



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

GRADO EN DISEÑO Y DESARROLLO DE VIDEOJUEGOS

Curso Académico 2023/2024

Trabajo Fin de Grado

**DISEÑO E IMPLEMENTACIÓN DE UN PERSONAJE PARA EL
VIDEOJUEGO RIVALS OF AETHER**

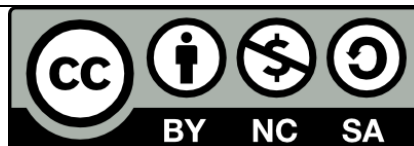
Autor: Javier Sánchez Muñoz

Directores: Julio Guillén García



Universidad
Rey Juan Carlos

Escuela Técnica Superior
Ingeniería Informática



Esta obra está bajo licencia Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/4.0/>.



Universidad
Rey Juan Carlos

Escuela Técnica Superior
Ingeniería Informática

A mi familia y amigos por el ánimo que me han dado durante este trabajo

A mi tutor por la ayuda y guía que me ha brindado durante todo el proceso

A la comunidad de Rivals of Aether en Discord y en especial a Frikinin, Aether's Asylum, Spadster y Onyx por ayudarme con cada problema que tenía y darme el feedback continuo para que el personaje haya llegado a ser algo de lo que pueda estar orgulloso.

Gracias



Resumen

En este trabajo se tiene como objetivo diseñar y desarrollar un personaje para un juego de lucha, específicamente un juego de lucha en plataformas, centrándose así en todos los pasos necesarios para la creación e implementación de este para un juego ya sacado al mercado, en forma de modificación de la comunidad.

Para cumplir este objetivo se ha realizado un análisis exhaustivo del género y del videojuego elegido para el proyecto. Además de esto, se ha llevado a cabo un largo proceso de diseño, animación e implementación del personaje para que sea jugable dentro de dicho videojuego. Además de esto, se destaca la importancia de subir el contenido creado a páginas conocidas en las que varios jugadores anónimos puedan descargarse dicha modificación y dar retroalimentación para poder realizar mejoras al personaje de manera continua ya que permite una fase de balanceo y rediseño continua.

Palabras Clave: **Mod, modificación, Steam Workshop, Rivals of Aether, Diseño de juegos, Aseprite, Pixel art, Gamedev, juego de lucha, juegos de lucha en plataformas, animación 2D, videojuego.**

Abstract

In this project, the objective is to design and develop a character for a fighting game, specifically a platform fighter, focusing on all the necessary steps for the creation and implementation of this character for a game already on the market, by creating a *mod*.

To achieve this goal, an exhaustive analysis of the genre and the chosen video game for the project has been carried out. In addition to this, a lengthy process of character design, animation, and implementation has been undertaken to make it playable within said videogame. Furthermore, it's highlighted the importance of uploading the created content to well-known websites where various anonymous players can download this mod and provide feedback to make continuous improvements to the character as it allows for a continuous phase of balancing and redesign.

Keywords: Mod, modification, Steam Workshop, Rivals of Aether, Game Design, Aseprite, Pixel art, Gamedev, Fighting Games, Platform Fighters, 2D animation, videogame.



Índice de contenidos

Índice de contenidos	6
Índice de ilustraciones	8
Glosario	9
Capítulo 1 Introducción	11
1.1 Descripción del problema y motivaciones	11
1.2 Objetivos	12
1.2.1 <i>Objetivo general</i>	12
1.2.2 <i>Objetivos específicos</i>	12
1.3 Planificación y metodología inicial.....	13
1.4 Metodología	13
Capítulo 2 Marco teórico	15
2.1 Análisis del juego y de su género.....	15
2.1.1 <i>Análisis de los juegos de lucha 2D</i>	15
2.1.2 <i>Análisis de los juegos de lucha en plataformas</i>	17
2.1.3 <i>Análisis de Rivals of Aether</i>	19
2.2 Casos de estudio	20
2.2.1 <i>Personajes ágiles en platform fighters</i>	21
2.2.2 <i>Grapplers en platform fighters</i>	22
2.2.3 <i>Conclusiones sobre el diseño</i>	25
2.3 Animaciones de personajes interactivos para videojuegos	26
Capítulo 3 Diseño de artes	29



3.1	Inspiración visual, referencias y moodboard.....	29
3.2	Arte conceptual	31
3.3	Trasfondo del personaje	31
3.4	Flujo de trabajo: Sprites y animaciones	32
3.5	Interfaz dentro del juego	35
3.6	Pruebas de colores y aspectos alternativos.....	36
Capítulo 4 Descripción informática.....		37
4.1	Análisis de requisitos.	37
4.2	Diseño.....	38
4.3	Implementación	40
4.4	Despliegue del personaje a la plataforma de descargas	46
Capítulo 5 Validación.....		47
5.1	Resultado final.....	47
5.2	Feedback.....	48
Capítulo 6 Conclusiones.....		51
6.1	Análisis de objetivos	51
6.2	Lecciones aprendidas	52
6.3	Líneas futuras	54
Bibliografía		55
Ludografía.....		56
Anexo I Hojas de <i>sprites</i>		57

Índice de ilustraciones

Ilustración 1. Diagrama de Gantt de la planificación	13
Ilustración 2. Fases, tareas y duraciones planificadas.....	14
Ilustración 3. Captura de pantalla de Street Fighter II: The World Warrior	16
Ilustración 4. Personajes de Guilty Gear	17
Ilustración 5. Imagen de Rivals of Aether.....	18
Ilustración 6. Maypul y Etalus de Rivals of Aether	19
Ilustración 7. 4 personajes de Rivals of Aether	20
Ilustración 8. Wrastor, de Rivals of Aether	22
Ilustración 9. Incineroar y Donkey Kong de Super Smash Bros. Ultimate	24
Ilustración 10. Ejemplos de sombras creadas	29
Ilustración 11. Moodboard.....	30
Ilustración 12. Turn-Around del personaje	31
Ilustración 13. Parte de la tabla de datos usada.	33
Ilustración 14. Spritesheet de la animación "Idle", reorganizada en filas para que se vea mejor.....	34
Ilustración 15. Spritesheet del "Down Air"	34
Ilustración 16. Vista de capas y frames del archivo <code>fstrong_stripX</code>	35
Ilustración 17. Aspecto de la interfaz del personaje dentro de partida.....	35
Ilustración 18. Ejemplo de un conjunto de colores alternativo.	36
Ilustración 19. Material promocional del personaje explicando algunos de sus ataques únicos.....	40
Ilustración 20. Nombres posibles para los ataques que se usen.	42
Ilustración 21. Ejemplo de cómo funciona el código de <code>colors.gml</code>	43
Ilustración 22. Archivo <code>nspcial.gml</code> dividido en las partes explicadas anteriormente.	45
Ilustración 23. Resultados de búsqueda del personaje.....	47
Ilustración 24. Miniatura del video de demostración.	48
Ilustración 25. Total de suscriptores acumulados a lo largo de los días.	49
Ilustración 26. Estadísticas generales del artículo en el día 08/07/2024.....	49
Ilustración 27. Comentario subido a la página pública de Steam del mod por uno de los jugadores .	50



Glosario

<i>Hitbox</i>	Caja invisible que puede tener cualquier tamaño y que representa la zona donde un ataque golpea. Este se considera que ha golpeado si tiene alguna intersección con la hurtbox de un personaje.
<i>Hurtbox</i>	Caja invisible que puede tener cualquier tamaño y que representa la zona donde un personaje puede ser golpeado. Se considera que un personaje es golpeado cuando tiene alguna intersección con la hitbox de un ataque.
<i>Sprite Sheet</i>	Imagen que representa una animación 2D. Dentro de esta imagen hay varias imágenes o sprites que representan cada uno de los frames principales que serán utilizados para la animación final.
<i>Steam Workshop</i>	Plataforma de subida y descarga de contenido creado por la comunidad para juegos dentro de la plataforma Steam.
<i>Fighting Game</i>	Nombre internacional dado al género de juegos basado en el enfrentamiento de varios personajes se enfrentan en combate dentro de escenarios específicos. En esos juegos realizar diferentes combinaciones de movimientos y botones hace que dichos personajes hagan diferentes ataques.
<i>Platform Fighter</i>	Nombre internacional dado al subgénero de los juegos de lucha, el cual lo mezcla con el juego de plataformas por lo que el objetivo principal es conseguir que el rival no sea capaz de volver a la plataforma principal a base de atacarle.
<i>Blast Zone</i>	Zona existente en los <i>Platform Fighters</i> en la cual si un personaje entra será eliminado, perdiendo una vida.
<i>Pixel Art</i>	Tipo de arte caracterizado por ser construido a través del uso de píxeles individualmente, haciendo que en el dibujo se puedan diferenciar los cuadrados que conforman dichos píxeles
<i>Parry</i>	También llamado “parada” o “desvío”. Acción defensiva que debe realizarse a la vez que se recibe un ataque, aturdiendo al atacante y permitiendo al defensor tomar la ofensiva.
<i>Grappler</i>	Arquetipo de personaje dentro de los juegos de lucha, los personajes dentro de este arquetipo se especializan en el uso de agarres, los cuales son ataques únicos capaces de golpear a rivales que se estén defendiendo.
<i>Frame Data</i>	Conjunto de información que explica la velocidad de los movimientos de un personaje a través de datos numéricos basados en los fotogramas de dicho movimiento. Está usualmente dividida entre



“*startup*”, “*active*” y “*recovery*”, aunque esto puede variar según el juego.

- Counter* Movimiento a la vez defensivo y ofensivo en el cual el personaje que lo realiza se prepara para contraatacar ante un ataque, ya sea específico o general. Si dicho personaje recibe dicho ataque durante este movimiento, realizará un movimiento único automático.
- Gameplan* Concepto abstracto que indica la manera en que se trata de ganar con un personaje de un juego de lucha en concreto, depende de cada personaje y, en ciertas ocasiones, también del personaje al que se enfrente.
- Mixup* En un juego de lucha, se conoce como *mixup* la situación en la que uno de los jugadores tiene la opción de realizar diferentes acciones que requieren de diferentes respuestas por parte del rival. Si el rival no es capaz de reaccionar a las opciones del jugador deberá adivinar cual usará, haciendo que defenderse de estas situaciones sea especialmente difícil.
- Killmove* En los *Platform Fighters* se conoce como *killmove* a aquellos movimientos que son capaces de acabar con el rival a bajos porcentajes bajos a la vez que son fácilmente accesibles y relativamente rápidos para su potencia, usualmente a cambio tienen una recuperación lenta o corto alcance.
- Golpe limpio* También conocido como *Clean Hit*. Para que un golpe limpio ocurra, el movimiento debe tener más de una *hitbox* y una de ellas debe tener un comportamiento diferente al resto, usualmente siendo más fuerte que las demás. Dicha *hitbox* suele ser más pequeña y difícil de dar, usualmente se refiere a esta *hitbox* como *sweetspot*. Cuando se golpea un movimiento con dicha *hitbox* se conoce como Golpe limpio o golpear limpiamente
- Spike* En un juego de lucha en plataformas, un ataque de empuje fuerte que lanza directamente en vertical hacia abajo, normalmente asegurando quitarle una vida al oponente si este no está directamente encima de una plataforma en el momento de ser golpeado.



Introducción

1.1 Descripción del problema y motivaciones

El problema que se trata de abordar en este trabajo es cómo empezar a construir un portfolio como diseñador de videojuegos y tener material sobre creación y balance de mecánicas y personajes dentro del mismo sin necesitar crear juegos enteros para tener versiones probables y demostrables de dichas ideas de diseño. Así mismo, este trabajo también permite obtener retroalimentación del trabajo que se ha hecho y así poder realizar varias actualizaciones para tener un producto de prueba lo más refinado posible. Esto también permite utilizar una comunidad ya existente en vez de necesitar crear una comunidad desde cero para que pruebe dicha idea.

La principal motivación para la realización de este trabajo fue la necesidad de elaborar un portfolio especializado en la creación de personajes y mecánicas en el desarrollo de videojuegos [3] sin necesitar crear videojuegos enteros para tener pruebas visuales (imágenes y videos) de dichos conceptos en funcionamiento dentro del ecosistema de un videojuego, permitiendo así también demostrar la capacidad de adaptación a mundos y sistemas ya creados con anterioridad. Así mismo, esta opción permite obtener opiniones de diferentes jugadores para demostrar si el concepto de dicho personaje ha sido bien recibido o no.

Se decidió escoger *Rivals of Aether* (Dan Fornace, 2017) al ser un juego con una gran comunidad dedicada a la modificación de este, teniendo extensos tutoriales creados tanto por los desarrolladores como por la comunidad. Además de esto, el juego lleva teniendo modificaciones desde hace varios años, por lo que dentro de estas comunidades sería fácil encontrar plantillas para diferentes funciones dentro del código y respuestas a posibles problemas ya que es probable que dichos problemas le surgieran a alguien más con anterioridad. Sumado a esto, el punto de entrada para la modificación de dicho juego es relativamente bajo, solo se necesita del juego, una herramienta para crear pixel art y una herramienta de código proporcionada por los desarrolladores en su página oficial. También es importante destacar que, durante la realización de este trabajo, el equipo de *Rivals of Aether* anunció un concurso relacionado con la creación de un personaje para la workshop de Steam, lo cual se vio como una buena oportunidad para obtener retroalimentación de múltiples usuarios.



1.2 Objetivos

1.2.1 Objetivo general

El objetivo general de este trabajo es:

OG: Desarrollar un personaje de videojuego de lucha al completo, diseñando al mismo desde el inicio, llevando a cabo los pasos necesarios para integrarlo en un videojuego comercial.

1.2.2 Objetivos específicos

OE1: Realizar un análisis exhaustivo del género

Analizar el género de los juegos de lucha, con especial énfasis en el subgénero de los *Platform Fighters* y *Rivals of Aether*, identificando las características, mecánicas y requisitos técnicos que hacen que estos juegos funcionen.

OE2: Diseñar un personaje con la complejidad y diversidad de movimientos adecuada.

Diseñar un personaje que cumpla con los estándares funcionales del juego, manteniendo la coherencia estética y el estilo de este dentro del juego en el que se va a implementar.

OE3: Crear y desarrollar las animaciones y sprites necesarias para completar los aspectos visuales del personaje.

Para este objetivo, se decidió descargar Aseprite con la intención de crear los sprites y las spritesheet necesarios para el funcionamiento correcto del personaje. Para esto se debe asegurar la fluidez y coherencia visual del personaje en todas sus animaciones.

OE4: Integrar y programar el personaje

Utilizando la versión modificada de GMEedit proporcionada por los desarrolladores del juego y la plantilla base de personajes creada por los mismos se implementará el personaje en el entorno del juego, integrando todas las animaciones y movimientos, asegurando el funcionamiento correcto del mismo dentro del juego.

OE5: Validar el personaje

Realizar pruebas y recoger retroalimentación de los jugadores para evaluar el rendimiento y balance del personaje con la intención de realizar los cambios necesarios para mejorar el funcionamiento de este dentro del juego.

OE6: Desplegar el personaje a la plataforma de descarga de mods de Steam

Hacer posible la descarga del personaje de manera automática y sencilla para que cualquier persona pueda buscarlo y utilizarlo dentro de su copia del juego.

1.3 Planificación y metodología inicial

La planificación de tareas ha sido dividida en 6 fases diferentes, cada una de ellas relacionada con cada una de las partes necesarias para que el proyecto pudiera ser realizado correctamente, comenzando por el análisis del género y la investigación sobre el balance dentro del mismo, además de un estudio más directo sobre el balance de **Rivals of Aether**. La segunda fase estaría enfocada en el diseño visual y jugable del personaje, durante esta fase se diseñarían todos los aspectos del personaje, desde sus ataques, funcionalidades y estadísticas hasta su diseño visual y trasfondo. Durante la tercera fase se animaron e implementaron cada uno de los movimientos, empezando por aquellos movimientos que no requerían de crear código nuevo para funcionar correctamente y continuando con el de los ataques que requirieran de crear código nuevo para su correcto funcionamiento. En cuarta fase del proyecto se presentó el personaje y se creó el material necesario para que fuera interesante para los jugadores, creando las imágenes necesarias para decorar la página de descarga del mod, con la intención de atraer jugadores para la quinta fase, durante la cual se buscaría recoger el *feedback* de los jugadores balancear correctamente el personaje y arreglar los posibles errores que presentase. La sexta y última fase estaría relacionada con la redacción y edición de la memoria del proyecto.

1.4 Metodología

Durante la creación del personaje se mantuvo una metodología en cascada dividido en una fase inicial de análisis del género y del videojuego en específico. Tras esto comenzó la fase de desarrollo del personaje en sí mismo, la cual comenzó con el diseño de los ataques del personaje y de sus estadísticas de combate. Tras terminar la fase de diseño, se comenzó la fase de desarrollo de dichos ataques, dividida entre el desarrollo visual, en el que se creaban las animaciones para cada uno de los movimientos y ataques, y el desarrollo técnico, en el cual se implementaban dichas animaciones a través de código y se les añadía la funcionalidad necesaria para que actuaran correctamente durante el juego. Por último, comenzó una larga fase de validación, en la cual con ayuda de varios jugadores se realizaron las pruebas necesarias para balancear al personaje dentro del juego.

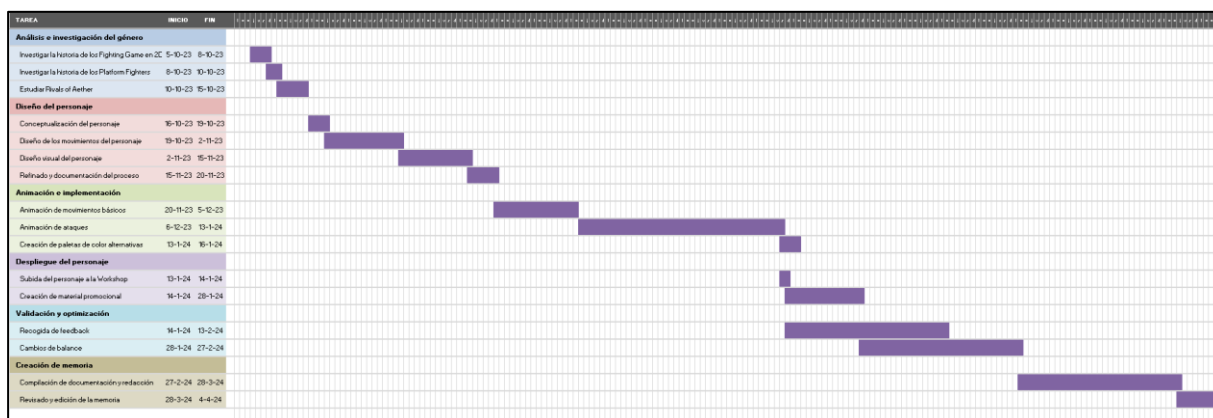


Ilustración 1. Diagrama de Gantt de la planificación



TAREA	INICIO	FIN
Análisis e investigación del género		
Investigar la historia de los Fighting Game en 2D	5-10-23	8-10-23
Investigar la historia de los Platform Fighters	8-10-23	10-10-23
Estudiar Rivals of Aether	10-10-23	15-10-23
Diseño del personaje		
Conceptualización del personaje	16-10-23	19-10-23
Diseño de los movimientos del personaje	19-10-23	2-11-23
Diseño visual del personaje	2-11-23	15-11-23
Refinado y documentación del proceso	15-11-23	20-11-23
Animación e implementación		
Animación de movimientos básicos	20-11-23	5-12-23
Animación de ataques	6-12-23	13-1-24
Creación de paletas de color alternativas	13-1-24	16-1-24
Despliegue del personaje		
Subida del personaje a la Workshop	13-1-24	14-1-24
Creación de material promocional	14-1-24	28-1-24
Validación y optimización		
Recogida de feedback	14-1-24	13-2-24
Cambios de balance	28-1-24	27-2-24
Creación de memoria		
Compilación de documentación y redacción	27-2-24	28-3-24
Revisado y edición de la memoria	28-3-24	4-4-24

Ilustración 2. Fases, tareas y duraciones planificadas



Marco teórico

2.1 Análisis del juego y de su género

Rivals of Aether es un juego de lucha en plataformas o *Platform fighter*, el cual es un subgénero de los juegos de lucha o *fighting games*. Estos juegos son principalmente juegos multijugador y competitivos, donde dos jugadores se enfrentan entre sí, aunque algunos tienen elementos adicionales como un modo historia, la posibilidad de jugar contra una inteligencia artificial o entrenar con un personaje en un terreno especial de pruebas.

Para definir correctamente el juego en cuestión, se ha decidido hacer un análisis del género en base al cual fue creado, yendo desde lo más general, es decir, describir qué es un juego de lucha y en qué se basa, hasta lo más específico, es decir, qué hace a *Rivals of Aether* único respecto a sus competidores.

2.1.1 Análisis de los juegos de lucha 2D

Los juegos de lucha 2D se basan en enfrentarse a dos o más jugadores entre sí en un escenario bidimensional, es decir, las únicas posiciones que importan en el mapeado son la posición horizontal y vertical de los personajes.

La condición de victoria de los juegos de lucha vencer al rival en el combate, para lo cual cada juego decide un sistema distinto, siendo el más común vaciar la barra de vida del enemigo. Para esto, cada jugador utiliza diferentes clases de movimientos. Usualmente, en los juegos de lucha, los jugadores pueden escoger entre multitud de personajes que tratan de diferenciarse del resto teniendo movimientos y habilidades únicos. Estos principios fueron normalizados por *Street Fighter II: The World Warrior* (Capcom, 1991), ya que fue el primer juego de lucha que atrajo un éxito global, convirtiéndose en uno de los juegos de lucha más con más influencia de la historia, popularizando el género por primera vez.

Los enfrentamientos entre los jugadores suelen darse en sets de rondas, cuya duración depende del juego. Durante cada una de las rondas, ambos jugadores se enfrentan hasta alcanzar el objetivo concreto del combate, ya sea vaciar la barra de vida del rival con los ataques, sacarlo del escenario o dar un número de golpes específico al rival. Durante estos enfrentamientos, ambos jugadores solo podrán usar los personajes elegidos antes de comenzar el enfrentamiento, debiendo utilizar los movimientos únicos de cada personaje para ganar las interacciones que ocurran entre ambos. Los juegos de lucha tienen como elemento principal la jugabilidad emergente[8], por lo que el interés en estos juegos aparece gracias al número de interacciones posibles entre ambos jugadores y la diferencia entre los personajes y la manera de jugar de cada jugador. La gran variedad de posibilidades que ocurre en un juego de lucha da lugar a muchas situaciones diferentes, que han sido apodadas por la comunidad de jugadores a lo largo de los años, dando así lugar a conceptos como el neutral,

es decir, la fase en la que ambos jugadores están distanciados entre sí y ninguno impone ninguna ventaja sobre el otro; la presión, que da nombre a la situación en la que un jugador tiene ventaja sobre el otro y la usa para tratar de mantener interacciones favorables con la intención de ganar los combates, la “*frame data*” que se conoce como el conjunto de datos técnicos sobre los ataques de cada personaje, permitiendo que los jugadores aprendan que ataques son más favorables en cada situación y cómo reaccionar correctamente a los mismos.



Ilustración 3. Captura de pantalla de Street Fighter II: The World Warrior

Al igual que los elementos generales del juego han adquirido importancia y nombres gracias al paso del tiempo, los personajes y sus movimientos también han adquirido nombres específicos, teniendo así diferentes arquetipos que caracterizan a cada personaje para saber cuál es la manera general en la que tratan de adquirir la victoria cada uno de los mismos, en varios juegos los personajes tienen su arquetipo escrito al lado de su nombre para guiar al jugador hacia los personajes que puedan interesarle más, como en **Guilty Gear: Strive** (Arc System Works, 2021). Además de esto, se han dividido los movimientos de los personajes entre “normales”, los cuales suelen ser similares entre los personajes y cumplen una serie de normas generales y los “especiales”, los cuales no tienen por qué tener ninguna relación entre los personajes y tienen propiedades únicas para cada personaje.

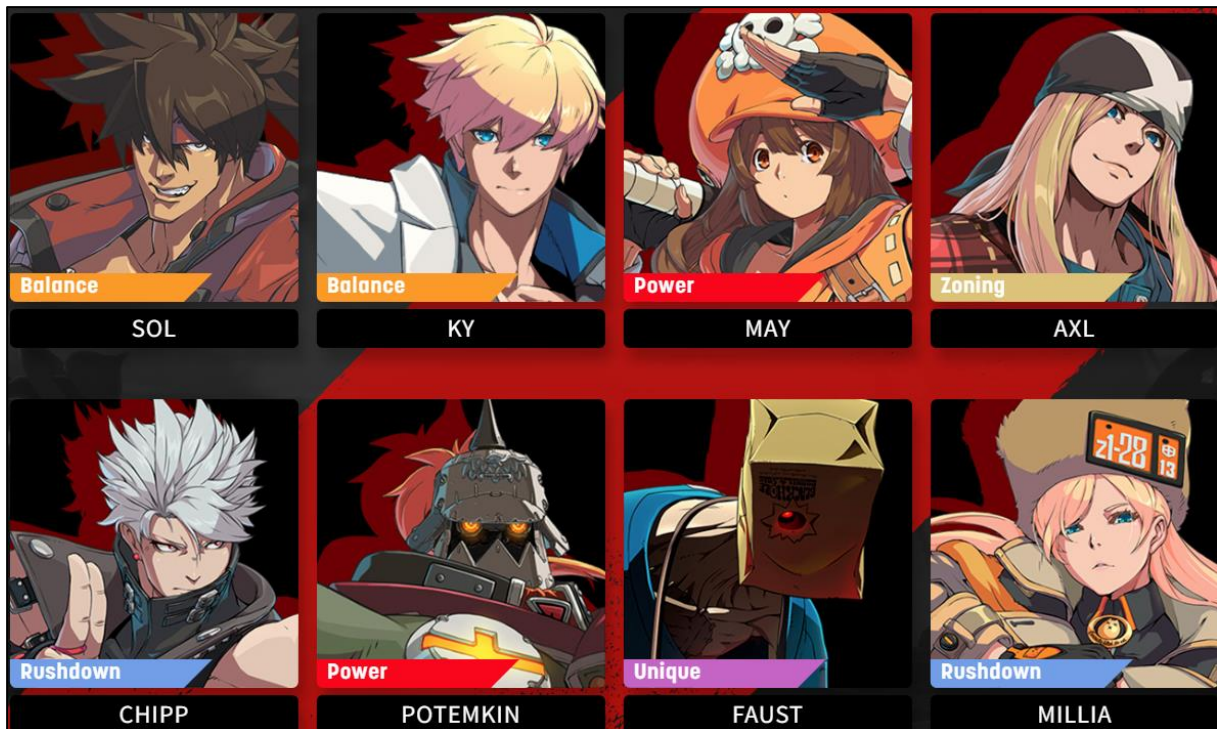


Ilustración 4. Personajes de Guilty Gear, con la clasificación de su arquetipo dentro del marco de personaje.

2.1.2 Análisis de los juegos de lucha en plataformas

Los juegos de lucha en plataformas, o *platform fighters*, son un subgénero de los juegos de lucha, se caracterizan porque los personajes tienen la movilidad propia de un juego de plataformas, teniendo así salto, doble salto, pudiendo controlar su movimiento en el aire, así como poder girarse en cualquier momento, además de esto, los mapas 2D toman más importancia, añadiendo posibles plataformas y cambiando la forma del escenario principal, normalmente siendo esta una plataforma flotante con abismos a los laterales. En estos juegos, la condición de victoria convencional es expulsar al oponente del mapa visible por los jugadores a unas zonas llamadas *Blast zones*, si uno de los personajes sobrepasa dicha zona, perderá una vida y reaparecerá sin daños en el centro del escenario.

En esta clase de juegos, los ataques tienen dos clases de daño principales:

- El daño que realizan, que normalmente se expresa en forma de porcentaje, así un ataque que haga un 10% de daño será más dañino que uno que realice un 5% de daño, este daño se acumula en los oponentes a los que impacte, sumándose entre sí y restaurándose a 0 cuando el personaje pierde una vida.
- El empuje o *knockback*, el cual suele estar dividido entre el empuje base y el escalado de empuje. El empuje es la manera principal de ganar el enfrentamiento, ya que es lo que impulsa a los oponentes fuera del mapa. Para alargar los enfrentamientos y aumentar la variabilidad del combate, este empuje se divide en empuje base, que determina el empuje que dicho ataque realizará independientemente del daño



acumulado por el enemigo, y el escalado de empuje, el cual se usa en una operación matemática junto al daño acumulado del personaje que ha recibido el ataque para determinar el aumento de empuje del ataque. Tras sumar ambos tipos de empuje, se obtiene el empuje que realiza un ataque en concreto en un enemigo estándar. Además de esto, los personajes suelen tener estadísticas propias como el “peso” o la fricción con el aire, las cuales modifican el empuje final que recibe el personaje en específico.

Además de estas particularidades en el sistema, generalmente, los personajes de los *platform fighters* también siguen una serie de normas específicas dentro de sus capacidades:

- Los personajes deben tener un movimiento de “recuperación”. Estos movimientos deben incluir un cambio en su velocidad horizontal, vertical o ambas para permitir que el personaje vuelva a la plataforma principal si ha sido expulsado de la misma, pero no ha llegado a sobrepasar las blast zones. En su defecto, dichos movimientos también pueden estar constituidos por un teletransporte o gancho para desplazarse de vuelta a la plataforma principal.
- El ataque que realiza el personaje se basa en su situación actual, diferenciando si el personaje está en el aire o no, la entrada de movimiento y el tipo de ataque realizado. Siguiendo esta norma, los personajes tienen la capacidad de tener un ataque por cada estado de movimiento (neutral, adelante, abajo y arriba), multiplicado por los tipos de ataques que haya en el juego (normalmente divididos entre normales, fuertes y especiales) y su estado actual entre tierra y aire, dando lugar a todos los movimientos del personaje.
- Los personajes tienen acceso a movimientos separados de los ataques que sirven para defenderse, normalmente manteniendo al menos la capacidad de esquivar ataques momentáneamente, aunque más métodos han sido introducidos por diferentes juegos.

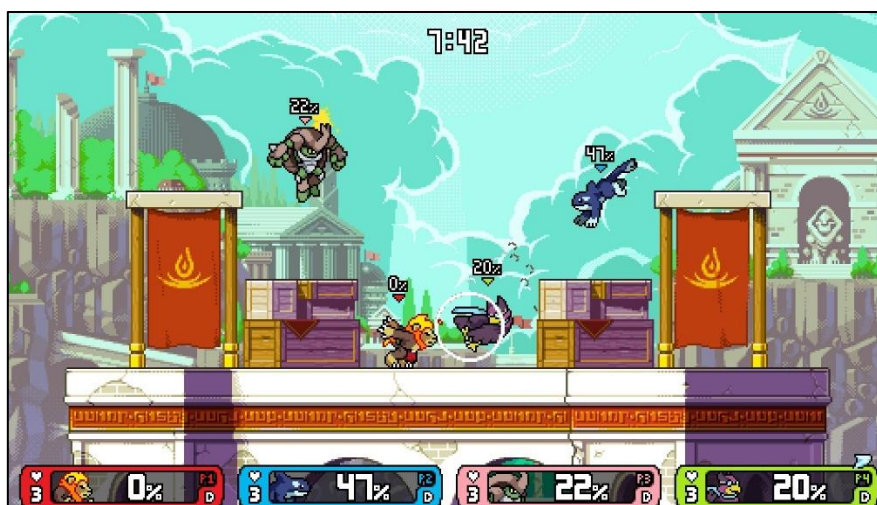


Ilustración 5. Imagen de Rivals of Aether

2.1.3 Análisis de Rivals of Aether

Gracias a las definiciones anteriores se puede definir *Rivals of Aether* como un *platform fighter* con un estilo artístico en píxel art. *Rivals of Aether* tiene varios elementos como su valor único con respecto a sus competidores, estos son:

- Los personajes son animales o criaturas antropomórficas, no hay humanos entre los personajes jugables.



Ilustración 6. Maypul y Etalus de Rivals of Aether

- Todos los personajes base del juego están anclados a uno de entre cuatro elementos, siendo estos aire, tierra, fuego y agua. El elemento del que forma parte cada uno de los personajes determina el diseño visual del personaje, siendo dichos elementos la base del diseño visual de las habilidades de los personajes y de los colores que usan en su diseño principal.
- Los personajes tienen como capacidades defensivas la esquivada, que se puede realizar tanto en el aire como en tierra, y la parada o *parry*, la cual solo se puede realizar en tierra. Si el personaje realiza una parada y recibe un ataque mientras que la parada está activa, el enemigo quedará aturdido durante un tiempo.
- Los personajes tienen un elemento único que los diferencia apodado *gimmick*. Las habilidades y la jugabilidad general de los personajes están fuertemente relacionadas con su *gimmick*. Estos elementos deben ser únicos entre los personajes y pueden ser tanto pasivos como activos.
- Los personajes tienen alguna forma de control de escenario. Es decir, todos los personajes son capaces de modificar el escenario existente de alguna manera o de limitar el acceso al mismo por parte del oponente. Esto puede ser por diversas maneras, ya sea creando plataformas, disminuyendo la visibilidad de una zona o creando zonas donde las habilidades enemigas no tengan efecto o las habilidades del personaje estén potenciadas.



Ilustración 7. 4 personajes de Rivals of Aether, cada uno representando un elemento: Fuego, Tierra, Aire y Agua

2.2 Casos de estudio

Por la naturaleza de los juegos de lucha, los personajes dentro de los mismos son extremadamente diferentes entre sí, lo cual significa que el primer paso lógico es decidir la clase de personaje que se quiere crear para tomar casos de estudio similares al personaje en cuestión. En los *fighting games* existen diferentes arquetipos, definidos por su manera de jugar y sus capacidades, así como por sus fortalezas y debilidades, existe una gran cantidad de arquetipos, como *zoners* que se dedican a lanzar proyectiles para controlar el espacio de combate o *puppeteers* que usan de un segundo personaje menos importante para atacar. Con eso en mente, se decidió empezar el análisis de los personajes priorizando los personajes que compartieran arquetipo con el del personaje que se pretendía crear. Con esto en mente se decidió que el arquetipo del personaje sería "*pixie grappler*", es decir, un personaje ágil cuyos movimientos únicos se basen principalmente en técnicas de agarre.

Teniendo esto en cuenta se trató de buscar personajes dentro del mismo arquetipo en diferentes *platform fighters*, tras una intensiva búsqueda, no llegó a encontrar ningún personaje que cumpliera con el arquetipo exacto que se estaba intentando crear para el mismo. Por este motivo se decidió dividir los casos de estudio en dos partes, personajes ágiles



en *platform fighters* y *grapplers* en *platform fighters*, tomando como ejemplos a los personajes más importantes entre todos los juegos del género, tratando de dar prioridad a *Rivals of Aether* sobre el resto.

2.2.1 Personajes ágiles en *platform fighters*

Se entiende por agilidad en un *platform fighter* la habilidad de un personaje para moverse por el escenario y recuperarse tras ser expulsado de la plataforma principal. Esto se obtiene a través de altas velocidades de movimiento tanto aéreo como terrestre, bajos tiempos de recuperación tras caer de un salto, movimientos especiales que permitan movimientos en direcciones únicas o rápidas ráfagas de movimiento o teletransportes y otras estadísticas y habilidades únicas relacionadas con acelerar el movimiento del personaje.

Para hacer un estudio de los personajes rápidos del con respecto a *Rivals of Aether*, se utilizó los personajes más ágiles de la entrega, siendo estos Wrastor y Maypul.

- **Wrastor:** Wrastor es el personaje más ágil del juego en cuanto a su movimiento aéreo, lo cual se ve potenciado gracias a su especial lateral, que crea una corriente de viento horizontal que aumenta su velocidad de movimiento mientras está en la misma. Además de esto, aprovecha dicha velocidad aumentada en el aire más de lo que la aprovecharía el resto de los personajes ya que Wrastor no tiene ningún ataque fuerte mientras está en el suelo y solo puede realizarlos en el aire, destacando su preferencia por mantenerse en el aire saltando en vez de quedarse en el suelo. Además de esto, Wrastor tiene 3 saltos aéreos comparado con el resto de los personajes que solo tienen un salto aéreo. Además de esto, la mayoría de sus movimientos especiales y algunos de sus ataques normales implican alguna forma de movimiento, su neutral especial hace que haga daño a su alrededor y esquive proyectiles mientras flota en el aire como un tornado, su arriba especial es un ataque ascendente y su abajo especial le hace propulsarse hacia adelante con una patada si está en el suelo o hacia abajo si está en el aire. A cambio de esta velocidad aérea adicional, Wrastor es el personaje más ligero del juego (es decir, sufre más empuje por cualquier ataque que cualquier otro personaje del juego) y varios de sus movimientos se basan en *sweetspots* por lo que debe utilizar parte de su movilidad para posicionarse correctamente con respecto al oponente para que sus ataques tengan toda la efectividad posible.
- **Maypul:** Maypul es menos ágil que Wrastor de manera general, pero tiene la capacidad única de marcar a los oponentes con algunos de sus ataques, lo cual le permite teletransportarse hasta ellos manteniendo su arriba especial. Además de esto, su lateral y abajo especiales son un movimiento rápido en dichas direcciones, dicho movimiento daña y marca a los enemigos por los que pasa, haciendo que pueda mantenerse en movimiento mientras aplica su marca única. El teletransporte único de Maypul le permite continuar combos a enemigos tras realizar movimientos con un

empuje mayor, pudiendo terminar dichos combos un ataque rápido justo al borde de la *blast zone* para terminar con una vida del oponente. Además de esto, Maypul puede crear una planta a sus pies mientras está en el suelo, dicha planta ataca a los oponentes siempre y cuando estén marcado o aturvidos por un ataque de Maypul, aumentando su control de escenario y haciendo que no necesite moverse tanto como Wrastor. A cambio de esta movilidad aumentada, Maypul es bastante más ligera que la media de los personajes y tiene una mala recuperación, haciendo que pierda vidas mucho antes que lo habitual según su daño acumulado.



Ilustración 8. Wrastor, de Rivals of Aether

2.2.2 Grapplers en platform fighters

Para analizar a los *Grapplers* de los platform fighters, primero es necesario analizar que es un *grappler* y como distinguir a dicho arquetipo de personaje. La conversación alrededor de este concepto es complicada, ya que no existe una tabla exacta de qué habilidades convierten a un personaje en un *grappler* y este arquetipo se basa más en la misma definición de agarre dentro del género. Por esto, primero hay que explicar que es un agarre en un *platform fighter*: Un agarre en un *platform fighter* es un ataque que, tras impactar, posiciona al personaje golpeado en una posición concreta con respecto al personaje realizando el agarre antes de realizar el empuje correspondiente, esto hace que la situación generada tras el agarre sea mucho más previsible y consistente con respecto a cualquier otro ataque del juego. Además de esto, ciertos agarres permiten al jugador elegir la dirección en la que lanzarán al rival, permitiendo modificar la situación tras dicho ataque a la que sea más útil para el jugador.

Con esto en cuenta podemos definir que un *grappler* en un *platform fighter* es un personaje cuyo *gameplan* está basado principalmente en la capacidad de acertar agarres contra los rivales con la intención de generar situaciones de ventaja predecibles y utilizar dicha predictibilidad a su favor para mantener una situación de ventaja sobre sus oponentes o realizar combos de manera consistente desde dichos agarres.



Aquí se crea el primer problema, Rivals of Aether no tiene ningún grappler y tiene especialmente pocos agarres entre sus personajes, siendo el principal el agarre de Clairen el cual funciona como herramienta de combo principalmente. Aparte de Clairen, Ori tiene un agarre que le permite agarrar tanto enemigos como proyectiles y lanzarlos en la dirección que desee, pero es especialmente difícil de dar a rivales por lo estricta que es la ventana de tiempo durante la cual el movimiento se considera activo.

Por dichos motivos, para buscar *grapplers* en platform fighters se ha tenido que llevar la investigación a otros juegos, principalmente la saga Super Smash Bros, al ser la principal representante del género. Con esto he decidido estudiar a Incineroar y a Donkey Kong, utilizando la versión de Super Smash Bros. Ultimate (Sora Ltd., 2018)

- Incineroar: Incineroar, a diferencia de los personajes antes presentados, es lento con respecto a la media de su juego, aun así, tiene dos maneras de movimiento único, una de ellas es su arriba especial, el cual asciende antes de caer contra el escenario con una explosión, y la otra es su lateral especial, el cual anula su velocidad vertical y le hace lanzarse hacia adelante agarrando al rival si se encuentra en su camino. Con respecto a qué hace que Incineroar se pueda considerar un grappler en su juego se debe destacar su abajo especial y su lateral especial, así como la relación entre ambos. Su abajo especial es un *counter* con el cual puede soportar un golpe del rival y acumular una mejora de daño y empuje que se aplica a su siguiente ataque, la potencia de esta mejora depende de la cantidad de *counters* que realice antes de usarla y la potencia de los movimientos a los que contraataca, su lateral especial es un agarre en movimiento que, al impactar, agarra al rival y realiza una continuación según la ventana de tiempo en la cual el jugador vuelva a pulsar ataque, pudiendo convertirse en un golpe con mucho empuje lateral. Combinando ambos especiales, Incineroar es capaz de lanzar al oponente fuera de las *blast zones* cuando apenas tienen daño acumulado, permitiéndole acabar con sus oponentes especialmente rápido. Esto hace que su agarre principal sea un peligro constante y una de las partes principales alrededor de las cuales basa su *game plan*. Además de esto, los agarres universales de Incineroar son más potentes de lo normal y se ven afectados por la mejora de su abajo especial, haciendo que estén entre los más fuertes del juego.
- Donkey Kong: Donkey Kong, al igual que Incineroar, es un personaje lento y, a diferencia de este, apenas tiene capacidades movimiento único, siendo un personaje especialmente poco ágil con respecto al juego en el que se encuentra. El punto principal del estudio de Donkey Kong es el motivo por el cual sus agarres sobresalen muy por encima de aquellos que tienen el resto de los personajes. El motivo principal de esto es que sus agarres pueden convertirse en un movimiento único apodado "*Cargo Throw*", al activar este movimiento único, Donkey Kong es capaz de moverse

con el enemigo agarrado por el mapa, hasta que Donkey Kong use uno de sus lanzamientos para terminar el agarre o el rival se libre del agarre por las mecánicas generales del juego. Esta habilidad es especialmente fuerte al permitir a Donkey Kong saltar hasta la plataforma más alta antes de lanzar al oponente hacia arriba para que le sea mucho más fácil eliminar la vida del rival y, en caso de que el rival sobreviva, tarde más tiempo en caer, durante el cual Donkey Kong puede preparar otros de sus ataques y preparar la continuación de su presión contra el rival. Además de esto, gracias a dicha habilidad, si el mapa carece de plataformas, puede decidir saltar fuera de la plataforma principal antes de realizar un lanzamiento lateral para alejar al enemigo de la plataforma principal mucho más que si realizara el lanzamiento desde donde comenzó el agarre.



Ilustración 9. Incineroar y Donkey Kong de Super Smash Bros. Ultimate

Habiendo investigado a estos personajes se puede comprobar que la instancia de *grapplers* ágiles en *platform fighters* es escasa por no decir prácticamente nula. Incluso investigando otros juegos como *Slap City* (Ludosity, 2020) o *Multiversus* (Player First Games, 2024), los *grapplers* de estos juegos son usualmente lentos y poco ágiles en general con respecto al juego en el que se encuentran.

Lo más cercano a un *grappler* ágil podría encontrarse en Luigi en *Super Smash Bros. Ultimate*. Pese a esto, tras un estudio del personaje, este no cumple con los requisitos generales para considerarse como tal, ya que el motivo por el cual su agarre es más fuerte de lo usual es porque el personaje es capaz de realizar combos únicos que eliminan al rival cuando aún no han sido dañados y dicho agarres son una manera de preparar y empezar dichos combos, los cuales solo funcionan al inicio de cada vida del rival y tras esto quedan inutilizados. Ya que el *gameplan* del personaje, pese a incluir agarres, no permite que sea su intención



principal a lo largo de toda la partida, se ha decidido descartar a este personaje de la lista de *grapplers* que estudiar.

2.2.3 Conclusiones sobre el diseño

Gracias a los estudios realizados se puede concluir que existe un problema principal con respecto al estudio de *pixie grapplers* en los juegos de peleas en plataformas. No existe un ejemplo claro en el cual basar el diseño del personaje que se plantea y, aunque se tomaran los ejemplos más cercanos de cada parte del significado, se necesitaría hacer una gran adaptación entre juegos del mismo género debido a la falta de *grapplers* en Rivals of Aether y juntar dos conceptos que por lo general parecen adversos, ya que los *grapplers* mencionados en los estudios no solo no eran ágiles, si no que eran especialmente lentos. Como la agilidad es una característica tan importante en los juegos de lucha en plataformas, el diseño de estos está claramente adaptado a esta característica, haciendo que sea imposible traspasar movimientos y características sin realizar primero grandes cambios al diseño de estos.

Pese a dicho problema, la tarea no es imposible y se ha decidido continuar con la idea, a través de utilizar las definiciones base de los arquetipos y trabajar a través de utilizar ideas de diferentes juegos y personajes, tratando de implementarlas de manera fluida haciendo uso de muchas modificaciones para formar un kit de movimientos variado y cohesivo para el personaje. Con esto en cuenta, también será necesaria una extensiva fase de pruebas y correcciones para realizar parches de balance al personaje, ya que será complicado encontrar un balance correcto con respecto al juego por la falta de ejemplos de personajes similares, necesitando modificar ligeramente sus capacidades hasta encontrar una jugabilidad que funcione correctamente con el juego. Por esto, se consideró aceptable que se lanzara al personaje en una primera versión beta al público y, con ayuda de más personas, estudiar y comparar el personaje en comunidad para encontrar posibles fallos en el diseño y corregirlos.

Aun teniendo en cuenta las valoraciones anteriores, los estudios realizados sirven de guía para la creación del personaje, dando conocimiento al respecto de la importancia de diferentes variables y su relación con el balance del juego, este estudio ha permitido concluir que la velocidad de un personaje y la libertad de este en su movimiento es usualmente el valor que más notable en la fuerza del mismo, seguido por la habilidad de controlar el espacio del escenario ya sea con proyectiles o ataques de gran área. Además, también ha permitido conocer que la fuerza general de los ataques del personaje no tiene tanta importancia como el acceso a un movimiento con capacidades de eliminar al rival a bajos porcentajes. Todas estas variables son importantes de tener en cuenta, ya que es necesario balancear las fortalezas y debilidades de un personaje en base a la importancia de estas. Otra enseñanza importante extraída de estos análisis es la importancia de que el personaje tenga un diseño único e interesante para los jugadores ya que estos tienden a utilizar los personajes más fuertes y mejor relacionados con su estilo de juego, lo cual significa que, si se trata de mantener el personaje correctamente balanceado, es necesario que sea único para que su uso no se vea eclipsado por aquellos personajes a los cuales los jugadores ya estén acostumbrados. Por último, también se ha notado que una gran parte de los jugadores, además de los motivos anteriores, los jugadores a veces se ven atraídos por un personaje por su personalidad y trasfondo, especialmente si esta se ve reflejada en la jugabilidad de alguna



manera, ejemplos de esto es la gran cantidad de jugadores que juegan a Zangief de **Street Fighter** e Incineroar de **Super Smash Bros. Ultimate** por su relación con la lucha libre y la manera en que esta se ve representado en la jugabilidad de ambos personajes.

2.3 Animaciones de personajes interactivos para videojuegos

Para poder animar correctamente al personaje se ha realizado un estudio del funcionamiento de las animaciones en videojuegos modernos, llegando así a varios principios importantes que deben respetarse, varios de ellos enfocados al ámbito de la animación mientras que se han podido encontrar algunos principios adicionales para la animación dirigida especialmente a videojuegos.

Los primeros principios de la animación se refieren a la misma de manera general, ya sea animación digital o no y sin importar para que medio se estén creando dichas animaciones.

Los principios generales de las animaciones son: [1]

- Estirar y encoger: Este principio dice que se pueden exagerar los movimientos en la animación a través de estirar y encoger al personaje, pero en el proceso debe mantener su volumen en todo momento.
- Puesta en escena: Este principio se usa para definir al observador lo que es más importante en su campo de visión, a través de movimientos de cámara o composición de luces y objetos o personajes.
- Anticipación: Esta técnica se usa para preparar al observador de una acción que va a ocurrir y especialmente importante en los videojuegos para que el jugador pueda reaccionar correctamente a lo que está ocurriendo.
- Animación directa y pose a pose: Referida directamente al proceso de animación, diferenciando entre la animación directa, en la cual se crea cada parte de la animación de fotograma en fotograma, y la animación pose a pose, en la cual se animan solo las poses clave y después se rellenan los espacios entre estas.
- Acciones complementarias y superpuestas: Son técnicas de animación usadas para dar una sensación más natural, es importante diferenciarlas y utilizarlas para hacer que parezca que la animación sigue las normas de la inercia. Las acciones complementarias las realizan objetos desvinculados que deberán seguir moviéndose después de que el movimiento principal paré para mostrar la amortiguación del movimiento principal y las acciones superpuestas se aplica a las partes del objeto que están más sueltas y tardan en comenzar su movimiento tras el inicio del movimiento principal del objeto.



- **Acelerar y desacelerar:** Este principio describe el resultado de la aceleración y deceleración de objetos en movimiento, durante dichos movimientos, el personaje debería moverse más lento tanto al inicio como al final de cualquiera de sus movimientos.
- **Arcos:** Este principio indica que la mayoría de los movimientos naturales, como el movimiento de las piernas o los brazos, suelen ocurrir en arcos y es importante mantener curvas naturales para mantener el sentimiento de realismo en la animación.
- **Acción secundaria:** Una acción secundaria se usa para complementar y dar énfasis a las acciones principales del personaje.
- **Atractivo:** El principio se basa en hacer que los objetos y personajes sean agradables a la vista, ya sean en animaciones realistas o estilizadas, pero la manera de llegar a ello es la más abstracta.
- **Sincronización:** Está intrínsecamente relacionado con la velocidad y el tiempo que toma a los objetos o individuos para empezar a moverse, dando sentimientos de peso y fuerza.
- **Exageración:** Este principio señala que tratar de copiar un movimiento real de manera directa lo hace parecer poco interesante e imperfecto, añadiendo que exagerar ligeramente los movimientos y expresiones del personaje ayuda a acentuar lo que está ocurriendo y lo hace más interesante para el observador.
- **Dibujo sólido:** El principio del dibujo sólido enfatiza la importancia de mantener correctamente la representación de las formas tridimensionales para que den la ilusión de volumen y peso, aunque estén en un medio con dos dimensiones.

Además de estos principios generales también se han podido encontrar cinco principios más ligados a la animación en videojuegos: [2]

- **Sentir:** El elemento principal que separa al videojuego de otros medios es su capacidad para interactuar, por lo que el jugador debe tener control completo de la experiencia mientras juega, permitiéndole interrumpir animaciones cuando sea preciso, por ejemplo, cancelar la animación de correr antes de que termine ya que no quiera seguir moviéndose en dicha dirección.
- **Fluidez:** Para mantener la fluidez, la animación de videojuegos suele estar compuesta de muchas animaciones cortas que puedan usarse en una misma secuencia, ya que normalmente comienzan y paran en intervalos inconsistentes, por lo que es importante mantener la fluidez en el cambio entre las distintas animaciones del personaje y las transiciones entre las mismas.



- **Legibilidad:** En los videojuegos es importante mantener la legibilidad de las animaciones en cualquier contexto y desde todos los ángulos en los que pueda verla el jugador. Las animaciones deben estar diseñadas para funcionar correctamente en cualquier contexto que pueda ocurrir a manos del juego o del jugador.
- **Contexto:** En los videojuegos, a diferencia de en series o películas, las animaciones están creadas sin una puesta en escena en específico y suelen reutilizarse lo largo de muchas escenas distintas e incluso en varios personajes distintos, por lo que es importante hacer que estas sean usables en los todos los contextos en los que vayan a utilizarse.
- **Elegancia:** Las animaciones en videojuegos no suelen actuar por sí mismas, suelen ser parte de un sistema mayor a través del cual el jugador las activa o desactiva, haciendo que fluyan unas con otras, deben entremezclarse, combinar, superponerse y complementarse de manera suave, sin que el jugador sea capaz de percatarse de cada una de las diferentes animaciones individuales que están ocurriendo a cada momento.

Diseño de artes

3.1 Inspiración visual, referencias y moodboard.

Para decidir cuáles serían las inspiraciones correctas para el personaje, se hizo un estudio del tipo de personajes que cuadraban dentro de la descripción física del tipo de ataques y movimientos decididos durante el diseño inicial del personaje, haciendo así que sea necesario buscar personajes de diferentes medios audiovisuales que realicen agarres y sean ágiles. Con esta descripción se decidió tomar como inspiración principal a diferentes ninjas y asesinos de varios videojuegos, ya que estos suelen cumplir con el aspecto ágil que se buscaba y suelen incluir artes marciales exageradas para resultar vistosos. [4]

Con esto en cuenta, se decidió también inspeccionar el arte de *Rivals of Aether*, los personajes de este universo son principalmente animales antropomórficos que siguen patrones de color relacionados con la región de la que provienen, así los personajes tienden a colores morados en la región del aire, naranjas y marrones en la región del fuego, azules y naranjas en la región del agua y verdes y marrones en la región de la tierra. Además de estos colores, todos los personajes pueden incluir colores blancos y grisáceos para su armadura y armas si es que usan metales.

Teniendo esto en cuenta, que realizaron varias pruebas de siluetas con algunos conceptos iniciales, con la intención de decidir que concepto parecía más interesante a primera vista.

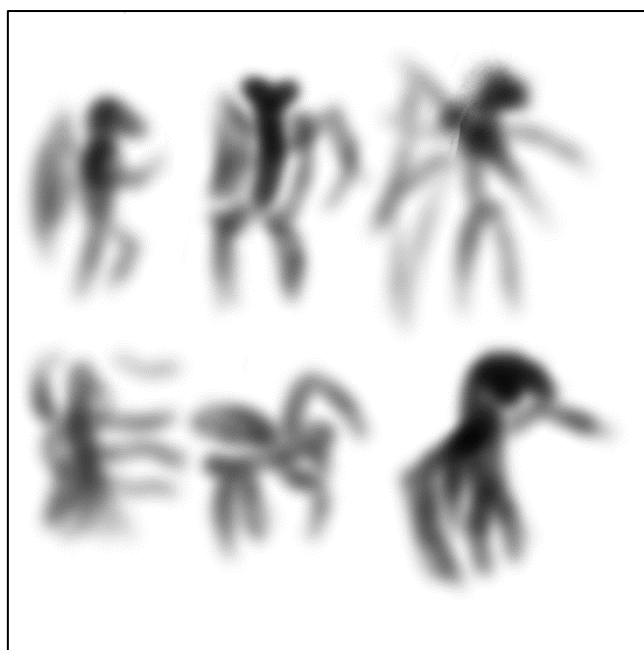


Ilustración 10. Ejemplos de sombras creadas



A través de este ejercicio inicial, se decidió que la opción más interesante era la de un ave humanoide, que tuviera tanto brazos como alas y perteneciese al reino del aire. Con este concepto en mente, se empezó a crear una moodboard para comenzar el diseño del personaje en sí mismo. Para esto, se eligió a Wrastor, el otro ave humanoide de *Rivals of Aether* que pertenecía al mismo elemento, junto a diferentes imágenes relacionadas con asesinos vistosos y especializados en artes marciales de otros videojuegos de combate como Vega de *Street Fighter* o los Ninjas del Ministerio de Interior de *Sekiro: Shadows Die Twice* (FromSoftware, 2019), junto a conceptos generales sobre aves humanoides y alas en criaturas humanoides.

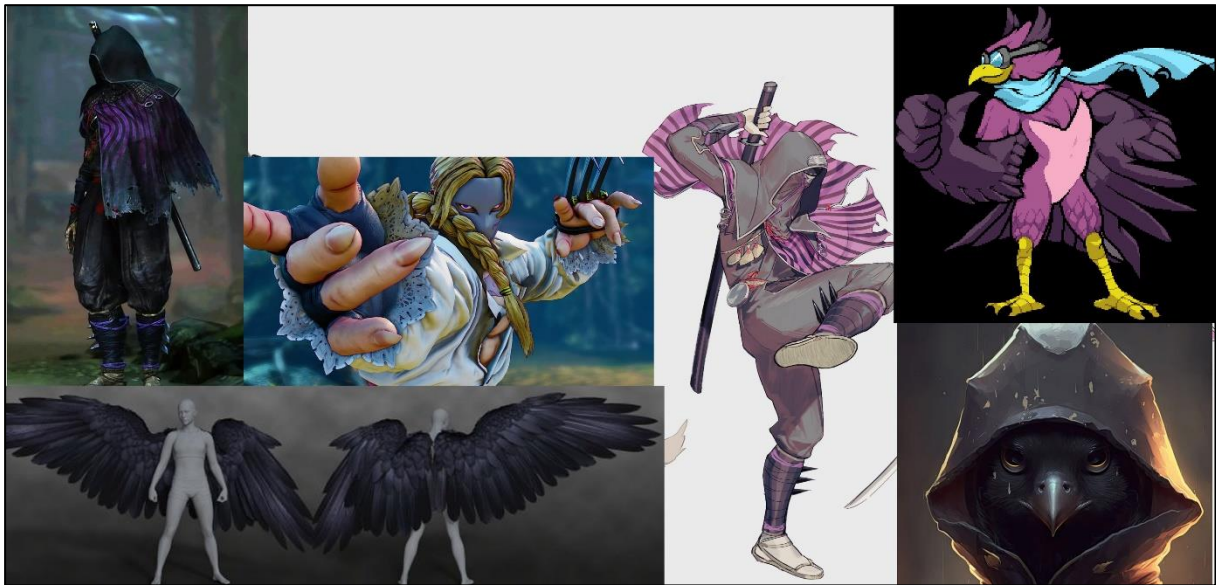


Ilustración 11. Moodboard

Con el aspecto general del personaje y las referencias que tomaría el mismo decididas, se debía elegir en qué se basarían sus movimientos y ataques, los cuales debían ser interesantes para el jugador y transmitir el hecho de que el personaje era rápido, ágil y usaba técnicas que modificaban directamente la posición del rival a través de agarres. Para esto, se decidió tomar como inspiración para su movimiento el de otros personajes rápidos o que utilizan poderes de viento para moverse, como Goenitz de la saga *The King of Fighters XV* (SNK, 2022) (este en concreto fue utilizado como inspiración para la animación de correr del personaje) mientras que los ataques y acciones ofensivas del personaje, estarían basadas en los movimientos típicos de los *grapplers* a través de la historia de los juegos de lucha, referenciando a varios movimientos típicos del medio y de diferentes artes marciales y dando prioridad a aquellos movimientos usados por personajes ágiles, como el *frankensteiner*, un agarre que se realiza con las piernas y utilizado por personajes como Clark de *King of Fighters XV* o Cammy de *Street Fighter* o el simbólico *Izuna Drop* el cual es directamente relacionado con asesinos y ninjas gracias a personajes como Ryu Hayabusa de la saga *Ninja Gaiden* (Tecmo, 1988).

3.2 Arte conceptual

Tras decidirse cual serían las influencias y las bases sobre las que se construiría el estilo visual del personaje, se decidió crear un *turn-around* del personaje a través de bocetos que después de definirían mejor, dicho *turn-around* se usaría más tarde para crear los diferentes *sprites* del personaje durante sus animaciones. Hay que tener en cuenta que los *sprites* que se usan dentro del juego son mucho menos definidos y más pequeños que el arte que se decidió crear, por lo que todo el arte conceptual del personaje deberá después ser adaptado a un primer *sprite* de mucho menor tamaño para su diseño dentro del juego.



Ilustración 12. Turn-Around del personaje

3.3 Trasfondo del personaje

Junto al personaje se decidió crear una historia que lo integrara en el mundo y justificara sus habilidades y ataques para que resulte más interesante a los jugadores [5]. Con esto en mente se comenzó un estudio de los eventos que ocurren durante el juego y qué papel podría tener el personaje fuera de estos eventos principales, como parte de justificar el hecho de que no sea parte de la historia del juego, pero sea afectado y esté interesado en el mismo para poder crear nuevas historias a lo largo del tiempo con este trasfondo.

Partiendo del estudio del mundo, se empezó decidiendo un habilidad única para el personaje que estuviese relacionada con el elemento al que pertenece, en este caso, el aire. Partiendo de esta idea, se decidió que la habilidad más interesante para un personaje que cumple el arquetipo de asesino y que es especialmente veloz sería la modificación de la presión del aire a su alrededor, permitiéndole crear armas hechas de aire a presión que no dejen rastro,



pudiendo modificar el sonido a su alrededor para ser más sigiloso y, a la vez, poder crear explosiones de aire a presión para propulsarse o mover a sus oponentes, de esta manera también se podría justificar el por qué usa agarres como método principal de combate.

Finalmente, con la habilidad desarrollada, utilizando como base el mundo que rodea al personaje y varias historias relacionadas con clanes y asesinos, se decidió crear una historia basada en la búsqueda de redención tras ejecutar a su maestro por el engaño del enemigo que amenaza al mundo de Aether, dejando atrás su nación y así, además, justificando el uso del color oscuro y de morados y púrpuras más oscuros que los del resto de personajes[6], simbolizando así que dicho personaje no pertenece a una región concreta del mundo dentro del universo de Aether.[7] Además de esto, se decidió escribir dicha historia en inglés, con la intención de que pudieran leerla la mayor cantidad de personas posible.

3.4 Flujo de trabajo: Sprites y animaciones

Una vez decidido el aspecto del personaje, su trasfondo y habilidades, se deberá comenzar con el desarrollo visual del mismo. Para esto, se decidió crear en dos pasos principales, la creación del movimiento en teoría, es decir, la velocidad, aspecto general y funcionalidad del ataque y, más adelante, en el segundo paso se crearían cada uno de los ataques por separado, siendo implementados uno por uno para poder compararlos entre sí. Se decidió este método porque la creación de ataques de un personaje debe ser cohesiva y lógica en cuanto a su velocidad y funcionalidad, por lo que se crea una gran cantidad de variables y cambios hasta encontrar un conjunto de habilidades que pueda funcionar en sintonía para el personaje y crear una jugabilidad interesante y entretenida para el jugador. Por este motivo, primero se debía teorizar y ajustar la funcionalidad de los movimientos antes de asignarles el valor visual a los mismos, ya que el aspecto visual podría modificarse a lo largo del desarrollo sin modificar su funcionalidad, pudiendo ajustarse más libremente una vez se sabe cómo se deberá comportar el personaje.

Para esto, se creó una tabla de datos que contenía cada uno de los ataques que tendría el personaje en su salida. Dicha tabla tendría una fila por cada movimiento diferente y una columna por cada uno de los datos importantes para la creación de su respectiva animación, por lo que dicha tabla tenía columnas para el inicio del ataque, el tiempo que estaría activo, el tiempo que duraría su recuperación, cómo de fuerte sería el ataque y cuánto empujaría, en qué dirección, qué clase de ataque sería y, tras rellenar todos estos apartados, también se crearía un último en el que se teorizaría una idea principal para la posible animación del movimiento.



MoveName	Startup	Active	Recovery	Total	Recovery (V)	Special/Cancel	Damage%	Pushback	Angle	Hitstun	Type	Use	Description
Jab	5	2	12	19		F	Low	Negligible	Up	Low	Attack	Poking	A stab forward with both claws
Jab 2	3	2	12	17		V	Low	Small	181°	Low	Attack	Combo filler	Open the arms and null the target in
FTilt	10	5	15	30		V	Medium	Medium	30°	Medium	Attack	LongRange Poke	A roundhouse kick
Dtilt	7	3	10	20		F	Low	Small	80°	Medium	Attack	Slagger	A quick step that explodes the ground in front of him with air
UTilt	8	3	16	27		F	Medium	Small	45° both sides	Medium	Attack	Prepare a possible USpecial	The character opens his wings in a X motion attacking enemies and launching in a 45° angle
Dash Attack	7	8	20	35		V	High	Medium	30°	High	Attack	Launcher	A dropkick
Nair	5	6	10	21		F	Low	Small	Angle flipper 1	Low	Attack	Poking and combo filler	A small air explosion surrounds the character
Fair	6	3	8	17		V	Low	Small	45°	Low	Attack	USpecial setup	A sommersault kick
Uair	8	2	10	20		F	Low	Small	90°	Low	Attack	Sharking	The char will use his claws to slash above
Uair SweetSpot	8	1	10	19		F	Low	Negligible	90°	Medium	Hijrab	Combo tool	The char will grab the opponent and put himself next to the enemy while keeping them in hitstun to combo into other air attack
Dair	8	5	10	23		F	Low	Medium	30°	Low	Attack	Failed Spike	A downwards claw attack
Dair SweetSpot	8	1	10	19		V	High	High	305°	High	Hijrab	Spike	If the claw hits the enemy clearly, the char will grab them and throw them down with a spin
Bair	8	4	10	22		V	Medium	Medium	30°	Medium	Attack	Ledguard	A fast kick backwards with good range
FStrong	17	6	20	43		F	Medium	Medium	45°	High	Attack	LongRange Attack	A forward Kick
FStrong Hijrab		2	20	22		V					Hijrab	Killmove	
UStrong	10	4	25	39		F		Small	Up-Back		Sweetspot into hijrab	Normal Protect above & combo filler	A frankensteiner. It will deal damage with the legs for the duration of the move
UStrong SweetSpot		2	20	22		F					Sweetspot Hijrab	Sweetspot Combo starter & Killmove	The tipper is at the front of the attack. If the attack hits on the sweetspot, the enemy will be grabbed

Ilustración 13. Parte de la tabla de datos usada.

Una vez con la tabla de datos terminada, se crearon las animaciones principales del personaje, estas son las animaciones que no corresponden a ningún ataque y aquellas de las que parten todas las animaciones, esto se debe a que los ataques del personaje, excepto en excepciones concretas, solo son realizables mientras el personaje está en un estado inactivo, por lo que el inicio de todas las animaciones está marcado por las animaciones del estado neutral del personaje, por lo que estas son las que se deben crear primero.



Ilustración 14. Spritesheet de la animación "Idle", reorganizada en filas para que se vea mejor.

Partiendo de estas animaciones de estado neutral, se crearon los animaciones de los diferentes movimientos del personaje. Hay que tener en cuenta que las animaciones de los ataques del personaje serían ajustadas más tarde a través del código del juego, por lo que no todos los *sprites* dentro de la *spritesheet* tienen que durar lo mismo, permitiendo evitar repeticiones y pudiendo crear saltos de la *spritesheet* para crear movimientos más complejos y secuenciados, como el caso de la animación del *Neutral Special* o del *DAir*, ya que la animación de estos movimientos cambia según si el movimiento golpea o no y según de las decisiones del jugador durante el mismo.



Ilustración 15. Spritesheet del "Down Air"

Por este motivo, tras la creación de cada una de las animaciones de los ataques y del movimiento, estas serían implementadas dentro del juego para probarlas y ajustarlas hasta que funcionaran correctamente. Al terminar el proceso con uno de los ataques comprobando que funcionase correctamente, se pasaría la siguiente ataque. En aquellos movimientos con múltiples posibles terminaciones, solo se pasaría a la siguiente terminación tras terminar la anterior.

Para crear cada uno de los *sprites* se utilizó el programa aseprite, utilizando una capa para hacer el lineado, una capa para cada uno de los colores utilizados y finalmente una capa para sombras y otra para los brillos. Cada *sprite* de un mismo movimiento era creado en el mismo

archivo cómo guía para el siguiente, ya que esto además permitía compararlos y crear la hoja de *sprites* automáticamente gracias al programa.

Por el funcionamiento de Rivals of Aether, cuando se convierte el archivo de Aseprite a la hoja de *sprites*, se debe añadir el número de fotogramas diferentes que tiene la animación al final del nombre del ataque, por ejemplo, el nombre de la animación del ataque fStrong se deberá llamar "fstrong_stripX", donde X es el número de fotogramas en la animación, 10 en el caso de este proyecto.

	1	2	3	4	5	6	7	8	9	10
☺ 🔒 ** 🗑️										
☺ 🔒 ** Aire	●	●	●	●	●	●	●	●	●	●
☺ 🔒 ** Brillos	●	●	●	●	●	●	●	●	●	●
☺ 🔒 ** Shadows	●	●	●	●	●	●	●	●	●	●
☺ 🔒 ** Outline	●	●	●	●	●	●	●	●	●	●
☺ 🔒 ** Metal	●	●	●	●	●	●	●	●	●	●
☺ 🔒 ** Dark purple	●	●	●	●	●	●	●	●	●	●
☺ 🔒 ** Purple	●	●	●	●	●	●	●	●	●	●
☺ 🔒 ** Hands	●	●	●	●	●	●	●	●	●	●
☺ 🔒 ** Skin Copy	●	●	●	●	●	●	●	●	●	●
☺ 🔒 ** Wings	●	●	●	●	●	●	●	●	●	●

Ilustración 16. Vista de capas y frames del archivo *fstrong_stripX*

3.5 Interfaz dentro del juego

Junto a la creación de los *sprites* para el juego, se debieron crear una serie de iconos para la interfaz del juego, tanto para la selección de personaje como dentro de la misma partida. El arte para la selección de personaje fue creado utilizando el arte de presentación de personaje reescalado al tamaño necesario como base, de este se creó una versión en *pixel art* con los colores correctos para el cambio de colores. El arte para la interfaz dentro de la partida fue creado siguiendo el mismo método descrito para los ataques y animaciones en el capítulo anterior, esta interfaz debe incluir un icono para cuando el personaje está fuera de pantalla, dos iconos que representen al personaje, uno en estado calmado y otro para cuando está siendo golpeado y, finalmente, cualquier icono que el personaje necesite para mostrar cualquier estado que el jugador necesite saber sobre el personaje, como cuando el personaje puede usar sus habilidades más importantes.



Ilustración 17. Aspecto de la interfaz del personaje dentro de partida.

3.6 Pruebas de colores y aspectos alternativos

Para realizar las pruebas de colores y los aspectos alternativos del personaje, **Rivals of Aether** utiliza un script llamado `colors.gml`, en dicho archivo se debe especificar cada uno de los colores usados y un rango para el matiz, contraste y brillo de dicho color. Tras esto, se crea un conjunto nuevo al que se le especifica unos nuevos colores nuevos y el juego interpreta dichos cambios creando los colores alternativos del personaje. Además de esto, los scripts de dibujado `pre_draw.gml` y `post_draw.gml` permiten realizar cambios visuales al personaje para uno de los colores en específico, permitiendo crear efectos especiales para algunos de estos conjuntos de colores.

Para facilitar el proceso de creación de colores del personaje, se utilizó el programa online de la comunidad alojado en <https://cl-9a.github.io/RoAColorsGmlHelper/>, el cual permite automatizar el progreso y ver los cambios realizados al personaje en tiempo real.

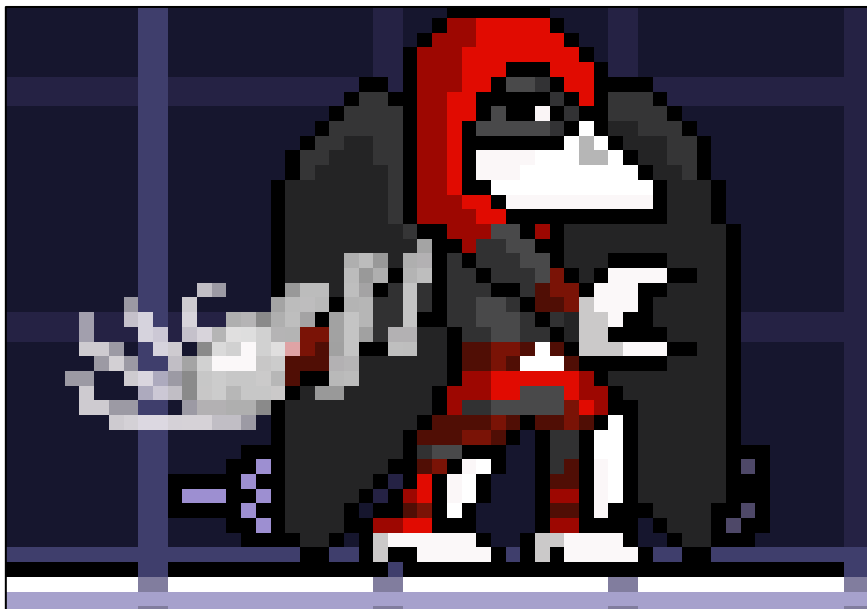


Ilustración 18. Ejemplo de un conjunto de colores alternativo.



Descripción informática

4.1 Análisis de requisitos.

Tomando como ejemplo el resto de los personajes de *Rivals of Aether* se creó una tabla de requisitos funcionales y requisitos no funcionales.

RF01: El mod es descargable a través de la interfaz de Steam Workshop

RF02: El personaje tiene, al menos, el número mínimo de movimientos con respecto al resto de personajes.

RF03: El personaje es jugable tanto en modos de un jugador como modos multijugador.

RF04: El personaje tiene una *hurtbox* precisa en la que puede ser atacado.

RF05: El personaje tiene una animación para cada una de sus acciones.

RF06: El personaje tiene diferentes paletas de colores desde las que el jugador puede elegir.

RF07: El personaje tiene sonidos en sus movimientos y ataques que acompañen dichos movimientos y ataques.

RF08: El personaje debe tener una interfaz propia que muestre al jugador la información que necesita.

Los siguientes son los requisitos no funcionales necesarios para el funcionamiento correcto del personaje.

RNF01:El personaje debe ser divertido de jugar con él y contra él.

RNF02:El personaje debe estar todo lo balanceado posible con respecto a los personajes base del juego.

RNF03:Los movimientos del personaje deben ser distinguibles a simple vista.

RNF04:La historia del personaje debe ser interesante para aquellos que la lean.

RNF05:El personaje debe sentirse único dentro del juego.



4.2 Diseño

Una vez se tuvieron en cuenta todos los requisitos del personaje se comenzó con el diseño más exacto del mismo. Para esto, se creó una tabla de datos que contenía cada uno de los ataques que el personaje tendría en su versión final, con una descripción de cada uno, tanto visual como funcional. Además de esto, dicha tabla tendría más información como la fuerza de empuje de los ataques y dirección de estos, así como el número de golpes y duración de las fases de inicio, activa y recuperación.

Para diseñar correctamente los tiempos y fuerzas de cada uno de los ataques del personaje, se decidió comparar cada uno de los ataques y sus funcionalidades con las de los personajes originales del juego, para esto se usaron dos tablas públicas que contienen todos los datos de ataques y estadísticas de los personajes originales del juego.

Se comenzó diseñando lo que sería la mecánica principal del personaje y aquella en la que se centraría su jugabilidad, se decidió adaptar una mecánica típica de los juegos de pelea tradicionales a este, haciendo que su lateral especial fuera una carrera especial cancelable y permitiendo a varios de sus ataques normales cancelarse a dicha carrera. Con esta idea principal creada, se creó un kit de movimientos centrado principalmente en realizar combos y mantener la presión sobre el rival, dándole varios ataques rápidos, pero de muy bajo empuje y solo unos pocos movimientos con alto empuje para poder acabar con rivales que hayan acumulado suficiente daño de los combos. Para contrarrestar la fuerza de esta estrategia, se decidió solo dar un ataque a distancia al personaje, un área que no realizaría ningún daño y solo acercaría al rival sin dar ninguna ventaja a ninguno de los dos tras la interacción, además, este ataque en área tendría un largo enfriamiento para que no pueda ser usada continuamente. Además de esto, se decidió que una de las cualidades únicas del personaje sería que este tendría la capacidad general de poder elegir hacia donde lanzar a su rival según el ataque escogido, por lo que los ataques empujarían en diferentes direcciones para dar esa posibilidad al jugador.

Con esto se decidió crear una cadena de jabs que mantuvieran al rival cerca a muy corta distancia, junto a otros normales en tierra que mantuvieran al personaje cerca para continuar con otros ataques, además de esto el ataque hacia abajo empujaría directamente hacia arriba, el ataque hacia adelante empujaría hacia adelante y el ataque hacia arriba empujaría ligeramente hacia la espalda del personaje. Tras esto, se crearon los ataques fuertes o cargados, estos servirían para terminar con los rivales una vez estos hubieran recibido el suficiente daño, cada uno empujando también en una dirección diferente: el fuerte hacia abajo empujaría hacia atrás, el fuerte hacia arriba hacia arriba y el fuerte hacia adelante, empujaría al frente del personaje. Tras esto se crearon los ataques aéreos siguiendo la misma filosofía de creación que los normales en tierra, el ataque hacia atrás serviría como ataque seguro para alejar a los rivales, empujando hacia atrás del personaje y teniendo buen rango a cambio de empujar menos y hacer menos daño, el aéreo hacia adelante empujaría hacia arriba para poder mantener al oponente en el aire para continuar atacando, el ataque hacia abajo funcionaría de *Spike*, es decir, un ataque que sirve para lanzar directamente hacia abajo al rival con la intención de eliminarlo si golpea fuera de la plataforma principal y finalmente el aéreo hacia arriba sería único, se decidió que al ya existir movimientos que empujan en



horizontal y vertical, este haría que el personaje se agarre al rival, haciéndole daño y poniéndose a su altura para poder realizar un *mixup*.

Tras haber decidido todos los ataques normales y fuertes, se debía decidir los movimientos especiales, primero se debía decidir que ataques se podrían realizar desde su mecánica única ya que el personaje estará centrado en la maniobrabilidad que aporta dicho ataque especial, por esto se decidieron cuatro terminaciones para el mismo: una parada que haría al personaje parar y recuperarse en un corto periodo de tiempo, un agarre en caída diagonal para atacar a enemigos debajo de la carrera, un agarre en ascenso para atacar a oponentes por encima del personaje y, por último, un *counter*, es decir, una técnica que hace que el personaje se proteja y, en caso de recibir un ataque, contraatacar al rival. Esto dejaría un punto ciego en horizontal al personaje durante la carrera, esto sería una vulnerabilidad diseñada con el propósito de limitar sus opciones y dar una debilidad a su mayor fortaleza. Además, para que no se pudiera usar como maniobra defensiva, huyendo del rival de manera continua, se decidió darle un enfriamiento del que solo empezaría a recuperarse tras tocar el suelo.

Tras decidir el funcionamiento normal del personaje, se debía decidir el resto de los ataques especiales. Se decidió por lo tanto que su neutral especial fuera un agarre de muy corta distancia tras el cual el jugador podría decidir hacia qué dirección lanzar a su oponente. Su arriba especial se mantuvo como su movimiento principal de recuperación, siendo un salto en diagonal que agarraría al rival si está cerca. Por último, su abajo especial sería el ataque comentado con anterioridad, un área que atrae a los oponentes, no realizando daño, pero permitiendo forzar una interacción entre los jugadores para poder acercar a los personajes más esquivos, a cambio este ataque tendría un largo enfriamiento y no permitiría realizar combos ni pondría a ninguno de los personajes en ventaja.

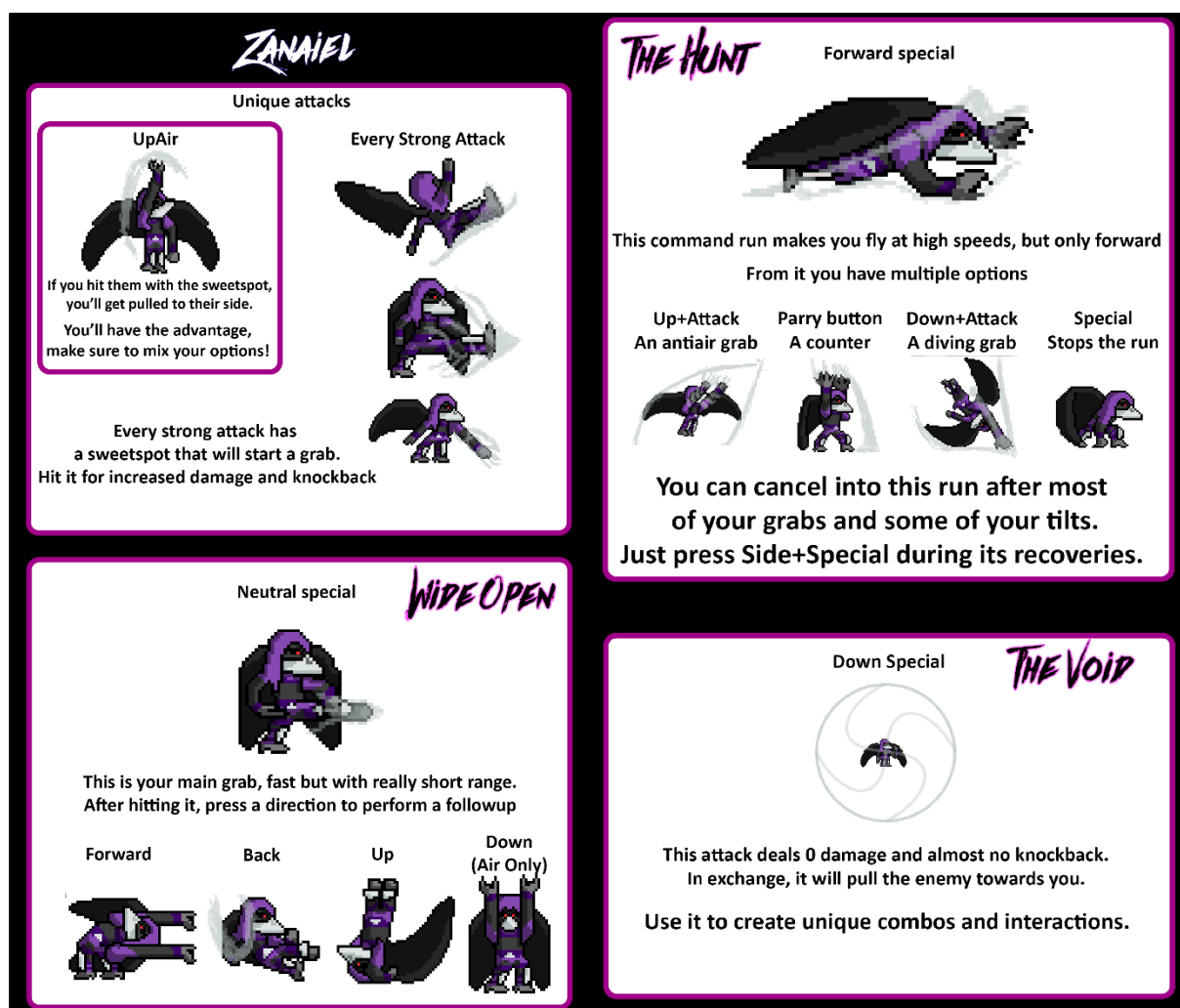


Ilustración 19. Material promocional del personaje explicando algunos de sus ataques únicos.

4.3 Implementación

Para implementar los comportamientos del personaje se ha utilizado la base de personaje proporcionada por los desarrolladores. Esta base viene por defecto con un archivo que incluye las diferentes carpetas necesitadas para crear un personaje cualquiera ordenadas, con fichas de *sprites* de ejemplo y un código para cada ataque creado como plantilla para poder crear cualquier otro personaje.

Para implementar un personaje, existe la carpeta principal, que tiene como nombre la ID del personaje en la workshop, utilizada para reconocer y actualizar el personaje cada vez que se vuelvan a subir sus datos a la workshop de Steam, dentro de esta carpeta existen imágenes para la interfaz del jugador y un archivo de configuración que tiene diferentes variables que incluyen el nombre del personaje, la región de la que procede y algunos valores con los que se subirá a la plataforma. Además de esto, dentro del archivo hay otras dos carpetas, una de ella deberá rellenarse con todas las fichas de *sprites* del personaje, incluyendo cualquier imagen usada en la interfaz que no sea general, estas imágenes deberán tener un nombre



significativo y reconocible, para que el juego interprete correctamente las fichas de *sprites* con más de un *sprite* en ellas, estas deberán terminar con *_stripX* donde X es el número de *sprites* diferentes en dicha ficha. La otra carpeta corresponde a la carpeta de scripts de código, donde están reunidos todos los archivos de código para que el personaje funcione, todos los generales, como *update.gml* que ocurre cada vez que se actualiza el juego o *init.gml* que asigna variables cuando comienza la partida, están en esta carpeta, mientras que los ataques están en otra carpeta dentro de la misma llamada *attacks*.

Para implementar los *sprites*, se deberán sustituir los *sprites* de ejemplo por los *sprites* del personaje, teniendo en cuenta de que, siguiendo el arte del juego, cada píxel en un *sprite* deberá corresponder a 4 píxeles dentro del juego, esto puede escalarse con una herramienta externa o modificando en el archivo *init.gml* la variable *small_sprites* a verdadera. Una vez hecho esto se deberán crear los *sprites* de las *hurtbox*, que simbolizan donde se puede golpear al personaje durante cada ataque, el convenio dicta que estas imágenes deberán compartir nombre con la imagen que se usa para la animación del ataque original, añadiendo *_hurt* al final del nombre del archivo, pero antes de la finalización *_stripX* si se trata de una imagen con varios *sprites* dentro de la misma. Después de esto se deberá utilizar el archivo *load.gml* para marcar el punto de inicio del *sprite*, el cual deberá estar en el centro de la base de este, para esto se usará la función de cambio de offset de la imagen, teniendo en cuenta que está basado en la forma final del *sprite* (es decir, si se han usado *sprites* pequeños, se deberá multiplicar el número por dos). Por último, en cada uno de los ataques se utilizará la función *set_attack_value(NombreDeLAtaque, AG_SPRITE, sprite_get("NombreDeLSprite"))* para hacer que dicho ataque utilice esa imagen. Hay que tener en cuenta que la *string* que simboliza el nombre del *sprite* no incluye la finalización *_stripX*.

Para implementar la funcionalidad del personaje se deberán usar el resto de los archivos disponibles. Los archivos que se encuentran fuera de la carpeta de ataques son proporcionados por los desarrolladores, en caso de no estar en el proyecto, simplemente se omitirán por el juego. En caso de añadirse, estos ocurrirán en un momento específico del mismo según el nombre del archivo, así por ejemplo lo que se programe en el archivo *update.gml* ocurrirá una vez con cada actualización del juego mientras que el archivo *got_hit.gml* ocurrirá cuando el personaje sea golpeado. Es importante destacar el archivo *init.gml*, ya que este deberá incluir todas las variables que use el personaje, incluyendo las generales (como su velocidad de movimiento o número de saltos) como las específicas del mismo que se quieran, como por ejemplo variables que guarden información sobre el último oponente golpeado.

Para implementar los ataques, se deberán usar los archivos dentro de la carpeta de ataques, cada uno de estos archivos deberá tener uno los nombres proporcionados por los desarrolladores, de esta manera, los nombres generales son llamados de manera automática cuando se realiza la combinación de botones correspondiente, mientras que hay otros nombre adicionales los cuales deberán empezar añadiendo la línea *set_attack(NombreDeLAtaque)* en la situación en la que se quieran utilizar para que se activen. Dentro de los archivos de ataque se modificará modificando las variables de este y de las *hitboxes* asociadas a este, para esto se utiliza *set_window_value* para modificar los



valores de cada parte del ataque en sí mismo, como por ejemplo si impulsa al personaje o no, cuánto dura o si genera algún efecto de sonido y `set_hitbox_value` para modificar los valores de las `hitboxes` como pueden ser el tamaño y forma, el empuje y daño o cuándo se crea.

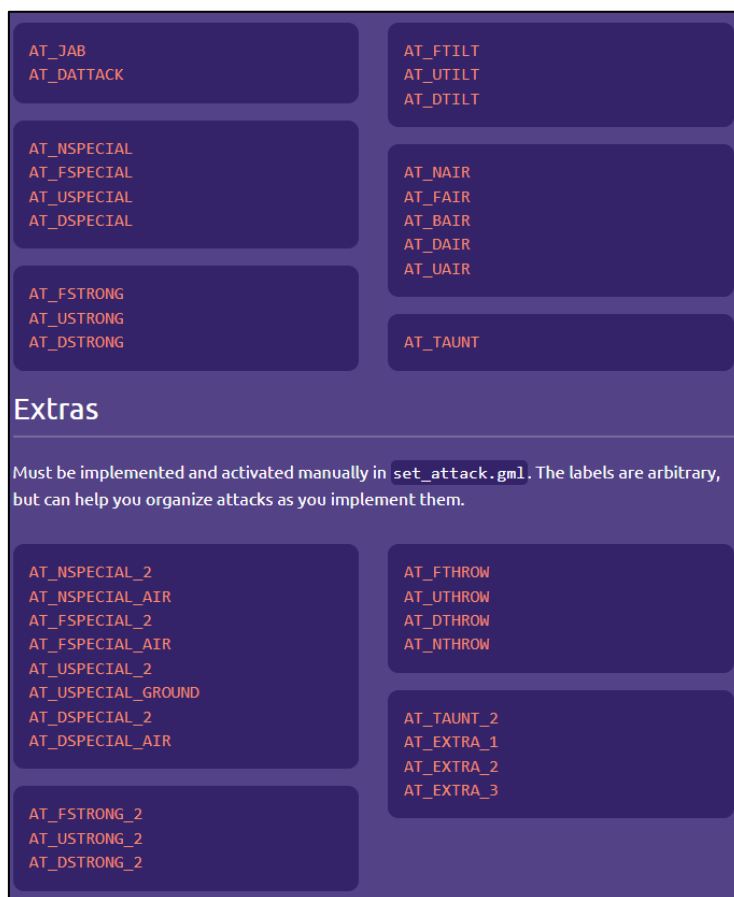


Ilustración 20. Nombres posibles para los ataques que se usen. Los de arriba están anclados a un input en concreto, mientras que los extras deben activarse por código creado por el creador del mod

Con todos estos conocimientos se procedió a comenzar la implementación informática del personaje, para lo cual primero se debía crear cada uno de los archivos necesarios para implementar todas sus funcionalidades, teniendo así los archivos

`attack_update.gml`:

En este archivo se escribirán todos los cambios a variables y funcionamientos del personaje que ocurren mientras que este esté realizando cualquier ataque, manejando así su movimiento, posiciones del enemigo durante agarres, enfriamientos de movimientos especiales, posibilidades de cancelar ataques entre sí y arreglos varios para asegurarse de que el personaje funciona correctamente. (Se debieron implementar arreglos para que la carga de los ataques fuertes pasará a su versión de golpe limpio)



`animation.gml`: Aquí se incluyeron los cambios necesarios para hacer las animaciones relacionadas con el oponente posibles, principalmente usado para modificar la rotación del *sprite* rival durante el USpecial

`colors.gml`: En dicho fichero está todo lo necesario para hacer que la modificación de colores del personaje funcione correctamente. Para esto es necesario marcar cuales son los colores usados y cuál es el rango de matiz, contraste y brillo de los estos y después generar variaciones con los nuevos colores para cada uno de los paquetes de color del personaje.

```
// Purple under air
set_color_profile_slot( 0, 0, 153, 138, 181 );
set_color_profile_slot_range( 0, 9, 45, 36 );

// Skin under air
set_color_profile_slot( 0, 1, 44, 46, 47 );
set_color_profile_slot_range( 1, 141, 8, 43 );

// Metal
set_color_profile_slot( 0, 2, 176, 186, 184 );
set_color_profile_slot_range( 2, 169, 6, 22 );

// Purple
set_color_profile_slot( 0, 3, 120, 74, 174 );
set_color_profile_slot_range( 3, 1, 3, 27 );
```

```
// Bloodbath
set_color_profile_slot( 1, 0, 74, 0, 5 ); //Purple under air
set_color_profile_slot( 1, 1, 8, 2, 0 ); //Skin under air
set_color_profile_slot( 1, 2, 105, 0, 0 ); //Metal
set_color_profile_slot( 1, 3, 140, 0, 9 ); //Purple
set_color_profile_slot( 1, 4, 105, 0, 6 ); //Dark Purple
set_color_profile_slot( 1, 5, 44, 49, 47 ); //Feathers
set_color_profile_slot( 1, 6, 121, 120, 120 ); //Hands
set_color_profile_slot( 1, 7, 209, 0, 14 ); //Eyes
```

Ilustración 21. Ejemplo de cómo funciona el código de `colors.gml`

`draw_hud.gml`: En dicho archivo se implementa el dibujado de *sprites* relacionados con la interfaz del jugador, utilizando la función `draw_sprite()`. Se creó para permitir añadir la interfaz relacionada con la disponibilidad de DSpecial ya que, al necesitar recargarse durante 5 segundos, era difícil para el jugador saber cuándo estaba disponible y cuando no. [10]



- `got_hit.gml`: El código en este fichero ocurre cuando el personaje es golpeado, se utilizó en este proyecto para hacer el código relacionado con la activación del *counter* del personaje, haciendo el cambio entre ventanas del ataque y eliminando el daño que habría recibido por el ataque que activó el *counter*.
- `hit_player.gml`: El código incluido en este archivo se activa cuando el personaje golpea a cualquier enemigo. Se utilizó para dos funcionalidades principales. La primera de ellas es modificar la variable que permite al personaje cancelar la recuperación de sus ataques a FSpecial solo cuando estos golpearan a algún rival. La segunda funcionalidad es modificar la posición y comportamiento del personaje rival tras agarrarlo.
- `init.gml`: Este fichero sirve para crear y modificar todas las variables que usará el personaje, incluyendo las estadísticas generales del mismo como número de saltos o velocidad horizontal en tierra o en aire.
- `load.gml`: Este archivo se utiliza para modificar el punto que se considera la base del personaje en cada uno de sus *sprites*. Para esto se utiliza la función `sprite_change_offset(nombre_del_sprite, offsetX, offsetY)`.
- `pre_draw.gml`: Se lleva a cabo antes del dibujado del personaje y se usa únicamente para cambios visuales en el personaje que deban ocurrir antes de que se dibuje el personaje. Se utilizó para crear un efecto visual único en uno de los colores del personaje.
- `update.gml`: Ocurre una vez por actualización del juego, se utilizó para actualizar el enfriamiento de algunos movimientos y modificar la *hurtbox* del personaje dependiendo del movimiento que esté realizando en ese momento.

Aparte de todos estos archivos, están los archivos de ataque, teniendo al menos uno por cada uno de los ataques, ya que aquellos movimientos más complejos y con ramificaciones, como por ejemplo FSpecial, necesitan de más de un archivo para mantener ordenado el proyecto del personaje. Gran parte de la programación de la funcionalidad de los ataques ocurre en el archivo relacionado con el mismo, pudiendo estos dividirse en 3 partes:

- Ajustado de los valores del ataque: En esta sección se ajustan los valores generales del ataque, como qué *sprites* usará para sus animaciones, si se considera un ataque aéreo o terrestre, el número de ventanas diferentes de las que se conforma el ataque, la *hurtbox* que tendrá el personaje mientras realiza dicho ataque y si el ataque tiene una recuperación especial si se comienza en el aire y toca el suelo mientras ocurre y, de ser así, cuánto dura. Esta sección utiliza principalmente la función `set_attack_value()`

- Ajustado de los valores de cada ventana: Las líneas de código correspondientes a esta sección sirven para modificar la funcionalidad de cada una de las ventanas en las que se divide el ataque, pudiendo declarar valores como el tipo de ventana, su duración, que partes de la spritesheet usa, si crea algún efecto de sonido o si modifica la velocidad del personaje de alguna manera, entre otras muchas posibles funciones. Se utiliza principalmente la función `set_window_value()`
- Ajustado de valores de cada *hitbox*: En esta sección primero se declara el número de *hitboxes* que tendrá el ataque en cuestión utilizando `set_num_hitboxes(nombre_del_ataque, número)`. Tras esto, se puede modificar cada una de ellas de manera individual, ajustando varios valores como el momento en el que aparecen, tamaño, forma, duración, daño, empuje, escalado de empuje o el ángulo de lanzamiento del ataque, entre otras varias posibilidades. Para esta sección se utiliza principalmente la función `set_hitbox_value()`.

<pre> 1 set_attack_value(AT_NSPECIAL, AG_CATEGORY, 2); 2 set_attack_value(AT_NSPECIAL, AG_SPRITE, sprite_get("nspecial")); 3 set_attack_value(AT_NSPECIAL, AG_NUM_WINDOWS, 3); 4 set_attack_value(AT_NSPECIAL, AG_HAS_LANDING_LAG, 4); 5 set_attack_value(AT_NSPECIAL, AG_AIR_SPRITE, sprite_get("nspecial")); 6 set_attack_value(AT_NSPECIAL, AG_HURTBOX_SPRITE, sprite_get("nspecial_hurt")); 7 8 set_window_value(AT_NSPECIAL, 1, AG_WINDOW_TYPE, 1); 9 set_window_value(AT_NSPECIAL, 1, AG_WINDOW_LENGTH, 6); 10 set_window_value(AT_NSPECIAL, 1, AG_WINDOW_ANIM_FRAMES, 2); 11 set_window_value(AT_NSPECIAL, 1, AG_WINDOW_HAS_SFX, 1); 12 set_window_value(AT_NSPECIAL, 1, AG_WINDOW_SFX, asset_get("sfx_swipe_medium1")); 13 set_window_value(AT_NSPECIAL, 1, AG_WINDOW_SFX_FRAME, 4); 14 set_window_value(AT_NSPECIAL, 1, AG_WINDOW_HSPEED_TYPE, 1); 15 set_window_value(AT_NSPECIAL, 1, AG_WINDOW_HSPEED, 0); 16 set_window_value(AT_NSPECIAL, 1, AG_WINDOW_VSPEED_TYPE, 1); 17 set_window_value(AT_NSPECIAL, 1, AG_WINDOW_VSPEED, 0); 18 19 set_window_value(AT_NSPECIAL, 2, AG_WINDOW_TYPE, 1); 20 set_window_value(AT_NSPECIAL, 2, AG_WINDOW_LENGTH, 4); 21 set_window_value(AT_NSPECIAL, 2, AG_WINDOW_ANIM_FRAMES, 1); 22 set_window_value(AT_NSPECIAL, 2, AG_WINDOW_ANIM_FRAME_START, 2); 23 24 set_window_value(AT_NSPECIAL, 3, AG_WINDOW_TYPE, 1); 25 set_window_value(AT_NSPECIAL, 3, AG_WINDOW_LENGTH, 5); 26 set_window_value(AT_NSPECIAL, 3, AG_WINDOW_ANIM_FRAMES, 2); 27 set_window_value(AT_NSPECIAL, 3, AG_WINDOW_ANIM_FRAME_START, 3); 28 set_window_value(AT_NSPECIAL, 3, AG_WINDOW_HSPEED_TYPE, 1); 29 set_window_value(AT_NSPECIAL, 3, AG_WINDOW_HSPEED, 0); 30 31 set_num_hitboxes(AT_NSPECIAL, 1); 32 33 set_hitbox_value(AT_NSPECIAL, 1, HG_HITBOX_TYPE, 1); 34 set_hitbox_value(AT_NSPECIAL, 1, HG_WINDOW, 2); 35 set_hitbox_value(AT_NSPECIAL, 1, HG_LIFETIME, 4); 36 set_hitbox_value(AT_NSPECIAL, 1, HG_HITBOX_X, 30); 37 set_hitbox_value(AT_NSPECIAL, 1, HG_HITBOX_Y, -24); 38 set_hitbox_value(AT_NSPECIAL, 1, HG_WIDTH, 54); 39 set_hitbox_value(AT_NSPECIAL, 1, HG_HEIGHT, 30); 40 set_hitbox_value(AT_NSPECIAL, 1, HG_SHAPE, 1); 41 set_hitbox_value(AT_NSPECIAL, 1, HG_PRIORITY, 10); 42 set_hitbox_value(AT_NSPECIAL, 1, HG_DAMAGE, 3); 43 set_hitbox_value(AT_NSPECIAL, 1, HG_ANGLE, 90); 44 set_hitbox_value(AT_NSPECIAL, 1, HG_BASE_KNOCKBACK, 1); 45 set_hitbox_value(AT_NSPECIAL, 1, HG_KNOCKBACK_SCALING, 0.2); 46 set_hitbox_value(AT_NSPECIAL, 1, HG_BASE_HITPAUSE, 5); 47 set_hitbox_value(AT_NSPECIAL, 1, HG_HITPAUSE_SCALING, 0.1); 48 set_hitbox_value(AT_NSPECIAL, 1, HG_EXTRA_CAMERA_SHAKE, -1); 49 set_hitbox_value(AT_NSPECIAL, 1, HG_HIT_SFX, asset_get("sfx_blow_medium1")); </pre>	<p style="color: #ff4500; font-weight: bold; margin: 0;">Ajustado de valores del ataque</p>
<pre> 8 set_window_value(AT_NSPECIAL, 1, AG_WINDOW_TYPE, 1); 9 set_window_value(AT_NSPECIAL, 1, AG_WINDOW_LENGTH, 6); 10 set_window_value(AT_NSPECIAL, 1, AG_WINDOW_ANIM_FRAMES, 2); 11 set_window_value(AT_NSPECIAL, 1, AG_WINDOW_HAS_SFX, 1); 12 set_window_value(AT_NSPECIAL, 1, AG_WINDOW_SFX, asset_get("sfx_swipe_medium1")); 13 set_window_value(AT_NSPECIAL, 1, AG_WINDOW_SFX_FRAME, 4); 14 set_window_value(AT_NSPECIAL, 1, AG_WINDOW_HSPEED_TYPE, 1); 15 set_window_value(AT_NSPECIAL, 1, AG_WINDOW_HSPEED, 0); 16 set_window_value(AT_NSPECIAL, 1, AG_WINDOW_VSPEED_TYPE, 1); 17 set_window_value(AT_NSPECIAL, 1, AG_WINDOW_VSPEED, 0); 18 19 set_window_value(AT_NSPECIAL, 2, AG_WINDOW_TYPE, 1); 20 set_window_value(AT_NSPECIAL, 2, AG_WINDOW_LENGTH, 4); 21 set_window_value(AT_NSPECIAL, 2, AG_WINDOW_ANIM_FRAMES, 1); 22 set_window_value(AT_NSPECIAL, 2, AG_WINDOW_ANIM_FRAME_START, 2); 23 24 set_window_value(AT_NSPECIAL, 3, AG_WINDOW_TYPE, 1); 25 set_window_value(AT_NSPECIAL, 3, AG_WINDOW_LENGTH, 5); 26 set_window_value(AT_NSPECIAL, 3, AG_WINDOW_ANIM_FRAMES, 2); 27 set_window_value(AT_NSPECIAL, 3, AG_WINDOW_ANIM_FRAME_START, 3); 28 set_window_value(AT_NSPECIAL, 3, AG_WINDOW_HSPEED_TYPE, 1); 29 set_window_value(AT_NSPECIAL, 3, AG_WINDOW_HSPEED, 0); 30 </pre>	<p style="color: #32cd32; font-weight: bold; margin: 0;">Ajustado de valores de cada ventana</p>
<pre> 31 set_num_hitboxes(AT_NSPECIAL, 1); 32 33 set_hitbox_value(AT_NSPECIAL, 1, HG_HITBOX_TYPE, 1); 34 set_hitbox_value(AT_NSPECIAL, 1, HG_WINDOW, 2); 35 set_hitbox_value(AT_NSPECIAL, 1, HG_LIFETIME, 4); 36 set_hitbox_value(AT_NSPECIAL, 1, HG_HITBOX_X, 30); 37 set_hitbox_value(AT_NSPECIAL, 1, HG_HITBOX_Y, -24); 38 set_hitbox_value(AT_NSPECIAL, 1, HG_WIDTH, 54); 39 set_hitbox_value(AT_NSPECIAL, 1, HG_HEIGHT, 30); 40 set_hitbox_value(AT_NSPECIAL, 1, HG_SHAPE, 1); 41 set_hitbox_value(AT_NSPECIAL, 1, HG_PRIORITY, 10); 42 set_hitbox_value(AT_NSPECIAL, 1, HG_DAMAGE, 3); 43 set_hitbox_value(AT_NSPECIAL, 1, HG_ANGLE, 90); 44 set_hitbox_value(AT_NSPECIAL, 1, HG_BASE_KNOCKBACK, 1); 45 set_hitbox_value(AT_NSPECIAL, 1, HG_KNOCKBACK_SCALING, 0.2); 46 set_hitbox_value(AT_NSPECIAL, 1, HG_BASE_HITPAUSE, 5); 47 set_hitbox_value(AT_NSPECIAL, 1, HG_HITPAUSE_SCALING, 0.1); 48 set_hitbox_value(AT_NSPECIAL, 1, HG_EXTRA_CAMERA_SHAKE, -1); 49 set_hitbox_value(AT_NSPECIAL, 1, HG_HIT_SFX, asset_get("sfx_blow_medium1")); </pre>	<p style="color: #ffff00; font-weight: bold; margin: 0;">Ajustado de valores de cada <i>hitbox</i></p>

Ilustración 22. Archivo nspecial.gml dividido en las partes explicadas anteriormente.



4.4 Despliegue del personaje a la plataforma de descargas

Una vez creado el personaje y terminado, se puede subir a la plataforma de Steam Workshop a través del mismo menú del juego, dentro del apartado de extras, el proceso de subida ocurrirá automáticamente. La página será subida en privado para que el creador pueda editarla antes de publicarla completamente, por si se quiere añadir imágenes, vídeos o una descripción junto al personaje. El personaje fue subido el día 14 de enero del 2024 utilizando este método y desde entonces se han realizado numerosos parches para arreglar errores y balancear mejor a dicho personaje.

Para actualizar el personaje, se deberá utilizar la misma url de subida en el archivo *config.ini* que la corresponde a la URL de la página del mod, una vez se esté seguro de que la URL es la misma, subir de nuevo el mod una vez actualizado simplemente lo actualizará en vez de crear una página entera nueva para el mismo, esto solo es posible si se tienen permisos para modificar la página del mod.

Capítulo 5

Validación

5.1 Resultado final

Una vez completado el personaje, este será accesible en el juego para todos aquellos que lo descarguen, una vez se activen los personajes de workshop en la configuración de la partida. Si se quiere jugar al personaje en línea, ambos jugadores deberán aceptar la configuración y tener al mismo personaje descargado desde el mismo artículo de la workshop.

El personaje final tiene un total de 21 ataques, tiene estadísticas propias, teniendo así un salto más de lo normal y teniendo una velocidad por encima de la media, además de que cumple la intención de crear un personaje especialmente ágil gracias a la alta velocidad de movimiento que permite su lateral especial.

El resultado es el que se planeó, creando así un personaje ágil, con un foco principal en sus agarres y que es capaz de cancelar movimientos normales y especiales a su especial lateral para continuar la agresión y crear combos únicos. El personaje tiene 6 paletas de colores desde las que elegir y un total de 43 hojas de *sprites* únicas, además de 23 hojas de *sprites* para las *hurtboxes*. El personaje es descargable desde la página oficial del mod y se puede encontrar utilizando el cuadro de búsqueda de contenido de la plataforma de Steam Workshop, permitiendo que sea jugado y descargado por cualquier persona que tenga el juego en Steam.

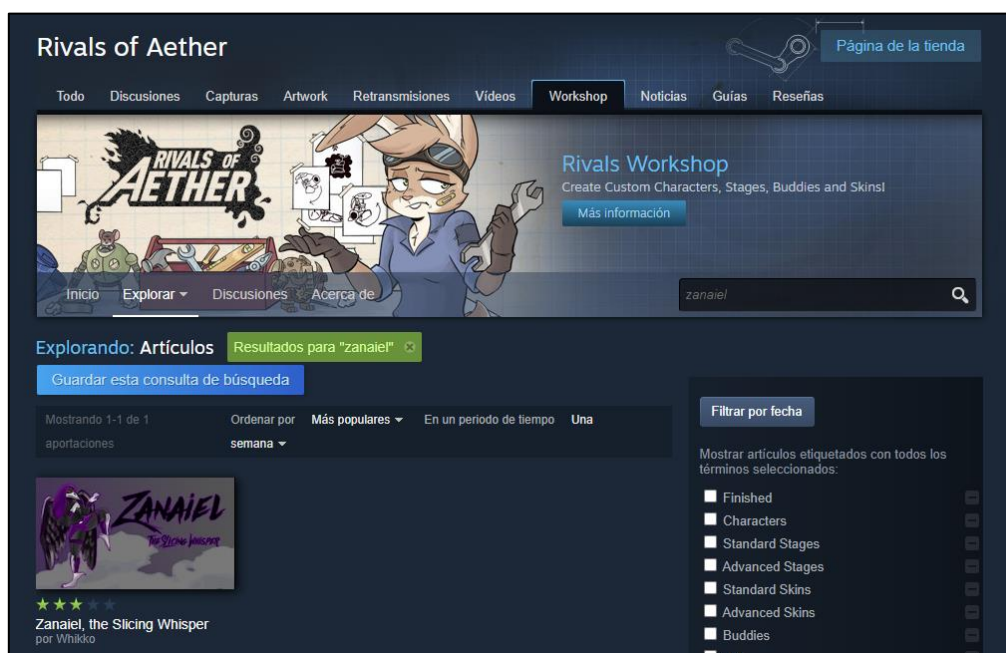


Ilustración 23. Resultados de búsqueda del personaje

Como parte de la creación de la página de descarga del personaje, se creó un vídeo compuesto por varios clips, en el cual se demuestran algunas de las habilidades del personaje y de las posibilidades que trae al juego, además de explicar muy brevemente la filosofía de diseño de este. Además de esto se incluyeron varios gifs en la página de descarga para la gente que no tuviera tiempo para ver el vídeo su funcionamiento, además de una larga descripción explicando más en detalle cómo funciona. Dicho video sobre su funcionamiento está publicado en YouTube. Se ha el enlace a dicho video en la imagen de abajo.



Ilustración 24. Miniatura del video de demostración.

5.2 Feedback

Gracias a la inclusión del personaje dentro del elenco de personajes del torneo oficial Genesis X, el cual fue elegido por los desarrolladores, el personaje llamó lo suficiente la atención de la comunidad para dar *feedback* sobre el mismo [9]. Pese a esto, la mayoría de las opiniones fueron compartidas a través de discord, dentro del servidor de creación de mods o a través de mensajes privados. Gracias a esta retroalimentación, se pudieron crear numerosas actualizaciones a lo largo del tiempo, con las cuales el personaje resultó aún más divertido y balanceado, haciendo que funcione mucho mejor dentro del juego.

En el día 23 de mayo de 2024, el personaje tiene más de un 75% de valoraciones positivas, con un total de 24 personas que han añadido el personaje a favoritos, además, ha sido descargado por más de 2400 usuarios. La recepción del personaje dentro de la comunidad ha sido principalmente positiva, aunque parte de esta opina que el personaje es más fuerte que la

Hipervínculos:

Trailer de personaje: <https://www.youtube.com/watch?v=ScD-qa3l4Lw>

Personajes en Genesis: <https://steamcommunity.com/sharedfiles/filedetails/?id=3151921503>



mayoría de los personajes del juego, por lo que es posible que no se haya logrado llegar a un balance ideal durante la creación pese a los varios parches que ha recibido a lo largo de 4 meses. Pese a esto, la mayoría del *feedback* señala al personaje como un personaje único y divertido, señalando que su principal problema ha sido su arte, ya que las animaciones de inicio no son lo suficientemente claras en algunos de sus ataques, haciendo que sea complicado reaccionar a dichos ataques de manera precisa.

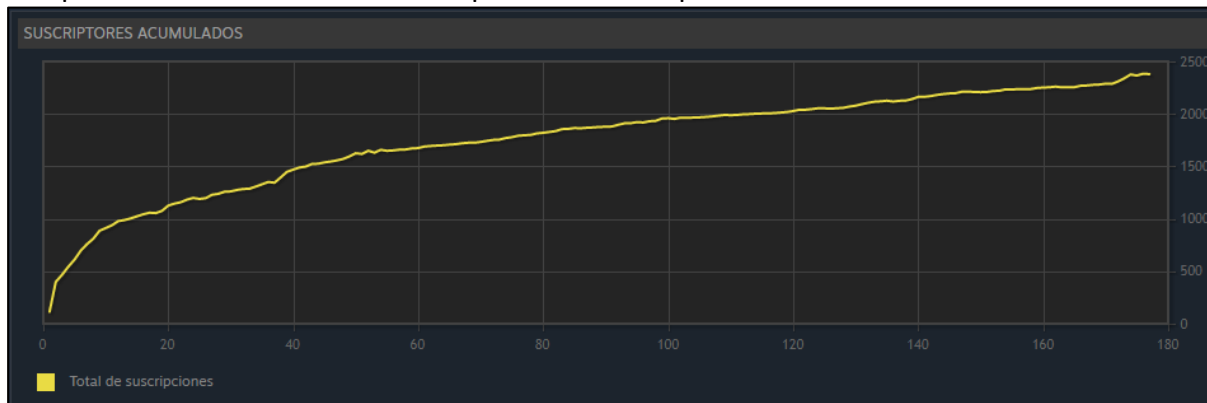


Ilustración 25. Total de suscriptores acumulados a lo largo de los días.



Ilustración 26. Estadísticas generales del artículo en el día 08/07/2024

Gracias a la gran cantidad de jugadores, se pudo obtener *feedback* directo de los mismos, junto a posibles consejos para solucionar los problemas con los que se encontró el personaje. Una parte del *feedback* negativo se basaba en los fallos de programación del personaje, que pudieron ser arreglados gracias a la información dada por los jugadores, principalmente fallos de sonidos y *hurtboxes* que funcionaban incorrectamente. Tras este primer grupo de retroalimentación, aparecieron jugadores más experimentados con información relativa al balance y fluidez del personaje. En la parte de la fluidez destacaba el tiempo que duraban los agarres del personaje, así como la duración de *hitstop* de algunos ataques. En cuanto al balance los principales problemas provenían de la velocidad y recuperación de algunos



ataques, principalmente BAir, NAir, FStrong y el ataque hacia abajo de FSpecial. Con la mayoría de estos problemas resueltos y un mayor tiempo de prueba del personaje, durante los últimos meses el *feedback* ha señalado que algunos movimientos necesitan ser más fácilmente legibles para los jugadores, ya que para jugadores no experimentados varios de los ataques se parecen mucho entre sí, resultando frustrante.

Pese a lo anterior, también se recopiló mucho *feedback* positivo para el personaje, principalmente basado en que el personaje es especialmente único, destacando mucho sobre otros personajes por como permite a los jugadores interactuar durante el combate, además de esto, varios jugadores señalaron que el personaje era muy divertido de jugar, con algunos clasificándolo como su personaje favorito del concurso. Por último, la gran mayoría de jugadores se vieron gratamente sorprendidos al notar que la habilidad que adquiría el jugador con el personaje se veía altamente recompensada por el kit del personaje, pudiendo aprender poco a poco cada una de sus habilidades y haciendo que utilizar tiempo en aprender todo de lo que es capaz, así como sus interacciones fuera especialmente gratificante y útil, añadiendo que el personaje tiene un gran potencial para cualquier jugador que quiera aprenderlo.

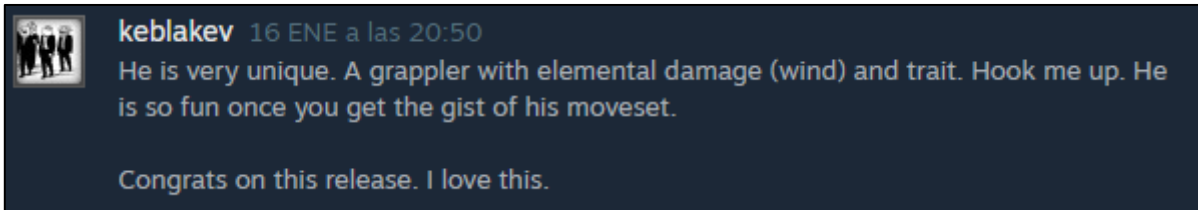


Ilustración 27. Comentario subido a la página pública de Steam del mod por uno de los jugadores



Conclusiones

6.1 Análisis de objetivos

Para comprobar si los objetivos del proyecto se han cumplido, se utilizarán los objetivos propuestos en el primer capítulo:

OE1: Realizar un análisis exhaustivo del género.

El objetivo se ha cumplido a través del análisis de todo el género de juegos de lucha 2D y juegos de lucha por plataformas, además del análisis de diferentes personajes con un arquetipo similar al del personaje que se planeaba crear.

OE2: Diseñar un personaje con la complejidad y diversidad de movimientos adecuada

El objetivo se ha cumplido creando un personaje que supera los estándares de complejidad del juego y mantiene la coherencia estética con el mismo a través del estudio del diseño y arte del juego.

OE3: Crear y desarrollar las animaciones y *sprites* necesarias para completar los aspectos visuales del personaje

El objetivo se ha cumplido ya que se han creado animaciones únicas para todas y cada uno de los movimientos del personaje y se han implementado dentro del juego

OE4: Integrar y programar el personaje

El objetivo se ha cumplido tras aprender y entender la versión modificada de GMEdit, permitiendo modificar la plantilla de personaje descargada para crear un personaje completamente funcional dentro del juego.

OE5: Desplegar el personaje a la plataforma de descarga de mods de Steam

El objetivo se ha cumplido, utilizando las herramientas internas del juego, se ha podido desplegar el personaje para que cualquier persona que tenga el juego en Steam pueda descargarlo e instalarlo con facilidad. Además de esto, cualquier persona que busque el nombre del personaje en dicha plataforma podrá encontrarlo.



OE6: Validar el personaje

El objetivo se ha cumplido, se han realizado pruebas a lo largo del tiempo y se ha recogido toda la retroalimentación generada por los jugadores, además de que se han llevado a cabo varias actualizaciones para arreglar errores y corregir problemas de balance del personaje.

Por último, el objetivo general, es decir, desarrollar un personaje de juego de lucha al completo, también ha sido completado, ya que el personaje está completo y es jugable por cualquier persona que quiera descargarlo.

Teniendo todo en cuenta, se han cumplido todos los objetivos del proyecto, completándolos uno por uno hasta tener un personaje completo y funcional subido a una plataforma desde la cual podrá ser descargado por cualquier persona que esté interesado en usarlo.

6.2 Lecciones aprendidas

Durante la creación de este proyecto se han cometido varios errores a través de los cuales se ha podido aprender para la creación de cualquier otro proyecto que quiera hacer. Ciertas decisiones con respecto al tiempo de desarrollo consideraban ciertas fases menos importantes de lo que finalmente han acabado siendo.

Por esto se han aprendido varias lecciones a lo largo del mismo que se consideran de especial importancia y que se cree que cualquier persona que quiera tratar de replicar este proyecto y crear su propio personaje, ya sea para modificar **Rivals of Aether** u otro juego cualquiera, o incluso crear un personaje desde cero para un juego propio, debe tener en cuenta.

- Es extremadamente importante planear el marketing del personaje. El principal problema que se ha encontrado este proyecto ha sido encontrar la suficiente gente para realizar pruebas tras su salida, así como obtener una retroalimentación variada de una gran cantidad de usuarios. Esto ha ocurrido porque el personaje fue subido sin previo aviso y el material de marketing no era lo suficientemente interesante para que los usuarios llegaran a probar el personaje, teniendo muchas menos visitas que otros personajes creados para el mismo concurso. Pese a que el objetivo principal del proyecto sea la creación en sí mismo de un personaje, es importante tomarse el tiempo adecuado para que dicho personaje llame la atención del público, ya que sin esto será especialmente complicado llevar a cabo actualizaciones y balancearlo adecuadamente.
- A la hora de crear animaciones para un personaje de un juego de lucha es importante que el inicio y final de estas animaciones sea claramente diferenciable, teniendo poses mucho más exageradas que en el resto de los videojuegos. Esto se debe a que los ataques suelen tener tiempos de inicio de menos de poco más que una décima de segundo, si es que llegan a una décima de segundo y por ello es importante que tanto el jugador como sus adversarios puedan reconocer al



instante lo que está ocurriendo para poder actuar en consecuencia. Es mejor tener un personaje que tenga movimientos exagerados que quizás sean menos cohesivos con su aspecto original, que tener movimientos que no sean diferenciables de su estancia neutral o de movimiento en combate. Además, debe ser muy claro cuando el personaje está en recuperación, para permitir a los oponentes reconocer cuando pueden castigar los malos hábitos del jugador que lo controla sin necesitar jugar al personaje ellos mismos.

- Las comunidades de creación de mods para juegos suelen tener servidores y foros donde recogen toda la información y herramientas que existen para ayudar a cualquier persona que quiera hacer un personaje. Se aconseja que se consulten dichos foros antes de empezar a crear el personaje, ya que las herramientas propuestas por la comunidad suelen hacer el proceso mucho más sencillo. Además de esto, pueden utilizarse estos foros para hacer preguntas y buscar soluciones a posibles problemas, ya que suelen tener gente activa que ha trabajado en diversos proyectos similares con anterioridad y podrán ayudar en lo que se necesite.
- Es especialmente complicado balancear correctamente un personaje para un juego que ya existe, esto requiere de cientos de horas de experiencia con el juego y de estudio de su escenario competitivo. Si no se tienen esta cantidad de tiempo para dedicarlo a la creación del personaje, el objetivo principal del desarrollo debe ser que el personaje sea divertido tanto para el jugador como para aquellos que jueguen contra él, sin dar tanta importancia a si el poder del personaje está todo lo cerca del de los personajes principales.
- En caso de querer tener una retroalimentación útil para el personaje, se aconseja encontrar a gente que tenga conocimientos sobre el juego con la que jugar tras haber experimentado con el personaje que se ha creado. Es complicado saber todas las posibilidades de un personaje para aquellos que no lo han diseñado, por lo que darle una explicación con relación al personaje y a los objetivos de su *gameplan* es importante para que puedan usarlo de manera efectiva y dar un *feedback* apropiado. Crear un grupo de personas con las que jugar y las cuales te expliquen de manera directa los puntos positivos y negativos del personaje ahorra mucho tiempo y dará una información de mucha más calidad para realizar parches de balance y arreglos de errores.



6.3 Líneas futuras

Tras la creación del mod en enero, se han realizado numerosas actualizaciones, hasta el punto en el que se considera que el personaje está bastante más balanceado que al inicio. Pese a esto, dicho personaje aun presenta varias carencias, principalmente en su arte y animaciones, que podrían recibir varias mejoras. En caso de querer continuar con el desarrollo del personaje, este sería el objetivo principal, actualizar sus animaciones para que tengan muchos más fotogramas y sean más fácilmente distinguibles ya que esta ha sido la queja principal que se ha obtenido durante la retroalimentación del personaje. Para esto, se utilizarían las antiguas como base, volviendo a utilizar los archivos de Aseprite que se utilizaron para su creación y modificando o añadiendo fotogramas para mejorar dichas animaciones. Además de esto, se pediría consejo y guía a la comunidad para tratar de que esta vez las animaciones estén más en línea con lo esperado de un personaje completo. Además, aún quedan varios problemas en su conjunto de movimientos y habilidades, ya que algunas personas consideran que es demasiado fuerte para un ámbito competitivo, por lo que se debería rediseñar o ajustar algunos de sus ataques para asegurarse de que su nivel de poder sea más cercano al de los personajes originales.



Bibliografía

- [1] F. Thomas and O. Johnston, *The illusion of life : Disney animation*. New York: Hyperion, 1995.
- [2] J. Cooper, *Game anim : video game animation explained*. Boca Raton, FL: CRC Press, 2021.
- [3] J. Pearl, *Becoming a Video Game Artist*. CRC Press, 2016.
- [4] H. Kuntjara and B. Almanfaluthi, "Character Design in Games Analysis of Character Design Theory," *Journal of Games, Game Art, and Gamification*, vol. 2, no. 2, Oct. 2021, doi: <https://doi.org/10.21512/jggag.v2i2.7197>.
- [5] L. Sheldon, *Character Development and Storytelling for Games*. CRC Press, 2022.
- [6] R. Pradantyo, M. V. Birk, and S. Bateman, "How the Visual Design of Video Game Antagonists Affects Perception of Morality," *Frontiers in Computer Science*, vol. 3, Apr. 2021, doi: <https://doi.org/10.3389/fcomp.2021.531713>.
- [7] Howard Sklar, "Narrative as Experience: The Pedagogical Implications of Sympathizing with Fictional Characters," *Partial Answers: Journal of Literature and the History of Ideas*, vol. 6, no. 2, pp. 481–501, 2008, doi: <https://doi.org/10.1353/pan.0.0023>.
- [8] E. Adams and J. Dormans, *Game mechanics : advanced game design*. Berkeley, Calif.: New Riders, 2012.
- [9] M. Salmond, *Video Game Design*. Bloomsbury Publishing, 2017.
- [10] E. Adams, *Fundamentals of game design*, 3rd ed. Berkeley, Ca: New Riders, 2014.

Ludografía

Guilty Gear: Strive	Arc System Works, 2021
King of Fighters XV	SNK, 2022
Ninja Gaiden	Team Ninja, Tecmo, 1988
Rivals of Aether	Dan Forance, 2017
Sekiro: Shadows Die Twice	From Software, Activision, 2019
Slap City	Ludosity, 2020
Street Fighter 2: The Wold Champion	Capcom, 1991
Super Smash Bros. Ultimate	Sora Ltd. y Bandai Namco, Nintendo, 2018
Multiversus	Player First Games, Warner Bros. Interactive Entertainment, 2024



Hojas de *sprites*

(Las hojas de *sprites* están recortadas y recolocadas para mejor visualización, cada página es una hoja de *sprites* diferente).





Universidad
Rey Juan Carlos

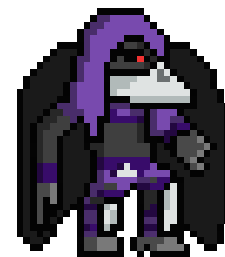
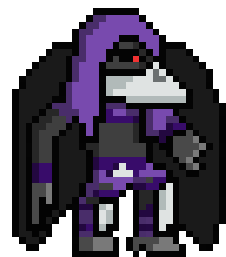
Escuela Técnica Superior
Ingeniería Informática





Universidad
Rey Juan Carlos

Escuela Técnica Superior
Ingeniería Informática





Universidad
Rey Juan Carlos

Escuela Técnica Superior
Ingeniería Informática





Universidad
Rey Juan Carlos

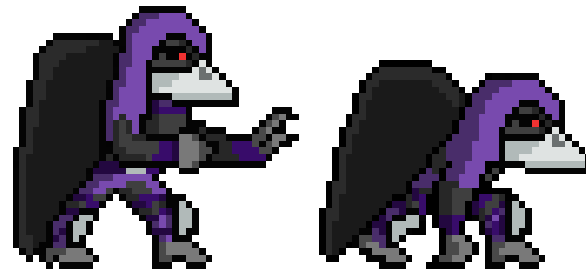
Escuela Técnica Superior
Ingeniería Informática





Universidad
Rey Juan Carlos

Escuela Técnica Superior
Ingeniería Informática





Universidad
Rey Juan Carlos

Escuela Técnica Superior
Ingeniería Informática





Universidad
Rey Juan Carlos

Escuela Técnica Superior
Ingeniería Informática





Universidad
Rey Juan Carlos

Escuela Técnica Superior
Ingeniería Informática





Universidad
Rey Juan Carlos

Escuela Técnica Superior
Ingeniería Informática



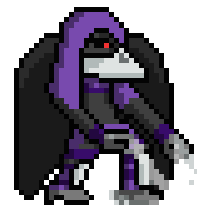


Universidad
Rey Juan Carlos

Escuela Técnica Superior
Ingeniería Informática









Universidad
Rey Juan Carlos

Escuela Técnica Superior
Ingeniería Informática







Universidad
Rey Juan Carlos

Escuela Técnica Superior
Ingeniería Informática





Universidad
Rey Juan Carlos

Escuela Técnica Superior
Ingeniería Informática





Universidad
Rey Juan Carlos

Escuela Técnica Superior
Ingeniería Informática





Universidad
Rey Juan Carlos

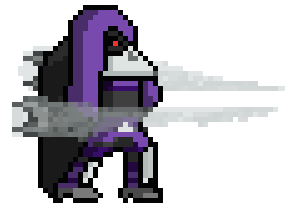
Escuela Técnica Superior
Ingeniería Informática





Universidad
Rey Juan Carlos

Escuela Técnica Superior
Ingeniería Informática





Universidad
Rey Juan Carlos

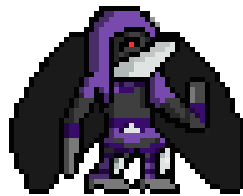
Escuela Técnica Superior
Ingeniería Informática





Universidad
Rey Juan Carlos

Escuela Técnica Superior
Ingeniería Informática





Universidad
Rey Juan Carlos

Escuela Técnica Superior
Ingeniería Informática

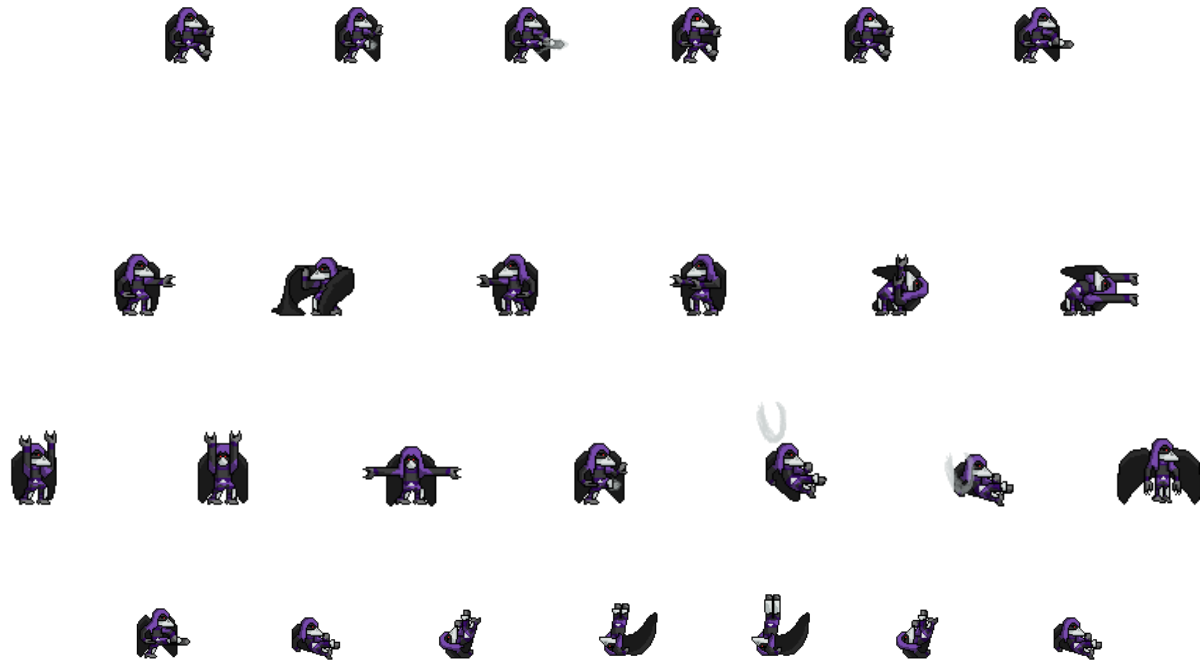




Universidad
Rey Juan Carlos

Escuela Técnica Superior
Ingeniería Informática







Universidad
Rey Juan Carlos

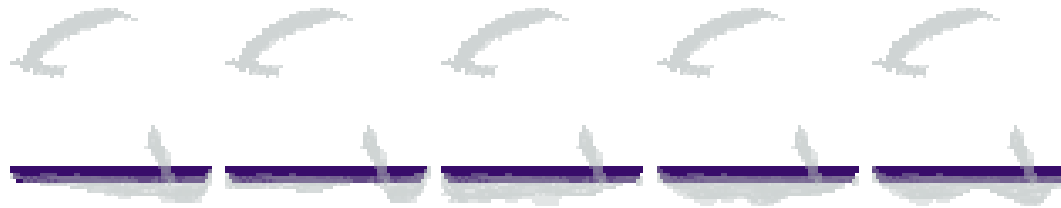
Escuela Técnica Superior
Ingeniería Informática





Universidad
Rey Juan Carlos

Escuela Técnica Superior
Ingeniería Informática





Universidad
Rey Juan Carlos

Escuela Técnica Superior
Ingeniería Informática







Universidad
Rey Juan Carlos

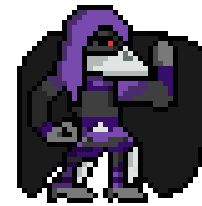
Escuela Técnica Superior
Ingeniería Informática





Universidad
Rey Juan Carlos

Escuela Técnica Superior
Ingeniería Informática











Universidad
Rey Juan Carlos

Escuela Técnica Superior
Ingeniería Informática





Universidad
Rey Juan Carlos

Escuela Técnica Superior
Ingeniería Informática





Universidad
Rey Juan Carlos

Escuela Técnica Superior
Ingeniería Informática





Universidad
Rey Juan Carlos

Escuela Técnica Superior
Ingeniería Informática





Universidad
Rey Juan Carlos

Escuela Técnica Superior
Ingeniería Informática

