del proyecto en la que se desea que vayan a guardarse los ingredientes que se generen con esta herramienta.
- Botones *Item shape* (forma del item/ingrediente): Son botones que permiten seleccionar la forma del ingrediente que se desea integrar, de forma que si selecciono un cilindro, el ingrediente que se genere después actúa como un cilindro en el sistema de crafteo del videojuego.
- Botón *Instantiate* (Instanciar) que instancia el Asset gráfico en la escena de Unity y un modelo 3D que actuará como una referencia (figura 23) de la forma seleccionada para el ingrediente. Al apretar este botón se dirigirá al menú siguiente para seguir ajustando el ingrediente.

Luego de presionar el Botón “*Instantiate*” se procede al siguiente menú que se compone de 3 partes:

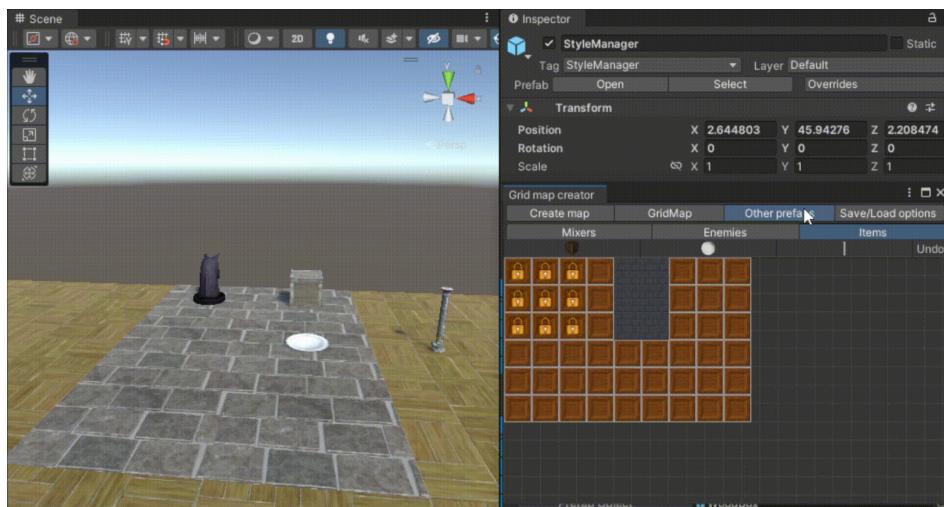


Figura 21: Escena de Unity y Herramienta de diseño de niveles

Fuente: Elaboración Propia

1. **Propiedades básicas del Objeto:** Contiene un campo para definir el nombre del ingrediente a generar y la posibilidad de elegir el Tag del GameObject en caso de que sea necesario.
2. **Item component:** Un bloque de campos destinado a ajustar el script de item generado automáticamente para que el item creado funcione como un ingrediente y pueda ser utilizado en el sistema de crafteo (figura 25). Los campos, valores y variables ajustables dentro de esta parte son los siguientes:
 - Un dropdown²¹ para seleccionar el material principal del ingrediente, ya que el sistema de crafteo exemplar utilizado dentro del videojuego que se está usando como base para el desarrollo de estas herramientas define una serie de materiales de los que pueden ser los ingredientes a utilizar. Este material tiene importancia de cara a las recetas.
 - Está la posibilidad de ajustar los materiales secundarios, ya que a lo mejor el ingrediente no solo está hecho de un material.
 - También existe la posibilidad de ajustar “funciones”, las cuales tienen que ver con otras características para el item, como por ejemplo que el ingrediente sea conductor eléctrico (esto también va acorde al sistema de crafteo implementado en el videojuego “Mix: the forgotten”).
3. **Transform component:** Un bloque de campos y ajustes relacionados al Transform del objeto, que tiene que ver con las dimensiones del objeto, el escalado, posición, etc. Esta parte tiene como objetivo dar la posibilidad de moldear el Asset gráfico para dejarlo acorde con las dimensiones de los ingredientes dentro del proyecto, ajustando las

²¹Dropdown: El Dropdown es uno de los elementos que ofrece Unity para el diseño de UI. Es utilizado para que el usuario pueda escoger una opción dentro una lista de opciones.

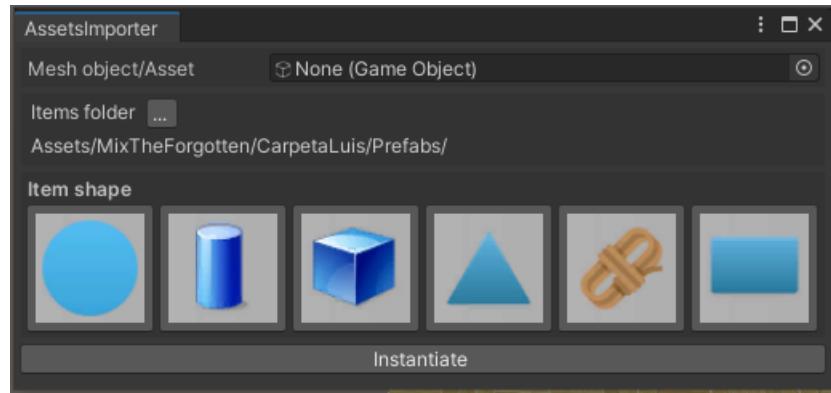


Figura 22: Menú inicial - Herramienta de importación de Assets gráficos y diseño de ingredientes

Fuente: Elaboración Propia

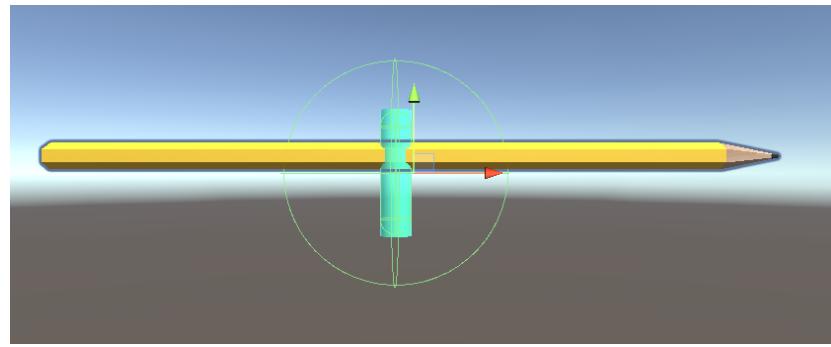


Figura 23: Modelo referencia instanciado - Herramienta de importación de Assets gráficos y diseño de ingredientes

Fuente: Elaboración Propia

dimensiones de forma dinámica para dejarlo como el modelo 3d de referencia (figura 26). Como este bloque está destinado a darle la forma necesaria al modelo gráfico que se usa para generar el ingrediente, para lo cual la herramienta ofrece lo siguiente en su interfaz:

- Un *slider*²² para ir ajustando la escala o tamaño del modelo de forma dinámica, buscando ajustarlo a las dimensiones requeridas, las cuales son mostradas por el modelo de referencia, como se puede ver en la imagen (figura 26).
- Aparte de poder cambiar la escala con el slider, también se tiene la posibilidad de elegir una escala distinta para cada eje (X, Y, Z) en caso de que el modelo se requiera alargar, ensanchar, etc.
- Para rotar el modelo se tienen 3 sliders, uno para cada eje.

²²Un slider o deslizador en el diseño de interfaces permiten a los usuarios seleccionar un valor dentro de un rango, como por ejemplo para ajustar los niveles de volumen del computador.

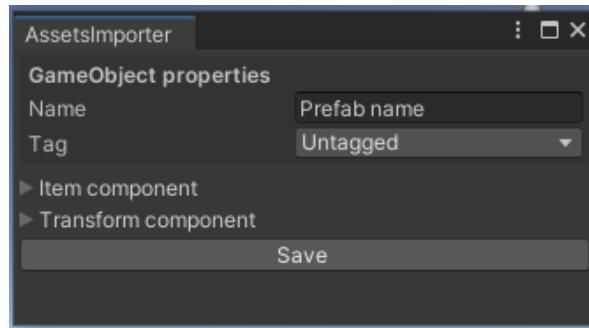


Figura 24: Menú de ajustes - Herramienta de importación de Assets gráficos y diseño de ingredientes

Fuente: Elaboración Propia

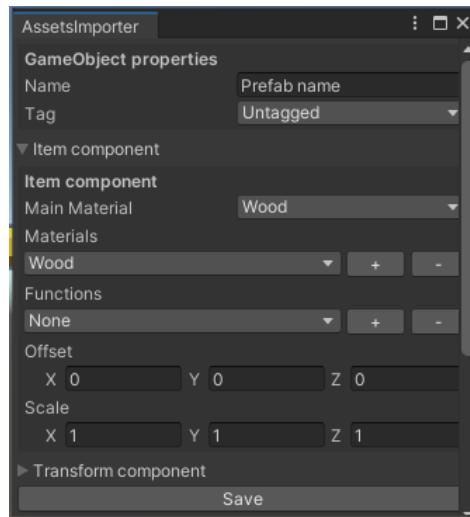


Figura 25: Item component - Herramienta de importación de Assets gráficos y diseño de ingredientes

Fuente: Elaboración Propia

- Se tienen 3 campos para elegir la posición del modelo, con tal de colocarlo al centro y calzar el modelo con el modelo de referencia.
- Por ultimo está el botón de guardado del ingrediente para cuando ya se encuentre listo el ingrediente, al apretar el botón se guardará un prefab del ingrediente creado dentro del directorio del proyecto de acuerdo a la carpeta seleccionada como destino anteriormente.

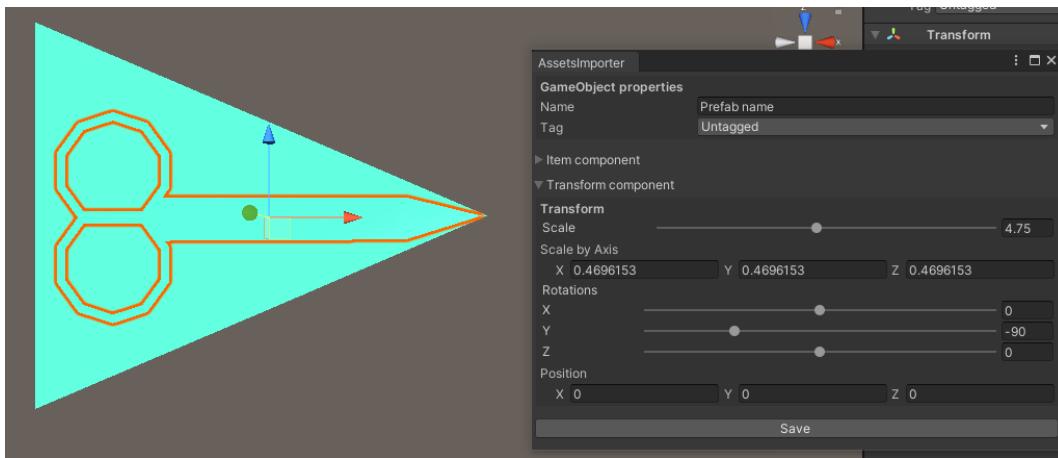


Figura 26: Transform component - Herramienta de importación de Assets gráficos y diseño de ingredientes

Fuente: Elaboración Propia

3.6.3. Herramienta Items crafteables

Esta tercera herramienta de apoyo tiene como finalidad facilitar y agilizar el proceso de creación de nuevos objetos crafteables dentro del videojuego (por ejemplo un arma), junto con todo lo que conlleva esto. Para lo anterior es que esta herramienta separa su funcionamiento en 2 objetivos:

1. Diseñar el objeto crafteable.
2. Crear la receta necesaria para que este nuevo objeto sea crafteable.

Entonces esta tercera herramienta propuesta como apoyo constará de 2 menús en su interfaz, de forma que primero aparece una interfaz gráfica dedicada al diseño de la receta del objeto que se tiene en mente, para lo cual se utilizó un método parecido a la herramienta de diseño de niveles, implementando un sistema de grilla de botones. A continuación se explican los botones, campos y variables que se ofrece la herramienta para crear la receta:

- Se tiene una grilla de cuadros grises que funciona de tal forma que cada cuadro puede ser reemplazado o pintado con un ingrediente, ofreciendo así una forma bien simple y gráfica de definir las recetas de los objetos crafteables.
- Botones “Crafting Shape”: Se tienen varios botones con imágenes asociadas a la forma que pueden tener los ingredientes dentro del videojuego. Estos botones están para que al presionarlos seleccionar la forma del ingrediente para luego pintar los cuadros de la grilla. Funciona de forma parecida como cuando seleccionamos un lápiz o pincel dentro de aplicaciones de dibujo como Paint.

- Este menú cuenta con 2 dropdowns; uno para elegir el Material del ingrediente y otro para elegir una función del ingrediente. Según los valores seleccionados el ingrediente que se pinte a continuación dentro de la grilla tendrá estos valores. Esto permite que una receta dependa no tan solo de la forma de los ingredientes, sino que puede requerir por ejemplo un cilindro de madera y un cubo de metal que sea conductor eléctrico.
- Abajo se tiene un campo de texto para escribir el nombre que tendrá la receta que se cree del objeto, es decir, el nombre del *ScriptableObject* asociado a la receta. Idealmente puede ser el mismo nombre del objeto de por sí.
- Por último se tiene el botón “Create ScriptableRecepy” que al presionarlo se dará paso al siguiente menú de la herramienta y se generará el archivo de clase *ScriptableRecepy* (figura 28) que es una clase creada en base al tipo *ScriptableObject*, que como vimos con anterioridad en el marco teórico son útiles de cara a la implementación de Objetos, Cartas, Armas y más elementos de un videojuego.

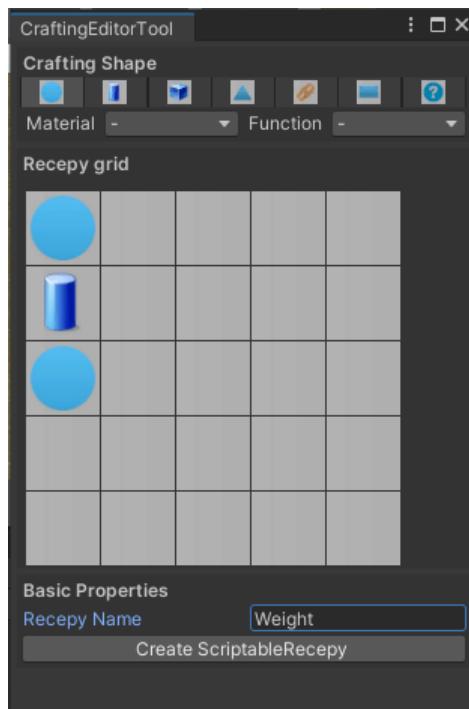


Figura 27: Menú creación de receta - Herramienta de Items crafteables
Fuente: Elaboración Propia

Luego de haber diseñado la receta del item crafteable, la herramienta da paso a un segundo menú para diseñar el prefab del item crafteable, es decir, el objeto de por sí. Para esto primero es necesario entender la estructura que tienen los items crafteables, para lo cual se ha tenido en cuenta la estructura utilizada en el videojuego.

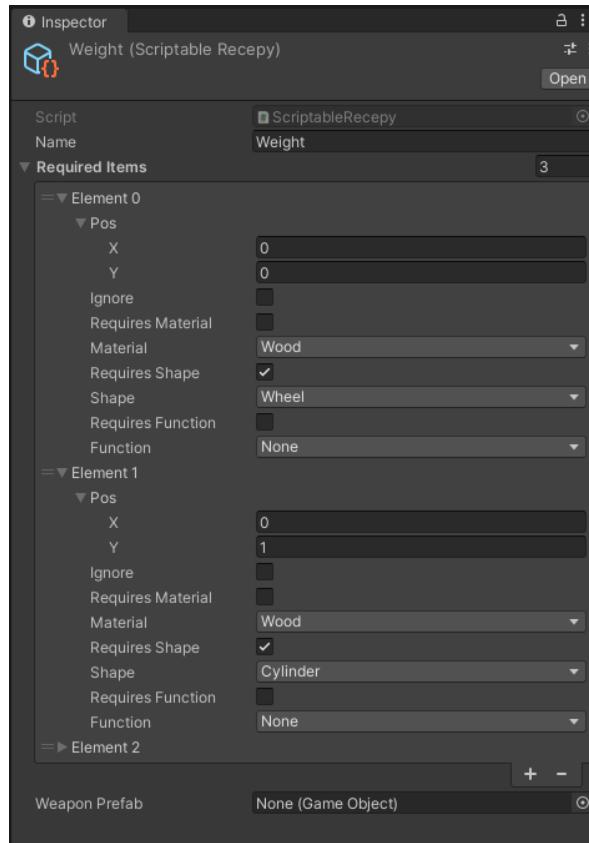


Figura 28: ScriptableRecepy - Herramienta de Items crafteables

Fuente: Elaboración Propia

En el videojuego utilizado como base de apoyo para la implementación de esta herramienta los objetos son crafteados con ciertos ingredientes necesarios según indique la receta, los cuales son arrojados en unas plataformas llamadas “mixers”, de forma que si se cumple una receta aparece el objeto crafteable, de modo que al craftearse dicho objeto desaparecen los ingredientes depositados en la plataforma, ya que pasan a conformar parte del objeto crafteado. Teniendo en cuenta lo anterior es que los prefab de los objetos crafteables siguen la siguiente estructura (figura 29) dentro de la *Hierarchy*²³ o jerarquía de Unity.

En la imagen anterior (figura 29) se tiene como ejemplo el prefab de los martillos, donde en la jerarquía se tienen los ingredientes (como el cubo y el cilindro) dentro de un objeto padre, sobre los cuales se puede modificar el Transform para escalar, rotar y cambiar de posición los ingredientes para darle forma al objeto final.

Entonces la herramienta ofrece la posibilidad de crear *placeholders*, los cuales vienen siendo los objetos padres sobre los cuales se guardan los ingredientes. Al agregar *placeholders* en

²³La *Hierarchy* contiene cada GameObject de la escena actual. Algunos de estos son instancias directas de archivos de asset como modelos 3D, y otras son instancias de Prefabs, objetos personalizados que van hacer gran parte de su juego.

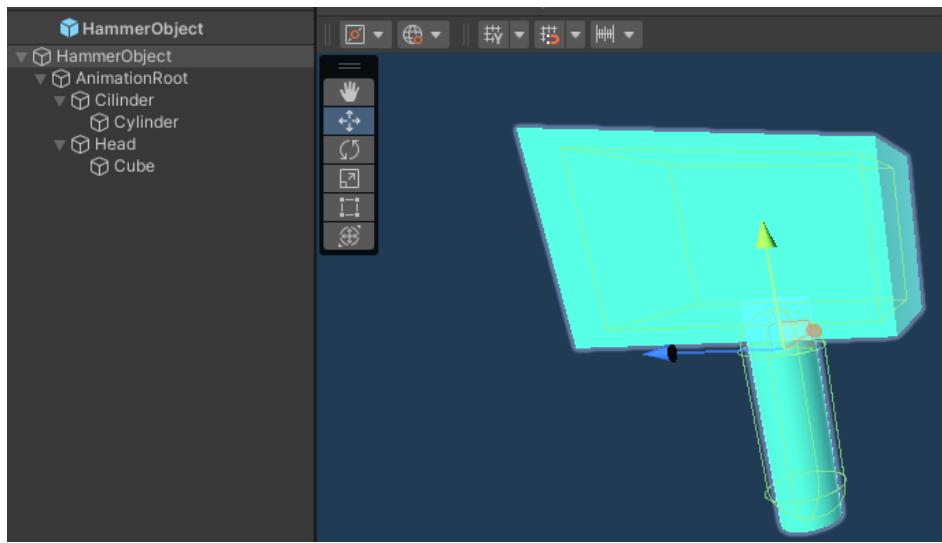


Figura 29: Ejemplo prefab Item crafteable - Herramienta de Items crafteables

Fuente: Elaboración Propia

la interfaz gráfica de la herramienta aparecen cuadrillas, sobre las que se deben arrastrar los ingredientes, de esta forma cuando se cree el Prefab la jerarquía de los objetos será acorde a como fueron organizados los *placeholders* e ingredientes.

En la imagen siguiente se observa un paso a paso de todo lo anterior para mostrar como es que funciona este menú (figura 30). En este caso se está diseñando una pesa de ejercicios que esta conformada por dos discos pesados y una barra:

Para poder realizar este procedimiento, este menú está compuesto por los siguientes campos, botones y opciones:

- Un campo de texto para designarle nombres a los placeholders mencionados anteriormente. Los nombres comúnmente puede ser una forma de distinguir la parte del objeto, por ejemplo puede ser el nombre del mismo ingrediente.
- Botón de añadir el placeholder según el nombre elegido.
- Botón de remover el último placeholder agregado.
- Una zona que lleva un título de “Drag and Drop prefab shapes”. En esta zona se pueden ver unos cuadros con imágenes de ciertas formas que tienen que ver con los ingredientes de la receta que hemos creado, las cuales debemos arrastrarlas para dejarlas dentro de uno de los placeholder creado.
- Los cuadros de los placeholders aparecen a medida que se van agregando. Estos placeholders se comportan como ranuras sobre las cuales se pueden arrastrar los ingredientes para luego dar como resultado el prefab resultante del objeto crafteable con la jerarquía acorde a como se organizaron los ingredientes y placeholders.

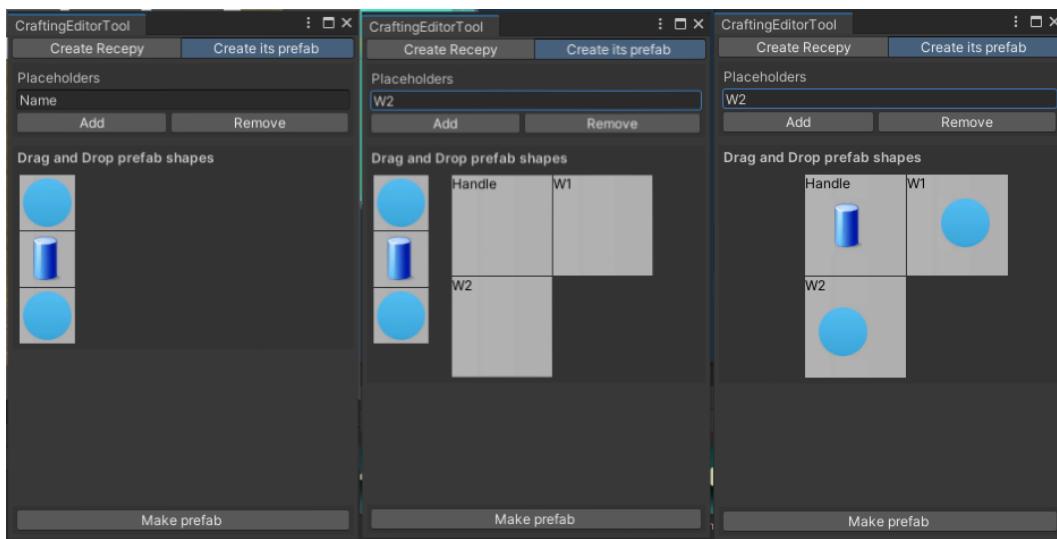


Figura 30: Creación prefab del objeto crafteable - Herramienta de Items crafteables
Fuente: Elaboración Propia

- Al final se tiene un botón con la etiqueta “Make prefab” que al ser presionado dará por finalizado el trabajo de la herramienta, dando como resultado el generar automáticamente el prefab del objeto crafteable junto con sus scripts y componentes necesarios para su funcionamiento, además del *ScriptableRecepyp* de la receta asociada al objeto.

El prefab generado ya puede ser crafteado en el videojuego, lo único que resta es darle forma al objeto modificando los transform (en los placeholders) y sus colliders (figura 31), además de añadirle el funcionamiento que se le quiera dar al objeto en caso de que sea requerido (agregándole scripts).

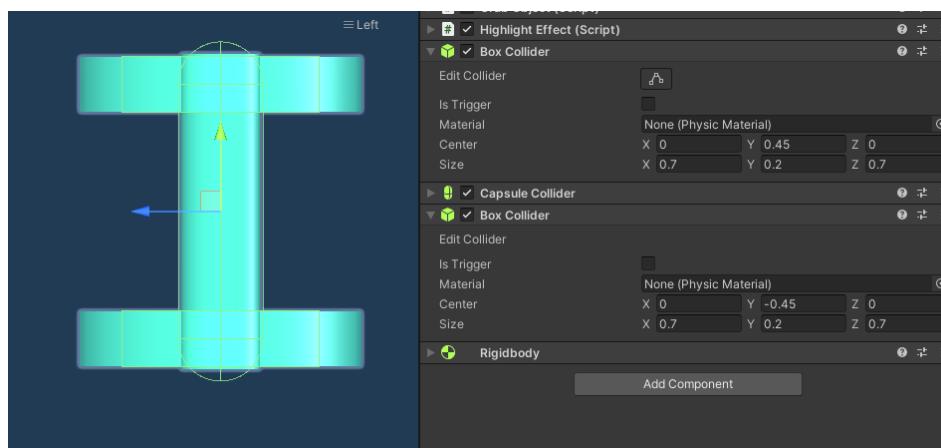


Figura 31: Ejemplo de Prefab final del objeto crafteable - Herramienta de Items crafteables
Fuente: Elaboración Propia

CAPÍTULO 4

VALIDACIÓN DE LA SOLUCIÓN

En esta sección se describe el proceso de validación de las herramientas desarrolladas y las evaluaciones realizadas a los usuarios muestrados. Asimismo, se hace una descripción de los usuarios y se desarrolla un análisis de los resultados obtenidos. Además, las herramientas desarrolladas en esta memoria ya están siendo usadas por el equipo de trabajo encargado del desarrollo de un videojuego de género crafting, lo que también se tendrá en cuenta al momento de validar las distintas herramientas y puede ser una muestra del buen funcionamiento de las herramientas.

4.1. Proceso de Validación mediante Testing de usuarios externos

Para asegurar la funcionalidad y utilidad de las herramientas desarrolladas, se diseñó un proceso de validación que incluye pruebas con estudiantes de informática y desarrolladores de videojuegos ajenos al proyecto sobre el cual se idearon inicialmente las herramientas.

4.1.1. Validación de la Herramienta de Diseño de Niveles

- **Participantes:** Estudiantes de informática y desarrolladores de videojuegos.
- **Procedimiento:** La herramienta de diseño de niveles fue empaquetada para facilitar su testeo de manera independiente al proyecto “Mix: the forgotten”. Los participantes realizaron tres tareas específicas diseñadas para evaluar la usabilidad y funcionalidad de la herramienta:
 - Crear planicie de 6 bloques de largo y ancho, con una plataforma de mezcla de objetos al medio. Da igual el material del bloque que conforma el suelo y debe guardarse el resultado final como un nuevo prefab.
 - Diseñar libremente un terreno con diferentes altitudes, que tenga tanto bloques de madera como de ladrillo. Guardar este terreno diseñado con la herramienta.
 - Cargar el terreno diseñado en la anterior prueba, modificarlo agregándole enemigos al escenario y luego borrar todo el escenario, dejando el escenario limpio.
- **Evaluación:** Después de completar las tareas, los participantes respondieron a una encuesta en Google Forms sobre su experiencia usando la herramienta. La encuesta incluyó las siguientes preguntas:
 - **Información del Usuario:**

- Edad
- Ocupación: Estudiante de Ingeniería civil Informática (o alguna carrera relacionada), Desarrollador de videojuegos, Otro.
- ¿Qué tanta experiencia tiene desarrollando juegos con Unity? (Escala de 1 a 5, donde 1 es Nada y 5 es Mucha)

- **Experiencia con la Herramienta:**

- ¿Cómo calificaría su experiencia general con la herramienta de diseño de niveles? (Escala de 1 a 5, donde 1 es Muy mala y 5 es Muy buena)
- ¿Le resultó fácil de aprender a usar la herramienta? (Escala de 1 a 5, donde 1 es Muy difícil y 5 es Muy fácil)
- ¿Encontró algún problema o dificultad al usar la herramienta? (Marque todas las que apliquen: Dificultad para entender las funciones, Problemas de navegación, Errores o fallos, Rendimiento lento, Otro)
- ¿Pudo realizar con éxito las tareas que se solicitaron realizar usando la herramienta? (Marque aquellas que sí pudo completar exitosamente: Tarea 1, Tarea 2, Tarea 3)
- ¿Qué dificultades ha tenido con la utilización de la herramienta al momento de realizar las tareas?
- ¿Cuánto tiempo aproximadamente le tomó completar la tarea 1 usando la herramienta de diseño de niveles?
- ¿Cuánto tiempo aproximadamente le tomó completar la tarea 2 usando la herramienta de diseño de niveles?
- ¿Cuánto tiempo aproximadamente le tomó completar la tarea 3 usando la herramienta de diseño de niveles?

- **Eficiencia en el Diseño:**

- En caso de haber trabajado con Unity diseñando niveles/terrenos anteriormente, ¿le pareció que la herramienta mejoró su eficiencia en comparación a cuando no la usa? (Escala de 1 a 5, donde 1 es Mucho menos eficiente y 5 es Mucho más eficiente)

- **Feedback General:**

- ¿Qué le gustó más de la herramienta?
- ¿Qué mejoraría de la herramienta?
- Comentarios adicionales:
 - ¿Recomendaría esta herramienta a otros diseñadores/desarrolladores de videojuegos? (Sí, No, Tal vez)

4.2. Resultados de las Encuestas de la Herramienta de Diseño de Niveles

A continuación, se presentan los resultados obtenidos de las encuestas para la herramienta de diseño de niveles:

- **Facilidad de Uso:** La mayoría de los participantes (7) calificaron la herramienta con un 5 (muy buena), mientras que 4 la calificaron con un 4 (buena). Esto indica una alta valoración de la facilidad de uso de la herramienta.
- **Eficiencia:** La herramienta mejoró significativamente la eficiencia en el diseño de niveles, con 7 participantes calificándola con un 5 (mucho más eficiente) y 1 con un 4 (más eficiente). Solo un participante la calificó con un 3 (regular) y 2 no respondieron.
- **Satisfacción General:** Los participantes expresaron una alta satisfacción general con la herramienta, destacando la facilidad de uso y la rapidez para crear niveles.
- **Sugerencias para Mejoras:** Las principales sugerencias para mejoras incluyeron la necesidad de una interfaz más intuitiva, la adición de una función de deshacer y una mejor visualización entre niveles.

Tarea	Éxito [%]	Tiempo Promedio [min:seg]
Tarea 1	100 %	1:33
Tarea 2	100 %	2:12
Tarea 3	100 %	2:00

Tabla 1: Porcentaje de Éxito y Tiempos Promedio en la Realización de Tareas para la Herramienta de Diseño de Niveles

A continuación se muestran gráficos con los resultados de los principales aspectos evaluados en las encuestas: Experiencia general con la herramienta, Facilidad de uso y Mejora en la eficiencia al utilizar la herramienta (en caso de querer ver los resultados de las pruebas en mayor detalle revisar los anexos incluidos al final del documento).

¿Cómo calificaría su experiencia general con la herramienta de diseño de niveles?
11 respuestas

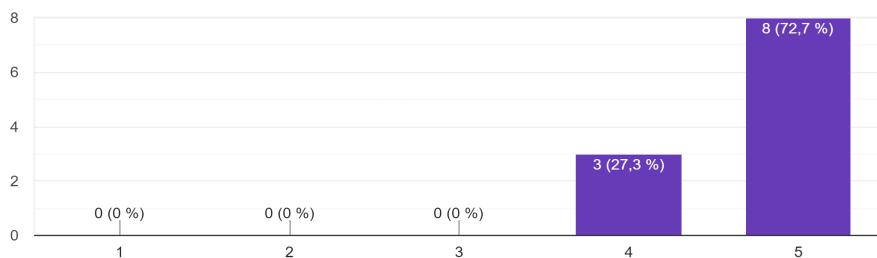


Figura 32: Experiencia general de los usuarios testeados al usar la herramienta
Fuente: Encuesta realizada durante el testeo (Elaboración propia).

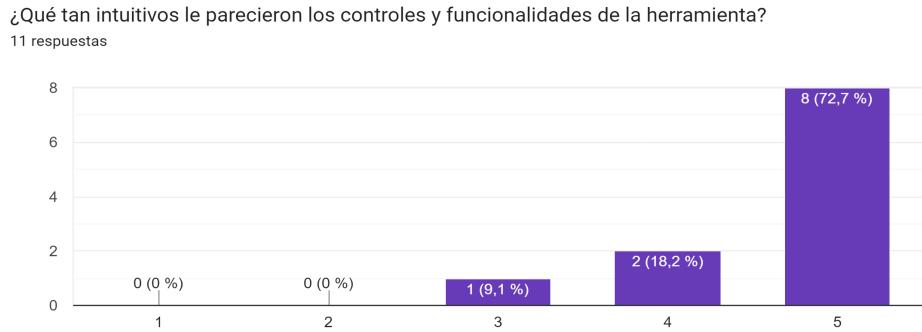


Figura 33: Que tan intuitiva encontraron la herramienta los usuarios testeados

Fuente: Encuesta realizada durante el testeo (Elaboración propia).

En caso de haber trabajado con Unity diseñando niveles/terrenos anteriormente, ¿le pareció que la herramienta mejoró su eficiencia en comparación a cuando no la usa?

10 respuestas

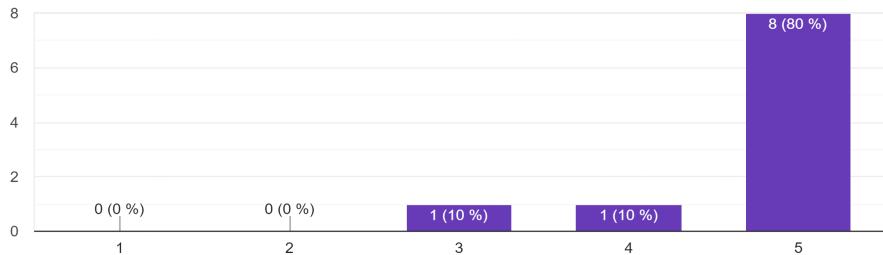


Figura 34: Comparación eficiencia usando la herramienta

Fuente: Encuesta realizada durante el testeo (Elaboración propia).

4.3. Análisis y Discusión de la Herramienta de Diseño de Niveles

El análisis de los resultados de las encuestas revela que la herramienta de diseño de niveles ha sido bien recibida por los usuarios, destacándose su facilidad de uso y la mejora en eficiencia que proporciona. Sin embargo, se identificaron áreas de mejora, principalmente en la interfaz de usuario y la necesidad de agregar funciones adicionales como deshacer.

Entre las principales dificultades mencionadas por los usuarios se encuentran la comprensión de las funciones de la herramienta y problemas de navegación. Algunos problemas y confusiones de los usuarios se debieron a que las tareas propuestas no eran explicativas por sí mismas. Por ejemplo, hubo confusión con términos específicos del proyecto, como “mi-

xer”, un elemento del proyecto “Mix: the forgotten”. Esto generó dificultades para usuarios externos al proyecto que no entendían qué era ni cómo debía ser usado.

La mayoría de las sugerencias de mejora se relacionaron con la documentación y la explicación de las funciones dentro de la interfaz de la herramienta. Las herramientas se desarrollaron inicialmente para el equipo de desarrollo de “Mix: the forgotten” y, durante su desarrollo, se trabajó con el feedback de ese equipo. Como resultado, la interfaz no contiene mucha información sobre lo que hace cada función. Cuando se testearon con personas externas que no tenían acceso a la documentación completa, surgieron dudas y falta de información sobre las funcionalidades. Esto se pudo evitar también con una mejor explicación previa, ya que al tratarse de usuarios externos trabajando con elementos de un proyecto al que son ajenos, hubiese sido útil una mayor contextualización.

Los tiempos para completar las tareas variaron entre los participantes, con algunos completándolas en menos de un minuto y otros tomando hasta cinco minutos. Esto sugiere que, aunque la herramienta es eficiente, hay una curva de aprendizaje que podría mejorarse con una mejor documentación y tutoriales.

La satisfacción general con la herramienta fue alta, y todos los participantes recomendaron la herramienta a otros desarrolladores de videojuegos. Las sugerencias para mejoras incluyeron una interfaz más intuitiva, mejor visualización entre niveles y la adición de funciones adicionales.

4.4. Conclusiones de la Validación de la Herramienta de Diseño de Niveles

La validación de la herramienta de diseño de niveles muestra que es una herramienta valiosa para los desarrolladores de videojuegos, especialmente para aquellos que trabajan con Unity. La herramienta fue bien recibida por los usuarios, quienes destacaron su facilidad de uso y la mejora en eficiencia que proporciona.

Sin embargo, se identificaron áreas de mejora que deben ser abordadas en futuras versiones de la herramienta. Estas incluyen una interfaz de usuario más intuitiva, la adición de una función de deshacer y una mejor visualización entre niveles.

La implementación de estas mejoras, junto con una mejor documentación y tutoriales, ayudará a reducir la curva de aprendizaje y aumentará aún más la satisfacción de los usuarios.

4.5. Validación de las Herramientas de Importación de Assets Gráficos y Creación de Items Craftables

En el caso de estas dos herramientas, al tratarse de implementaciones más específicas que fueron desarrolladas tomando como base el sistema de crafteo de un videojuego en desarrollo, es necesario llevar a cabo un proceso de validación distinto al aplicado para la herramienta de diseño de niveles. Para lo cual se tendrá en cuenta el uso de estas dos herramientas en el proyecto sobre el cual se ha trabajado.

El proceso de validación pensado para estas dos herramientas contempla dos cosas:

1. Evaluación de las herramientas por parte de un experto en el desarrollo de videojuegos.
2. El uso de las herramientas por los desarrolladores de Abstract Digital (desarrolladores a cargo del proyecto “Mix: the forgotten”).

4.5.1. Uso de la Solución en el Desarrollo de un Proyecto

Como bien se ha mencionado anteriormente, para el desarrollo de esta memoria se ha trabajado con el equipo de desarrollo del proyecto “Mix: the forgotten”, quienes ya han estado utilizando las herramientas desarrolladas en este trabajo y al menos 4 desarrolladores distintos han usado las herramientas. Estas herramientas han permitido la integración de nuevos objetos e ingredientes, mejorando significativamente la eficiencia y la calidad del desarrollo.

A partir del uso de ambas herramientas de apoyo se vio un gran progreso en el proyecto. En un principio, previo a la creación de las herramientas de apoyo para el proyecto, se tenían solo 3 objetos craftables implementados (un martillo, un espejo y una pistola de agua). Actualmente se han implementado diversos objetos craftables gracias a las herramientas desarrolladas, tales como: Escalera, Scooter, Hélice, Ventilador, Caña de pescar, Boomerang, Televisor, etc (video demostración de algunos de los objetos que han sido implementados). Asimismo, la herramienta de importación de assets y creación de nuevos ingredientes ha permitido incorporar mástil de 20 nuevos ingredientes al proyecto, como pilas, lápices, entre otros.

Para validar la eficiencia de ambas herramientas dedicadas a la implementación de nuevos elementos en el sistema de crafteo, se midió el tiempo necesario para completar tareas específicas (las mismas que se solicitaron en el testeo con usuarios externos) tanto usando las herramientas como sin usarlas, para poder comprobar si existe una mejora en los tiempos empleados, y así ver si las herramientas agilizan el trabajo de los desarrolladores. Los resultados se presentan en la Tabla 2.

Estos resultados demuestran una mejora significativa en la eficiencia y la agilidad del trabajo

Herramienta	Tarea	Con la Herramienta [s]	Sin la Herramienta [s]
Importación de Assets	Tarea 1	105	539
	Tarea 2	33	294
Creación de Objetos Craftables	Tarea 1	60	1339
	Tarea 2	66	422
	Tarea 3	56	622

Tabla 2: Tiempos de Realización de Tareas con y sin el Uso de las Herramientas

al utilizar las herramientas desarrolladas. A continuación, se tiene puede ver en porcentajes cuento tiempo se reduce gracias a la utilización de las herramientas (Tabla 3).

Herramienta	Tarea	Mejora en Tiempo [%]
Importación de Assets	Tarea 1	80.5
	Tarea 2	88.8
Creación de Objetos Craftables	Tarea 1	95.5
	Tarea 2	84.4
	Tarea 3	91.0

Tabla 3: Porcentaje de Mejora en los Tiempos de Realización de Tareas al Usar las Herramientas

4.5.2. Evaluación con experto

En esta segunda parte de la validación se solicitó a un desarrollador profesional, con 10 años de experiencia haciendo videojuegos comerciales, realizar una evaluación de la herramienta. Se diseñó para cada herramienta una prueba parecida a la que se llevó a cabo en el proceso de validación de la primera herramienta, con la diferencia de que el evaluador es un experto del área. Se idearon tareas para poder probar el correcto funcionamiento de las herramientas y se tiene un formulario para cada herramienta, en el cual el experto puede evaluar los distintos aspectos de las herramientas y también entregar retroalimentación.

- **Participante:** Desarrollador profesional de videojuegos, experto en la materia.
- **Procedimiento:** Estas herramientas, debido a su complejidad, fueron probadas en el contexto de un videojuego en desarrollo, específicamente el proyecto “Mix: the forgotten”. El experto tuvo que realizar tres tareas específicas para cada herramienta, ideadas para evaluar su integración y funcionalidad en un entorno de desarrollo real:
 - **Importación de Assets Gráficos:**
 - Implementar un nuevo ingrediente con forma de cilindro (se tiene a disposición el modelo 3D de un lápiz).

- Implementar nuevo ingrediente con forma de cubo (se tiene a disposición el modelo 3D de un dado).
 - Craftear en el juego un martillo con los ingredientes recientemente implementados.
- **Creación de Items Crafteables:**
 - Implementar una caja como nuevo item crafteable, cuya receta no requiera materiales ni funcionalidades específicas.
 - Implementar un letrero, cuya receta requiera materiales de madera.
 - Implementar un robot, cuya receta requiera materiales metálicos y que tengan la función de ser conductores eléctricos.
- **Evaluación:** Al finalizar las tareas, después de que el experto haya probado las herramientas, se le da acceso a una encuesta en Google Forms sobre su experiencia, similar a la utilizada para la herramienta de diseño de niveles. La encuesta evaluó aspectos como la facilidad de integración, eficiencia en el flujo de trabajo y sugerencias para mejoras.

4.6. Resultados de la evaluación con el experto de las Herramientas de Importación de Assets Gráficos y Creación de Items Crafteables

A continuación se presentan los resultados obtenidos de la encuesta para la herramienta de importación de assets gráficos:

4.6.1. Resultados de la Herramienta de Importación de Assets Gráficos

- **Facilidad de Uso:** La herramienta fue calificada con un 5 en todas las categorías por el experto evaluador, indicando una alta facilidad de uso.
- **Eficiencia:** El experto no pudo comparar la eficiencia con otros métodos debido a la falta de experiencia previa en tareas similares, por lo que no proporcionó una calificación en esta categoría. Sin embargo, los tiempos registrados para completar las tareas fueron muy bajos, lo que sugiere una gran eficiencia en el uso de la herramienta.
- **Satisfacción General:** El experto expresó una alta satisfacción general con la herramienta, destacando su facilidad y rapidez de uso.
- **Sugerencias para Mejoras:** La principal sugerencia fue la adición de tooltips en las variables para mejorar la comprensión.

Tarea	Éxito [%]	Tiempo [min]
Tarea 1	100 %	2.5
Tarea 2	100 %	1.5
Tarea 3	100 %	5

Tabla 4: Porcentaje de Éxito y Tiempos en la Realización de Tareas para la Herramienta de Importación de Assets Gráficos

Si desea ver imágenes de lo obtenido utilizando la herramienta durante las pruebas revisar los anexos incluidos al final del documento.

4.6.2. Resultados de la Herramienta de Creación de Items Crafteables

- **Facilidad de Uso:** La herramienta fue calificada con un 4 en todas las categorías por el experto evaluador, indicando una buena facilidad de uso pero con margen de mejora.
- **Eficiencia:** Similar a la herramienta de importación de assets gráficos, el experto no pudo proporcionar una comparación de eficiencia debido a la falta de experiencia previa en tareas similares. Sin embargo, los tiempos registrados para completar las tareas fueron muy bajos, lo que sugiere una gran eficiencia en el uso de la herramienta.
- **Satisfacción General:** El experto expresó una buena satisfacción general con la herramienta, destacando su rapidez y facilidad de uso para crear objetos crafteables.
- **Sugerencias para Mejoras:** Las sugerencias incluyeron la mejora en la visualización de materiales y funcionalidades de las partes, y la transmisión de la configuración de la grilla al prefab.

Tarea	Éxito [%]	Tiempo [min]
Tarea 1	100 %	10
Tarea 2	100 %	4
Tarea 3	100 %	6

Tabla 5: Porcentaje de Éxito y Tiempos en la Realización de Tareas para la Herramienta de Creación de Items Crafteables

Si desea ver imágenes de lo obtenido utilizando la herramienta durante las pruebas revisar los anexos incluidos al final del documento.

4.7. Análisis y Discusión de las Herramientas de Importación de Assets Gráficos y Creación de Items Crafteables

El uso de ambas herramientas en el desarrollo del proyecto “Mix: the forgotten” ha significado un apoyo increíble para los desarrolladores implicados. A partir de la implementación de estas herramientas se han implementado más de 20 objetos en el videojuego, y sobre todo se evidencia una gran mejora en el avance con respecto a la implementación de objetos/herramientas crafteables dentro del videojuego, ya que en un principio significaba un proceso muy complejo para los desarrolladores, tan así que solo existían 3 objetos crafteables, mientras que actualmente hay al menos 4 desarrolladores encargados de tal trabajo, y que, gracias a la utilización de las herramientas han podido implementar diversos nuevos objetos.

En cuanto al análisis de los resultados de la evaluación con el experto, revela que tanto la herramienta de importación de assets gráficos como la herramienta de creación de items crafteables han sido bien recibidas por el experto evaluador, destacándose su facilidad de uso y la mejora en eficiencia que proporcionan.

La herramienta de importación de assets gráficos mostró buenos resultados en términos de facilidad de uso y satisfacción general, aunque se sugirió la adición de tooltips para mejorar la comprensión de las variables. La eficiencia no pudo ser comparada por el experto debido a la falta de experiencia previa en tareas similares.

Por último, la herramienta de creación de items crafteables fue bien recibida, con una buena calificación en facilidad de uso y satisfacción general. Sin embargo, se identificaron dificultades en la visualización de materiales y funcionalidades de las partes, y la sugerencia de mejorar la transmisión de la configuración de la grilla al prefab.

4.8. Conclusiones de la Validación de las Herramientas de Importación de Assets Gráficos y Creación de Items Crafteables

La validación de las herramientas de importación de assets gráficos y creación de items crafteables mediante el uso en un proyecto, muestra que son herramientas valiosas para los desarrolladores de videojuegos, siendo de gran ayuda debido a que les permite agilizar procesos complejos, con una mejora muy importante en los tiempos que demoran y una menor cantidad de errores gracias a la automatización de ciertos procesos.

En cuanto a la evaluación de las herramientas por parte de un experto, las herramientas fueron bien recibidas por el experto evaluador, quien destacó su facilidad de uso y la mejora en eficiencia que proporcionan. Sin embargo, se identificaron áreas de mejora que deben ser abordadas en futuras versiones de las herramientas. Estas incluyen la adición de tooltips, la mejora en la visualización de materiales y funcionalidades de las partes, y la transmisión de

la configuración de la grilla al prefab.

A partir de todo esto, se concluye que la implementación de estas mejoras, junto con una mejor documentación y tutoriales, ayudará a reducir la curva de aprendizaje y aumentará aún más la satisfacción de los usuarios.

CAPÍTULO 5

CONCLUSIONES

5.1. Alcances y Limitaciones de la Propuesta de Solución

5.1.1. Alcances

El desarrollo de tres herramientas para Unity en este proyecto ha sido una experiencia enriquecedora, permitiendo identificar y superar diversos desafíos en el diseño y la implementación de soluciones para el desarrollo de videojuegos. Las herramientas desarrolladas han demostrado ser útiles y efectivas en el contexto de desarrollo de "Mix: the forgotten". A continuación, se detallan los alcances específicos de cada herramienta:

- **Herramienta de Diseño de Niveles:** Esta herramienta permite a los desarrolladores crear y manipular mapas 2D de manera eficiente, facilitando el proceso de diseño de niveles. Los usuarios pueden utilizar sus propios conceptos y creatividad para desarrollar sus ideas, mejorando significativamente el flujo de trabajo. Esta herramienta ha demostrado ser especialmente útil para diseñadores que buscan iterar rápidamente sobre sus ideas y probar diferentes configuraciones de niveles. Además, su implementación permite una rápida adaptación y modificación de los niveles, lo que es crucial en el desarrollo ágil de videojuegos. La herramienta ha sido valorada por su capacidad de integrar diferentes elementos de manera coherente, permitiendo a los diseñadores experimentar y ajustar sus diseños en tiempo real.
- **Herramienta de Importación de Assets Gráficos:** Facilita la integración de assets gráficos en el sistema de crafteo del juego, permitiendo una mayor personalización y reutilización de elementos gráficos dentro del proyecto. Esto permite a los desarrolladores trabajar de manera más eficiente bajo sus propias condiciones y capacidades, reduciendo el tiempo y esfuerzo necesarios para incorporar nuevos elementos visuales en el juego. La herramienta permite una importación fluida y sin complicaciones de diversos tipos de assets, lo cual es fundamental para mantener la consistencia visual y funcional del proyecto. Además, la capacidad de gestionar grandes cantidades de assets gráficos sin comprometer el rendimiento del juego ha sido un factor clave en su éxito.
- **Herramienta de Creación de Objetos Crafteables:** Permite la creación de nuevos objetos crafteables, generando automáticamente las recetas prefab necesarias. Esto agiliza el proceso de desarrollo y creación de contenido dentro del juego, permitiendo a los desarrolladores enfocarse en aspectos más creativos del desarrollo. La capacidad de generar recetas prefab de manera automática simplifica considerablemente el trabajo de los desarrolladores, especialmente en proyectos con una gran cantidad de items

crafteables. Esta herramienta ofrece una solución integral para la creación y gestión de objetos complejos, optimizando el flujo de trabajo de los desarrolladores. La flexibilidad y escalabilidad de la herramienta han sido destacadas como características fundamentales que permiten adaptarse a diferentes tipos de proyectos y necesidades de desarrollo.

5.1.2. Limitaciones

A pesar de los logros alcanzados, las herramientas desarrolladas también presentan algunas limitaciones que deben ser consideradas:

- **Herramienta de Diseño de Niveles:** La herramienta puede verse limitada en términos de interfaz en casos de mapas extremadamente grandes. Además, la complejidad de la interfaz podría representar un desafío para usuarios menos experimentados. Es importante considerar futuras mejoras en la interfaz de usuario para hacerla más accesible y fácil de usar. Además, la herramienta podría beneficiarse de funciones adicionales que permitan una personalización más detallada y específica de los niveles.
- **Herramienta de Importación de Assets Gráficos:** La integración de nuevos elementos requiere ajustes manuales, lo que podría limitar la usabilidad para usuarios sin experiencia técnica. Asimismo, el proceso de importación podría beneficiarse de una mayor automatización para mejorar la eficiencia. La herramienta actual, aunque funcional, puede ser optimizada para manejar un mayor volumen de assets gráficos con menos intervención manual. También sería beneficioso desarrollar una documentación más detallada que guíe a los usuarios en el proceso de importación.
- **Herramienta de Creación de Objetos Crafteables:** La herramienta podría beneficiarse de una interfaz más intuitiva y una documentación más exhaustiva para facilitar su uso por parte de desarrolladores menos experimentados. Además, la generación automática de recetas prefab podría ser mejorada para manejar casos más complejos y específicos. La herramienta podría incluir funcionalidades adicionales que permitan una mayor personalización de las recetas y una integración más fluida con otros sistemas del juego.

5.2. Resultados

Los resultados obtenidos a través de las pruebas con usuarios demostraron que las herramientas desarrolladas son eficaces en términos de usabilidad y funcionalidad. Los participantes completaron las tareas asignadas con éxito y proporcionaron feedback positivo sobre la interfaz y las características de cada herramienta.

5.2.1. Herramienta de Diseño de Niveles

La herramienta de diseño de niveles fue bien recibida por los usuarios, quienes destacaron su facilidad de uso y la capacidad de iterar rápidamente sobre diferentes configuraciones de niveles. Los usuarios pudieron completar las tareas asignadas de manera eficiente y encontraron que la herramienta mejoraba significativamente su flujo de trabajo en comparación con métodos tradicionales de diseño de niveles en Unity. Además, la posibilidad de visualizar y modificar los niveles en tiempo real fue considerada una ventaja significativa por los diseñadores.

5.2.2. Herramienta de Importación de Assets Gráficos

La herramienta de importación de assets gráficos facilitó la integración de nuevos elementos visuales en el juego, permitiendo a los desarrolladores personalizar y reutilizar assets gráficos de manera eficiente. Los desarrolladores valoraron la capacidad de la herramienta para simplificar el proceso de importación y destacaron su utilidad en proyectos con una gran cantidad de assets gráficos. La capacidad de la herramienta para mantener la coherencia visual y funcional del juego, incluso con un gran volumen de assets, fue uno de los aspectos más valorados por los usuarios.

5.2.3. Herramienta de Creación de Objetos Crafteables

La herramienta de creación de objetos crafteables permitió a los desarrolladores generar recetas prefab de manera automática, agilizando el proceso de creación de contenido. Los desarrolladores destacaron la utilidad de la herramienta para manejar la complejidad de los sistemas de crafteo y la facilidad con la que podían integrar nuevos objetos en el juego. La herramienta también fue valorada por su capacidad de generar recetas complejas de manera rápida y eficiente, permitiendo a los desarrolladores centrarse en aspectos más creativos y estratégicos del desarrollo del juego.

5.3. Validez de los Objetivos Formulados

Los objetivos formulados al inicio del proyecto fueron validados a través de diversas pruebas y evaluaciones. Las tres herramientas cumplían con los requisitos iniciales de proporcionar soluciones intuitivas y eficientes para el desarrollo en Unity. Además, se lograron implementar las funcionalidades propuestas de manera exitosa, lo que confirma la validez de los objetivos establecidos. La validación de los objetivos a través de pruebas exhaustivas con usuarios ha demostrado que las herramientas desarrolladas no solo son funcionales, sino también valiosas en el contexto del desarrollo de videojuegos.

5.4. Contribuciones y Aplicaciones

Las principales contribuciones de este trabajo incluyen:

- **Desarrollo de Herramientas:** Se desarrollaron tres herramientas prácticas y útiles. Una de ellas, el editor de niveles puede ser un aporte a la comunidad de desarrolladores de videojuegos que usan Unity y en el caso de las herramientas más específicas desarrolladas para la empresa Abstract Digital.
- **Mejoras en el Flujo de Trabajo:** Las herramientas desarrolladas permitieron una mayor eficiencia y creatividad en la creación de videojuegos, reduciendo el tiempo necesario para completar tareas complejas y permitiendo a los desarrolladores enfocarse en aspectos más creativos del desarrollo. La optimización del flujo de trabajo ha sido uno de los aspectos más significativos, permitiendo a los desarrolladores realizar iteraciones rápidas y mejorar la calidad del producto final.
- **Aporte a la Documentación:** El proyecto contribuyó a la documentación y recursos disponibles para el desarrollo de herramientas en Unity, identificando áreas que requieren mayor investigación y desarrollo. La creación de guías y tutoriales detallados ha sido fundamental para apoyar a los desarrolladores en el uso de las nuevas herramientas.

5.5. Impacto y Aporte

El impacto de estas herramientas en la comunidad de desarrollo de videojuegos es significativo, facilitando diversos aspectos del proceso de desarrollo y reduciendo el tiempo necesario para completar tareas complejas. Esto permitió especializarse en el área de creación de herramientas para Unity y crear software para apoyar a otros desarrolladores en su proceso. Además, estas herramientas tienen el potencial de ser mejoradas y ampliadas, proporcionando una base sólida para futuros desarrollos. La implementación de estas herramientas en el proyecto "Mix: the forgotten" ha demostrado su utilidad y ha permitido a los desarrolladores integrar nuevos objetos e ingredientes de manera eficiente, mejorando la calidad y el alcance del juego.

5.6. Recomendaciones y Futuras Ampliaciones

Para quienes estén interesados en profundizar en el tema, se recomiendan las siguientes líneas de investigación y desarrollo:

- **Mejorar la Disposición de la Interfaz:** Mejorar la disposición de la interfaz de la herramienta de diseño de niveles para trabajar de manera más eficiente con mapas de gran tamaño, optimizando la disposición y la gestión de recursos en la interfaz.
- **Ampliar la Documentación:** Ampliar la documentación y recursos de soporte para usuarios de las tres herramientas, proporcionando guías más detalladas y ejemplos prácticos. La creación de tutoriales en video y la documentación interactiva pueden ser estrategias efectivas para apoyar a los usuarios.
- **Integrar Funciones Avanzadas:** Integrar más funciones avanzadas que permitan una mayor personalización y flexibilidad en el diseño de niveles, importación de assets gráficos y creación de objetos crafteables. La inclusión de herramientas de análisis y depuración puede ayudar a los desarrolladores a identificar y resolver problemas de manera más eficiente.
- **Realizar Pruebas Adicionales:** Realizar pruebas adicionales con una mayor diversidad de usuarios, incluyendo desarrolladores profesionales, para obtener un feedback más amplio y detallado que permita identificar áreas de mejora y nuevas funcionalidades a implementar. La colaboración con estudios de desarrollo de videojuegos puede proporcionar insights²⁴ valiosos y oportunidades de mejora continua.

Estas recomendaciones y futuras ampliaciones no solo contribuirán a mejorar las herramientas desarrolladas en este proyecto, sino que también proporcionarán una base sólida para futuras investigaciones y desarrollos en el ámbito de las herramientas para el desarrollo de videojuegos en Unity. El potencial de estas herramientas para evolucionar y adaptarse a las necesidades cambiantes de la industria subraya su relevancia y la importancia de continuar invirtiendo en su desarrollo y mejora.

La implementación de estas mejoras junto con una mejor documentación y tutoriales ayudará a reducir la curva de aprendizaje y aumentará aún más la satisfacción de los usuarios. La colaboración con la comunidad de desarrolladores y la recopilación continua de feedback serán esenciales para mantener las herramientas actualizadas y alineadas con las necesidades reales de la industria del desarrollo de videojuegos.

²⁴Los “insights” son observaciones y comprensiones profundas obtenidas del análisis de datos, que proporcionan información valiosa y accionable para la toma de decisiones estratégicas.

ANEXOS

5.1. Imágenes de las pruebas realizadas con usuarios

5.1.1. Herramienta de diseño de niveles

A continuación se muestran algunas imágenes extraídas de las pruebas en el proceso de validación junto con los usuarios.

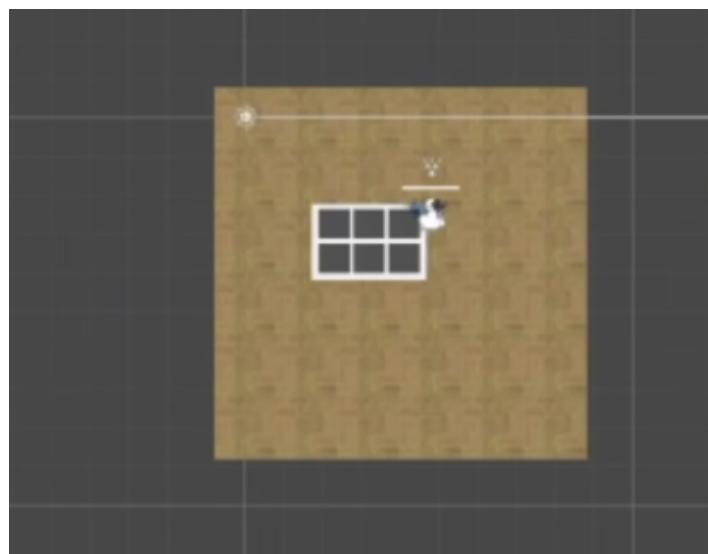


Figura 35: Tarea 1 - Prueba herramienta diseño de niveles.

Fuente: Videos del testeo realizado con usuarios (Elaboración propia).

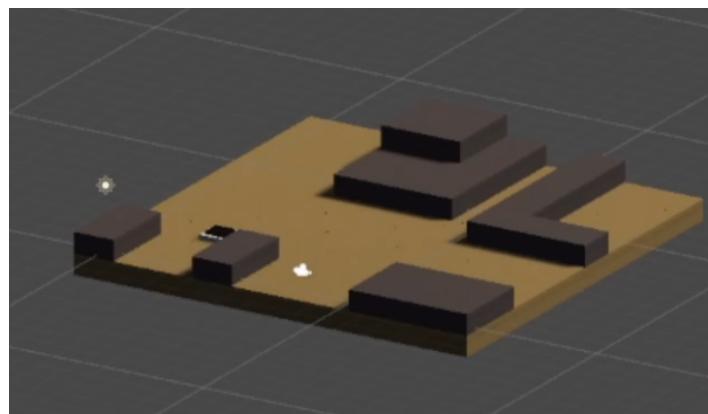


Figura 36: Tarea 2 - Prueba herramienta diseño de niveles.

Fuente: Videos del testeo realizado con usuarios (Elaboración propia).

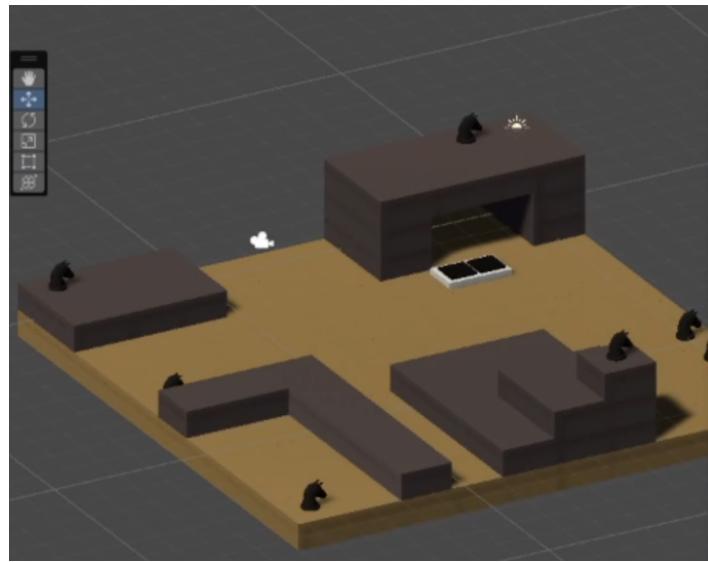


Figura 37: Tarea 3 - Prueba herramienta diseño de niveles.

Fuente: Videos del testeo realizado con usuarios (Elaboración propia).

5.1.2. Herramienta de importación de assets y creación de ingredientes

A continuación se muestran algunas imágenes extraídas de las pruebas en el proceso de validación junto a los usuarios.

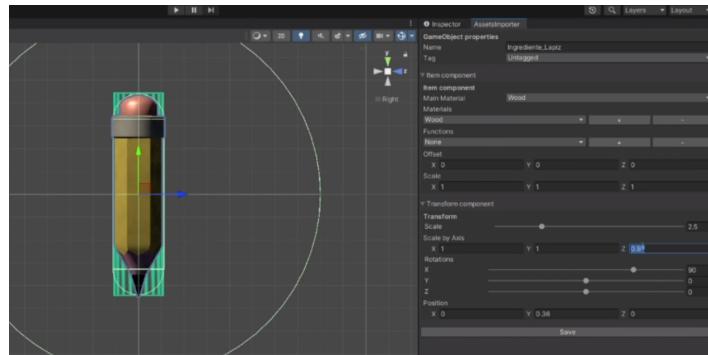


Figura 38: Tarea 1 - Prueba de la herramienta importación de assets gráficos y creación de nuevos ingredientes.

Fuente: Videos del testeo realizado con usuarios (Elaboración propia).

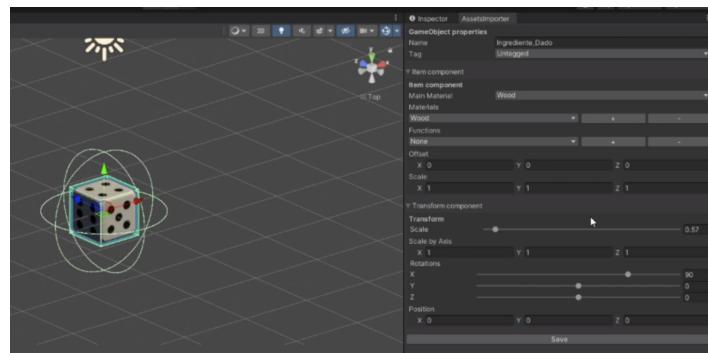


Figura 39: Tarea 2 - Prueba de la herramienta importación de assets gráficos y creación de nuevos ingredientes.

Fuente: Videos del testeo realizado con usuarios (Elaboración propia).



Figura 40: Tarea 3 - Prueba de la herramienta importación de assets gráficos y creación de nuevos ingredientes.

Fuente: Videos del testeo realizado con usuarios (Elaboración propia).

5.1.3. Herramienta de creación de nuevos objetos crafteables

A continuación se muestran algunas imágenes extraídas de las pruebas en el proceso de validación junto a los usuarios.

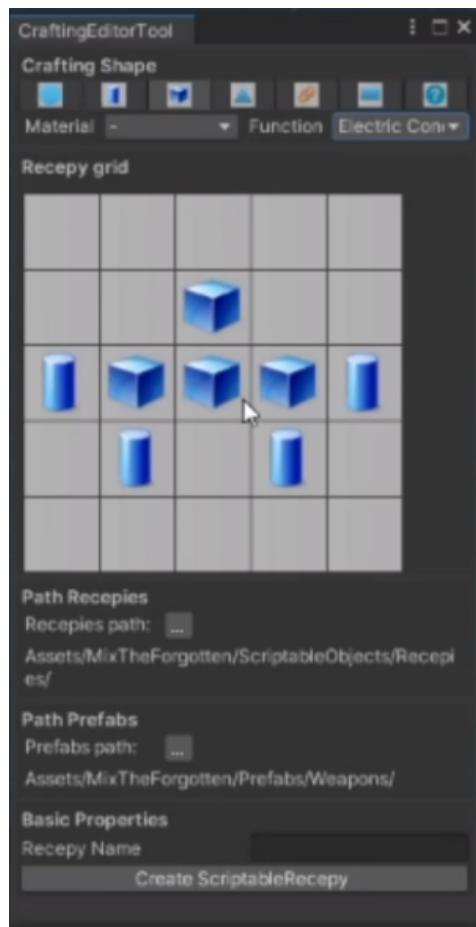


Figura 41: Tarea 3 - Prueba de la herramienta importación de assets gráficos y creación de nuevos ingredientes.

Fuente: Videos del testeo realizado con usuarios (Elaboración propia).

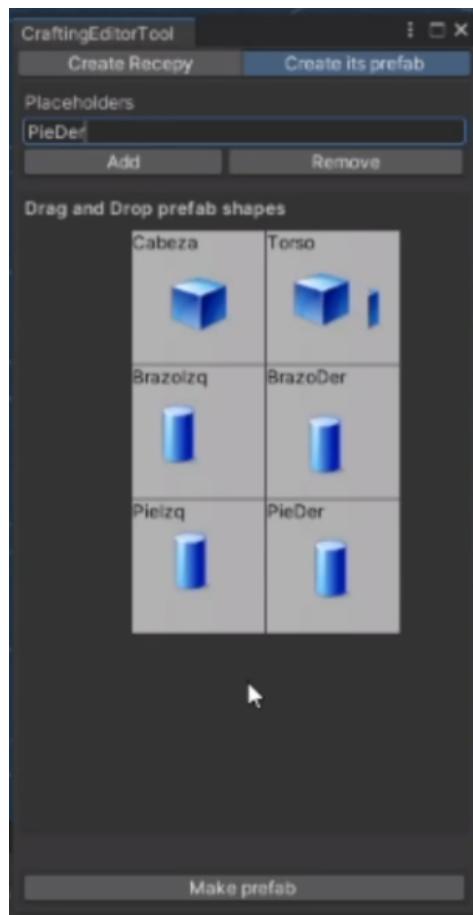


Figura 42: Tarea 3 - Prueba de la herramienta importación de assets gráficos y creación de nuevos ingredientes.

Fuente: Videos del testeo realizado con usuarios (Elaboración propia).

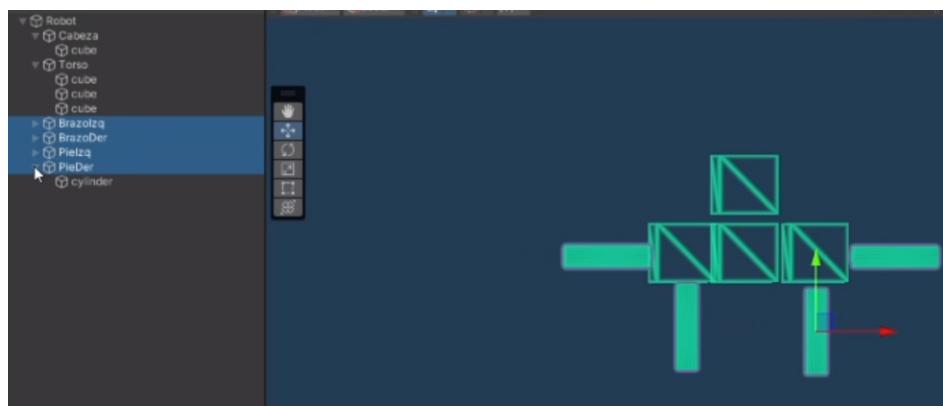


Figura 43: Tarea 3 - Prueba de la herramienta importación de assets gráficos y creación de nuevos ingredientes.

Fuente: Videos del testeo realizado con usuarios (Elaboración propia).

5.2. Resultados de las encuestas - Herramienta de diseño de niveles

Aquí puede observar los resultados de la encuesta con mayor detalle.

5.2.1. Gráficos

A continuación se presentan algunos gráficos que se obtuvieron con los resultados de la encuesta, con los que se pueden ver más detalladamente los resultados de la encuesta para la validación de esta herramienta.

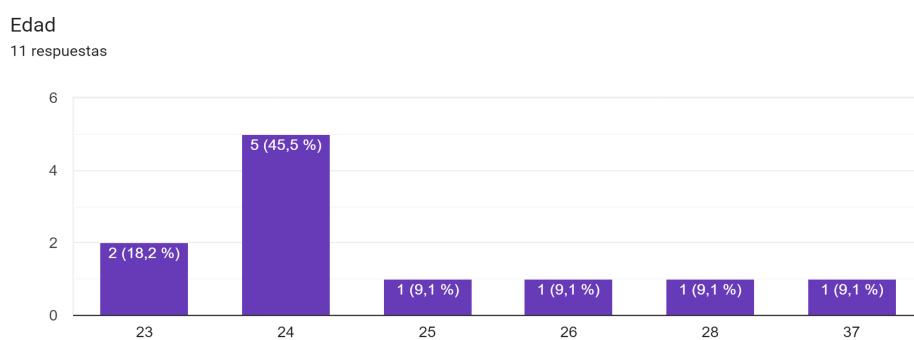


Figura 44: Edad de los usuarios testeados

Fuente: Encuesta realizada durante el testeo (Elaboración propia).

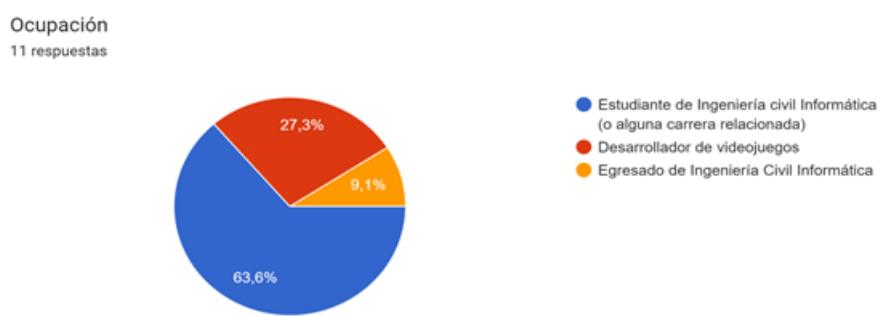


Figura 45: Ocupación de los usuarios testeados

Fuente: Encuesta realizada durante el testeo (Elaboración propia).

¿Qué tanta experiencia tiene desarrollando juegos con Unity?

11 respuestas

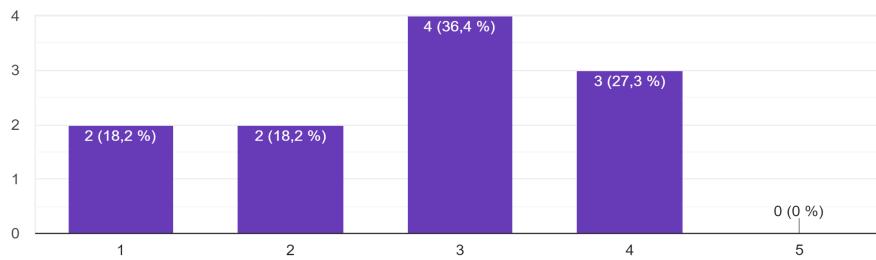


Figura 46: Experiencia desarrollando videojuegos con Unity

Fuente: Encuesta realizada durante el testeо (Elaboración propia).

¿Cómo calificaría su experiencia general con la herramienta de diseño de niveles?

11 respuestas

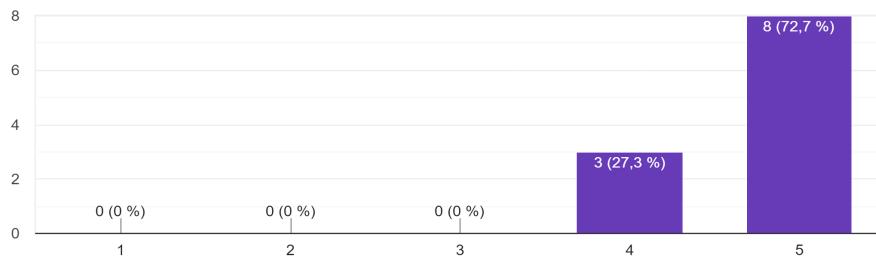


Figura 47: Experiencia general de los usuarios testeados al usar la herramienta

Fuente: Encuesta realizada durante el testeо (Elaboración propia).

¿Le resultó fácil de aprender a usar la herramienta?
11 respuestas

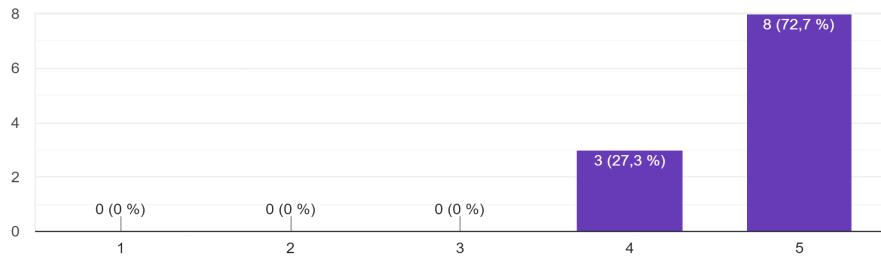


Figura 48: Facilidad al momento de aprender a usar la herramienta según los usuarios testeados

Fuente: Encuesta realizada durante el testeo (Elaboración propia).

¿Cómo calificaría la usabilidad de la interfaz de usuario de la herramienta?
11 respuestas

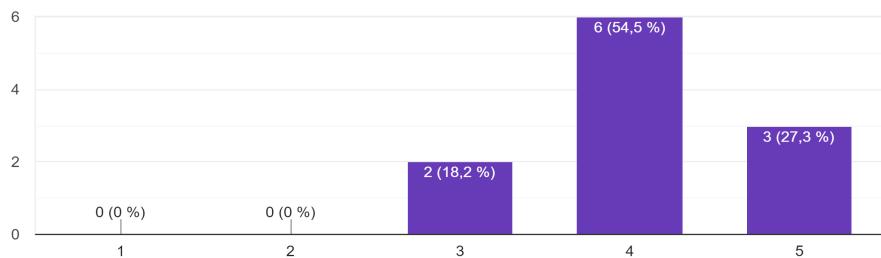


Figura 49: Usabilidad de la interfaz de la herramienta según los usuarios testeados

Fuente: Encuesta realizada durante el testeo (Elaboración propia).

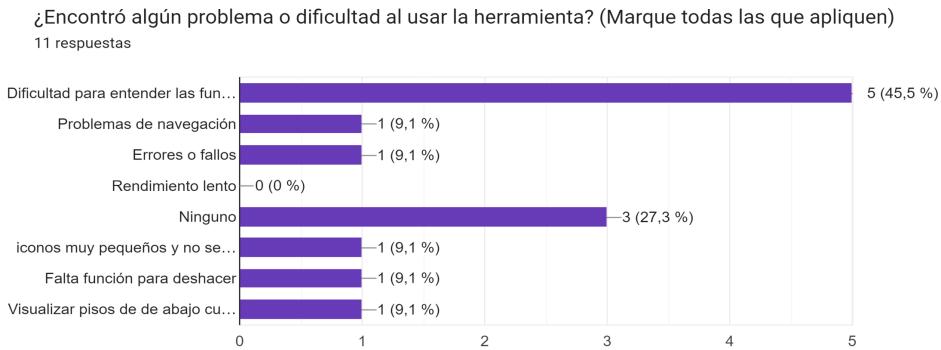


Figura 50: Problemas en el testeo por parte de los usuarios testeados

Fuente: Encuesta realizada durante el testeo (Elaboración propia).

¿Qué tan intuitivos le parecieron los controles y funcionalidades de la herramienta?
11 respuestas

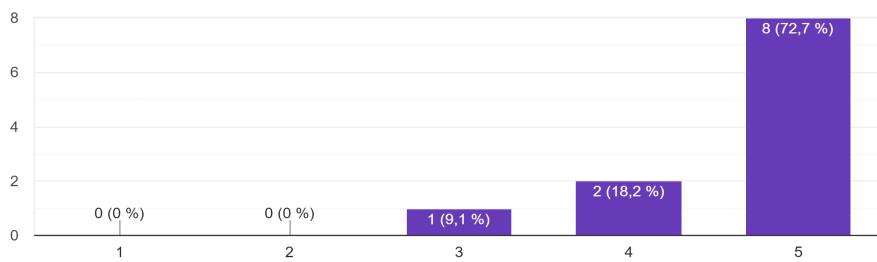


Figura 51: Que tan intuitiva encontraron la herramienta los usuarios testeados

Fuente: Encuesta realizada durante el testeo (Elaboración propia).

En caso de haber trabajado con Unity diseñando niveles/terrenos anteriormente, ¿le pareció que la herramienta mejoró su eficiencia en comparación a cuando no la usa?
10 respuestas

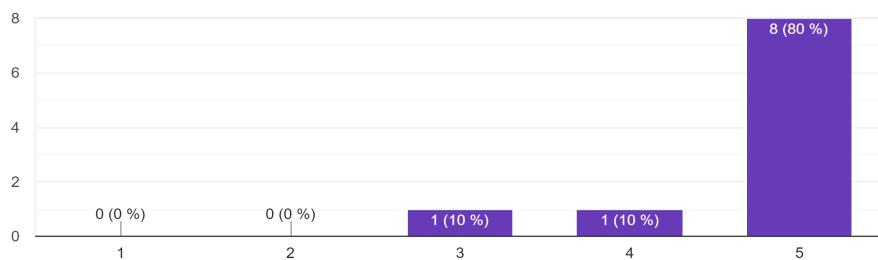


Figura 52: Comparación eficiencia usando la herramienta
Fuente: Encuesta realizada durante el testeo (Elaboración propia).

¿Recomendaría esta herramienta a otros diseñadores/desarrolladores de videojuegos?
11 respuestas

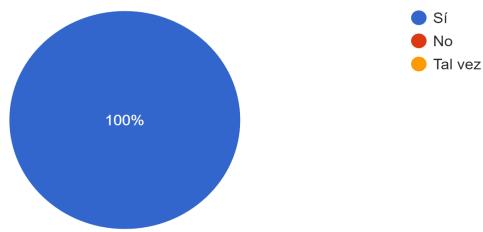


Figura 53: Recomienda el usuario la herramienta a otros desarrolladores de videojuegos
Fuente: Encuesta realizada durante el testeo (Elaboración propia).

5.2.2. Dificultades del usuario al realizar tareas

Respuestas de los usuarios testeados a la pregunta: “¿Qué dificultades ha tenido con la utilización de la herramienta al momento de realizar las tareas?”

- Ninguna.
- No veía algunas cosas reflejadas en la vista de la herramienta entonces no sabia si se habían agregado correctamente.
- Me costó un poco entender la funcionalidad de algunos botones.
- No entendía muy bien para qué servía cada botón

- Posición de botones de agregar y quitar filas/columnas. Undo en la herramienta de bloques. No poder cambiar de nivel cuando se ponen enemigos.
- Entender que el grid que crea al inicio es la base y no esta dentro del juego.
- En la primer tarea no sabia bien a que se refería con un “Mixer” ni como debía colocarlo, pero luego navegando por la herramienta encontré lo que buscaba.
- Ninguna, pude realizar todas las tareas sin problema.
- Dificultades como tal no tuve, pero hay cosas de la interfaz que podrían mejorar.
- No entendí bien que era la plataforma de mezcla.

5.2.3. Retroalimentación abierta

Respuestas de los usuarios testeados a la pregunta: “¿Qué le gustó más de la herramienta?”

- Que viene con elementos listos, como diferentes tipos de terreno entonces es llegar y ocupar.
- Me gustó la facilidad de uso que se tiene al crear el terreno.
- La facilidad para poder hacer niveles y darle texturas al piso.
- Lo intuitiva que es en su uso.
- Fácil de usar, resultados inmediatos.
- La facilidad que te permite para crear niveles, ahorrando harto tiempo en detalles.
- Lo rápido que puedo ir creando el escenario, la diferencia a trabajar sin la herramienta muchísimo.
- Me facilitó muchísimo las tareas que se pedían, sin la herramienta me tardaría una eternidad en realizar las tareas que me pidieron y con la herramienta fue muy fácil todo.
- La eficiencia, ya que la diferencia entre realizar las tareas con la herramienta y sin la herramienta es muchísima. También me encantó que al apretar en una tablita 2D iban apareciendo los elementos 3D en simultaneo automáticamente, encuentro genial que no tenga que ir guardando o generando el terreno a medida que termine el escenario.
- Lo rápido y eficiente que se torna crear el terreno, además de lo flexible en cuanto a que se puedan integrar nuevos elementos a la herramienta o modificar los que se tienen.

Respuestas de los usuarios testeados a la pregunta: “¿Qué mejoraría de la herramienta?”

- Agregaría una interfaz para poder agregar texturas propias con mayor facilidad.
- La interfaz de la herramienta.
- Agregar infoboxes.
- Función para deshacer. Posibilidad de remover columnas/filas desde otros extremos. Visualización de los mapas almacenados para cargar.
- Mejor visualización entre niveles.
- Que te muestre dentro de tus carpetas los guardados que tienes.
- Si bien pude entender los botones y funciones de la herramienta, creo que la interfaz podría mejorar agregándose algunos cuadros de texto que explicativos o elementos de diseño de interfaces que permitan explicar que hace cada botón o parte de la herramienta.
- Creo que podría mejorar el diseño de la interfaz.
- Cuando agregaba elementos adicionales (como el Mixer) al terreno, en la interfaz 2D no se apreciaba la posición en que se agregó, a diferencia de los bloques del suelo que la grilla si me muestra donde voy colocando los bloques.
- La interfaz de usuario, hay elementos que con otros nombres puede ser más intuitivo. O que dentro de la herramienta haya más texto o explicaciones sobre la función de cada elemento.

Comentarios adicionales entregados por algunos usuarios:

- Me gustó la herramienta y le veo potencial :).
- Le veo muy buenos usos en juegos de plataformas.
- Que en un proyecto 3D me complico mucho la creación de objetos y todo el entorno, esto me hubiera salvado la vida, así que me gusta la idea.
- La herramienta realmente es muy útil y eficiente, me permite acelerar muchísimo el diseñar un escenario en Unity.
- Me ayudó muchísimo a crear escenarios y me encantaría seguir probándola con algún proyecto.

- Me gustó mucho la herramienta y me ha sido muy útil, definitivamente la ocuparía para realizar un juego de plataforma 3D. Además, la opción de poder agregar tus propios bloques y prefabs es una ventaja increíble con respecto a otras herramientas que he utilizado en Unity.
- La herramienta esta súper buena, ayuda mucho a crear escenarios en mucho menos tiempo. Además de ser más flexible que otras herramientas disponibles en la tienda de Unity que he probado.

5.3. Detalles de las Mediciones de Tiempos en el uso de las herramientas

En la validación de la propuesta se realizaron mediciones detalladas del tiempo necesario para completar tareas específicas tanto usando las herramientas desarrolladas como sin usarlas. Para esas mediciones se realizaron las mismas tareas que se les pidió realizar a los usuarios externos:

5.3.1. Herramienta de Importación de Assets

- **Tarea 1:** Implementar un nuevo ingrediente con forma de cilindro (modelo 3D de un lápiz).
- **Tarea 2:** Implementar nuevo ingrediente con forma de cubo (modelo 3D de un dado).

5.3.2. Herramienta de Creación de Objetos Craftables

- **Tarea 1:** Implementar una caja como nuevo item craftable, cuya receta no requiera materiales ni funcionalidades específicas.
- **Tarea 2:** Implementar un letrero, cuya receta requiera materiales de madera.
- **Tarea 3:** Implementar un robot, cuya receta requiera materiales metálicos y que tengan la función de ser conductores eléctricos.

5.4. Enlaces

- Encuesta herramienta diseño de niveles
- Paquete de Unity con la herramienta de diseño de niveles
- Video demostración de nuevos objetos implementados usando las herramientas

- Página de Steam²⁵ del videojuego “Mix: the forgotten”

²⁵Plataforma de distribución digital de videojuegos y software desarrollada por Valve Corporation.

REFERENCIAS BIBLIOGRÁFICAS

- [Carrasco, 2018] Carrasco, A. (2018). ¿qué es un motor de videojuegos? URL: <https://blogs.upm.es/observatoriogate/2018/07/04/que-es-un-motor-de-videojuegos/>. Recuperado el 8 de diciembre de 2022.
- [ComputerCareers, 2022] ComputerCareers (2022). Is game development hard? skills required learning tips. URL: <https://www.computercareers.org/is-game-development-hard-skills-required-learning-tips/>. Recuperado el 1 de diciembre de 2022.
- [Cruscuolo, 2021] Cruscuolo, I. (2021). ¿qué son el tileset y el tilemap en el desarrollo de videojuegos? URL: <https://www.domestika.org/es/blog/7053-que-son-el-tileset-y-el-tilemap-en-el-desarrollo-de-videojuegos>. Recuperado el 28 de Agosto de 2023.
- [freeCodeCamp, 2019] freeCodeCamp (2019). What is game development. URL: <https://www.freecodecamp.org/news/what-is-game-development/>. Recuperado el 8 de diciembre de 2022.
- [French, 2022] French, J. (2022). Scriptable objects in unity (how and when to use them). URL: <https://gamedevbeginner.com/scriptable-objects-in-unity/>. Recuperado el 5 de diciembre de 2022.
- [Hipple, 2020] Hipple, R. (2020). Three ways to architect your game with scriptableobjects. URL: <https://unity.com/how-to/architect-game-code-scriptable-objects>. Recuperado el 6 de noviembre de 2022.
- [Hussain et al., 2020] Hussain, A., Shakeel, H., Hussain, F., Uddin, N., y Ghouri, T. L. (2020). Unity game development engine: A technical survey. *Univ. Sindh J. Inf. Commun. Technol.*, 4(2):73-81.
- [Jin, 2020] Jin, J. (2020). Video game development with unity.
- [KinematicSoup, 2016] KinematicSoup (2016). Unity development tools that help create games faster. URL: <https://www.kinematicsoup.com/news/2016/5/6/unity-development-tools-that-help-create-games-faster>. Recuperado el 7 de diciembre de 2022.
- [King, 2015] King, A. (2015). 5 approaches to crafting systems in games (and where to use them). URL: <https://gamedevelopment.tutsplus.com/articles/5-approaches-to-crafting-systems-in-games-and-where-to-use-them--cms-22628>. Recuperado el 10 de diciembre de 2022.
- [Kok, 2021] Kok, B. (2021). Object spawner tool using editortool and scriptableObject. En *Beginning Unity Editor Scripting*, pp. 111-134. Springer.

[Kunal, 2018] Kunal, T. (2018). Scriptable objects in unity. URL: <https://medium.com/@kunaltandon.kt/scriptable-objects-in-unity-8add2c7938d3>. Recuperado el 6 de diciembre de 2022.

[Serpa, 2022] Serpa, Y. R. (2022). Fast prototyping in unity: Scriptable objects. URL: <https://blog.logrocket.com/fast-prototyping-unity-scriptable-objects/>. Recuperado el 5 de diciembre de 2022.

[Skoczylas, 2022] Skoczylas, N. (2022). Most common software used for game development. URL: <https://indiegamecloud.com/most-common-software-used-for-game-development/>. Recuperado el 3 de diciembre de 2022.

[Talbert, 2019] Talbert, L. (2019). Building a level editor in unity. URL: <https://www.red-gate.com/simple-talk/development/dotnet-development/building-a-level-editor-in-unity/>. Recuperado el 6 de diciembre de 2022.