

COMP4998 Reporte

Aramis Matos, Lenier Gerena, Jorge Huertas

Primer Semestre, 2022-2033

Contents

1	Introducción	2
	Introducción	2
1.1	Antecedentes	2
	Antecedentes	2
1.2	Objetivos	3
	Objetivos	3
1.2.1	Objetivo General	3
	Objetivo General	3
1.2.2	Objetivos Específicos	3
	Objetivos Específicos	3
1.3	Justificación	3
	Justificación	3
1.4	Alcance y Limitaciones	4
	Alcance y Limitaciones	4
2	Fundamentos Teóricos	5
	Fundamentos Teóricos	5
	Acronyms	7
	References	8

Chapter 1

Introducción

1.1 Antecedentes

El cálculo de ventaja de "frame data", por lo general, ha sido una tarea que se hace a mano (,) y (Dustloop, 2022). Esto se debe a que los números involucrados son pequeños por lo general y por ende, no se ha necesitado una gran cantidad de recursos computacionales. Sin embargo, con la venida de torneos de video juegos competitivos (,), se ha visto una necesidad de conocer estados de ventaja rápida y efectivamente.

HADOOKEN (,)

Por la sencillez de los cálculos, no se ha innovado mucho en el espacio de calculadoras de frame data. Mas aún, las diferencias en mecánicas internas entre juegos ha complicado el asunto de una calculadora generalizada pero han habido algunos intentos para crear una, entre ellas están:

- FAT - Frame Data! (,)
- Smash Ultimate Calculator (,)

FAT - Frame Data! es una aplicación móvil que provee una gran cantidad de estadísticas para los juegos Street Fighter III: 3rd Strike (,), Ultra Street Fighter IV (,), Street Fighter V (,) y Guilty Gear Strive (,). Entre sus funcionalidades están tablas completas de Los valores de startup frames, active frames, recovery frames y blockstun para una movida (frame data), búsqueda de frame data mediante el nombre del personaje y una movida, listas completas de movidas y su entrada correspondiente en el control, comparación de estadísticas internas de cada personaje y una calculadora de frame data. Sin embargo, la calculadora no calcula estados de ventaja, sino para calcular si es posible que el oponente ataque entre dos movidas si la primera es bloqueada. No se calcula estados de ventaja y desventaja. En términos de interfaz de usuario,

la aplicación no se ve mal estéticamente. Utiliza el negro como color de fondo y el azul para detalles. La aplicación presenta la información claramente y segmentada lógicamente para así no intimidar al usuario. Entre todo, es un interfaz bien hecho.

Smash Ultimate Calculator es una aplicación web que calcula la cantidad de daño que sufre el oponente y la cantidad de daño necesaria para que el oponente pierda una vida en el juego Super Smash Bros. Ultimate (?, ?). Es notable que esta aplicación solo se puede aplicar al juego de Super Smash Bros. Ultimate, ningún otro. Hay varias razones por la que esto es así. Primero, Super Smash Bros. Ultimate no es un juego de pelea tradicional ya que sus condiciones de victoria y manera de mover el personaje son radicalmente distintas a juegos como Street Fighter V y Guilty Gear Strive. Segundo, Super Smash Bros. Ultimate tiene unos mecanismos que divergen lo suficiente para que no apliquen las reglas de los juegos anteriores en su serie como Super Smash Bros. Melee (?, ?). Es por estas razones que Smash Ultimate Calculator no tiene aplicabilidad fuera del juego de la cual fue diseñado. Aparte de esto, la interfaz se ve primitiva. Muchos de los items en la páginas se ven dispersos y sin clara definición de donde elementos relacionados empiezan y terminan. A la misma vez, el interfaz se ve demasiado ocupado como que intimida y confunde al usuario.

En conclusión, a pesar de algunos buenos intentos como FAT - Frame Data!, no hay muchas aplicaciones que presenten las estadísticas de estado de ventaja y desventaja de manera limpia y concisa.

1.2 Objetivos

1.2.1 Objetivo General

El grupo desea crear un interfaz mas amigable y atractivo para una aplicación de cálculo de estado de ventaja para juegos de pelea.

1.2.2 Objetivos Específicos

El grupo tiene como objetivo tres metas:

- Rehacer y mejorar el interfaz de la aplicación SkyboundDB (?, ?)
- Volver a implementar el interfaz de la aplicación en HTML y CSS
- Si es posible, volver a implementar la aplicación en JavaScript en vez de Python

1.3 Justificación

SkyboundDB se creó originalmente como un proyecto final para otro curso. Sin embargo, las razones por las cuales se decidió crear el proyecto original fueron

varias. Uno de autores principales de SkyboundDB, Lenier Gerena, siempre ha sido un jugador ávido de los juegos de pelea y participante activo en torneos competitivos de tales juegos. Siempre ha querido de contribuir algo a la comunidad que le ha ofrecido tantas horas de entretenimiento y un sinnúmero de amistades. Este deseo se combinó con el deseo del segundo autor principal, Aramis Matos, de crear una aplicación gráfica de usuario. Por esto, SkyboundDB resultó ser una aplicación con GUI que calcula estados de ventaja y desventaja para algunos personajes del juego Granblue Fantasy Versus (?, ?).

SkyboundDB se creó con el propósito de simplificar el proceso de experimentación necesario para ver que movidas se pueden usar para castigar a otras. Por lo general, esto es un proceso iterativo que requiere paciencia e intuición debido a la gran cantidad de posibles combinaciones y variables. Esto es un proceso frustrante que realmente desilusiona al principiante de los juegos de pelea. Esta es otra razón por la cual se desarrolló SkyboundDB.

1.4 Alcance y Limitaciones

Se piensa rehacer el interfaz de las 4 pantallas principales que tiene SkyboundDB:

- Selección de personajes (pantalla principal)
- Selección de movida de un personaje
- Comparación de una movida con otra
- Comparación de una movida con todas otras movidas de otro personaje

Además, si es posible, se gustaría volver a implementar el programa en HTML, CSS y JavaScript para así tener un prototipo funcional.

En términos de limitaciones, no se piensa expandir al la cantidad secciones que tiene el programa por cuestiones de tiempo. Solo se creará el prototipo funcional si es posible dado el tiempo, no es una garantía que se valla hacer.

Otra limitación con respecto a este proyecto es el juego en que se estará importando y procesando la información sobre, pues solamente será Granblue Fantasy Versus (GBVS). Muchos juegos de pelea fundamentalmente tienen la misma información que GBVS tiene para sus movidas, y las calculaciones posibles se forman de la misma manera a través de todos los juegos de pelea. Sin embargo, importar esta información de otros juegos, al igual que verificar si en realidad tienen convalidación entre cada uno sin tener que tomar en cuenta que hay más/menos información de movidas (un juego que use un esquema de 5 botones o más en vez de cuatro como GBVS) requeriría su propia implementación por separado y tiempo.

Chapter 2

Fundamentos Teóricos

El propósito principal de un juego de pelea es la interacción entre dos personajes que mediante sus propias herramientas únicas puedan restar el recurso de su vida (o Health Points, HP (*The Fighting Game Glossary* / *Infil.Net*, n.d.)) igual a 0. Bajo esta premisa sencilla, uno puede asumir que la mejor manera de llegar a dicha meta es usar todo movimiento que sea ofensivo y de mayor daño, pero dichas herramientas ofensivas pierden contra las herramientas defensivas. Ser defensivo no te lleva más cerca a la condición de ganar, sino que te mantiene sin perderla (mientras las herramientas se usen correctamente). Esto lleva a una interacción más compleja entre los jugadores, llevando el juego así a un extremo donde se quiere saber en qué momento es mejor tomar una por la otra para alcanzar el requisito de la victoria. Aquí entonces en donde importa el conocimiento de frame advantage relativamente a cada personaje.

La mayoría de los juegos de pelea (y en todos los juegos de peleas modernos) corren consistentemente a 60 “frames per second”. Esto dicta qué tan bien se ve el juego cuando está corriendo, pues es la cantidad de imágenes que se procesan en dicho juego por segundo, pero también es una base principal para cómo funcionan las animaciones de dicho juego de pelea. Un “frame” se usa como una unidad básica de tiempo relativo ($1 \text{ frame} = 1/60 \text{ segundos}$) y es lo que permite que se pueda calcular el frame data de los personajes en el juego. Para propósitos competitivos, no se toma la idea de que se pueda reaccionar a lo que sería un frame, pues estarías reaccionando a algo que es imposible para la capacidad humana, sino se toma esta unidad para extraer situaciones de ventaja y desventaja en ciertas situaciones donde ambos jugadores presionan un botón al unísono (Fighting Game glossary, Infil, 2022).

Por consiguiente, es necesario conocer un poco acerca de la terminología que se utiliza en el espacio de los video juegos de pelea. Por lo general, los juegos de pelea se juegan con dos jugadores que se atacan con ciertas movidas. Estas movidas consumen un tiempo particular en completar. A continuación se

presenta una lista definiendo términos importantes:

- Movida que un jugador seleccionó (move)
- Cantidad de frames para comenzar una movida (startup frames)
- Cantidad de frames en la cual es posible colisionar con el opnente (active frames)
- Cantidad de frames en la cual es imposible admitir otra movida (recovery frames)
- Cantidad de frames en que el atacado no puede hacer nada aparte de bloquear (blockstun)
- Cantidad de frames de ventaja o desventaja luego de restar recovery frames y blockstun (on block)

frame data son particulares para cada movida de un personaje. A consecuencia de esto, es posible que algunas movidas sean mas rápidas que otras. Este dato es importante a la hora de calcular la ventaja que tiene una movida contra otra.

Vamos asumir que hay una movida A que tiene 25 startup frames, 3 active frames, 30 recovery frames y -14 on block y una movida B que tiene 7 startup frames, 3 active frames, 6 recovery frames y $+3$ on block. Si el personaje de la movida A ataca al personaje de la movida B pero el personaje de la movida B bloquea la movida, ahora el personaje A experimenta recovery frames. Ahora, el personaje de la movida A no puede hacer nada por 30 frames. Al solo tener 16 frames de blockstun, el personaje de la movida A experimenta 14 que no puede hacer nada pero puede ser atacado. A este estado se le llama desventaja. Al mismo tiempo, el personaje de la movida B tiene 14 frames para hacer lo que quiera. Como su movida solo se tarda 7 frames en salir, puede atacar al personaje de la movida A sin miedo de ser atacado.

Acronyms

active frames Cantidad de frames en la cual es posible colisionar con el oponente 6

blockstun Cantidad de frames en que el atacado no puede hacer nada aparte de bloquear 6

frame data Los valores de startup frames, active frames, recovery frames y blockstun para una movida 2, 6

on block Cantidad de frames de ventaja o desventaja luego de restar recovery frames y blockstun 6

recovery frames Cantidad de frames en la cual es imposible admitir otra movida 6

startup frames Cantidad de frames para comenzar una movida 6

References

Dustloop. (2022, August). *Using Frame Data*.
https://www.dustloop.com/w/Using_Frame_Data.
The Fighting Game Glossary / *infil.net*. (n.d.).
<https://glossary.infil.net/?t=Health%20Bar>.