Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación
IIC2113 Diseño Detallado de Software

Proyecto: Shin Megami Tensei

Francisco Ignacio Gazitúa Requena
Cristian Andrés Hinostroza Espinoza

1. Introducción

Shin Megami Tensei es una serie de videojuegos del género JRPG desarrollados por Atlus, siendo parte de la franquicia de videojuegos *Megami Tensei*.

Su origen se remonta a *Digital Devil Story: Megami Tensei*, primer juego de la franquicia *Megami Tensei* lanzado para la Famicom en 1987 para el mercado japonés. Sin embargo, la serie empezó con el lanzamiento de *Shin Megami Tensei* para la Super Famicom en 1992, también para el mercado japonés. Desde entonces, la serie ha evolucionado su sistema de juego y ha alcanzado reconocimiento internacional, con su entrega más reciente, *Shin Megami Tensei V: Vengance*, siendo lanzada internacionalmente en 2024 para *Nintendo Switch*, *Playstation 4*, *Playstation 5*, *Xbox One*, *Xbox series X/S* y *Microsoft Windows*. Actualmente la serie cuenta con 9 entradas en su serie principal y 4 spin-offs.

Su estilo de juego es más conocido por la serie *Persona*, otra serie de la franquicia *Megami Tensei* que combinó un sistema de combate más accesible con elementos de simulación social. Uno de sus personajes principales, Joker, es parte del elenco de la serie Super Smash Bros.



Figura 1: All-Out attack de Joker en Persona 5

El objetivo de este proyecto es implementar una versión simplificada y modificada del sistema de combate *Press Turn* de *Shin Megami Tensei IV*. A grandes rasgos, el juego consiste en un enfrentamiento entre dos equipos, quienes buscarán explotar las debilidades de su rivales mientras evitan que este pueda hacer lo mismo.

2. Setup

En el juego, dos jugadores enfrentarán a sus equipos en un combate hasta conseguir la victoria. En cada equipo existen dos tipos de unidades, los samurai, quienes lideran el equipo, y los ~~demonios~~ monstruos, quienes apoyan al samurai en el combate. Los jugadores armaran sus equipos de la siguiente manera:

- Cada jugador podrá escoger un samurai y un máximo de 7 monstruos para conformar su equipo.
- Cada equipo debe tener un solo samurai. No pueden haber equipos con más de un samurai o sin un samurai.
- El jugador podrá escoger un máximo de 8 habilidades para su samurai. Este también podría no tener habilidades
- Un samurai no podrá tener habilidades repetidas. Por ejemplo, si tiene la habilidad **Agi**, no podrá tener un segundo **Agi** en su lista de habilidades.
- Un equipo podría estar conformado solo de un samurai, sin monstruos.
- Un jugador no podrá tener monstruos repetidos en su equipo, pero su contrincante podrá tener los mismos monstruos que el jugador. Por ejemplo, si el primer jugador tiene al monstruo **Jack Frost** en su equipo, no podrá tener un segundo **Jack Frost**, pero el otro jugador si podrá tener un único **Jack Frost** en el mismo juego.

Luego de armar los equipos, las unidades del jugador se organizarán en dos posiciones, en un tablero y en la reserva. El tablero cuenta con cuatro puestos activos por jugador, en los cuales estos podrán organizar a su samurai y a tres de sus monstruos.

Al iniciar el juego, el samurai y los tres primeros monstruos seleccionados se organizarán en el tablero, donde el samurai se colocará en la posición de más a la izquierda y los monstruos se ordenarán de izquierda a derecha en los otros tres puestos en el orden en que fueron seleccionados por el jugador. Si el jugador seleccionó menos de 3 monstruos, entonces quedarán espacios vacíos.

Veamos esto con un ejemplo. Supongamos que el primer jugador seleccionó al samurai **Flynn** y a los monstruos **Jack Frost**, **Black Frost**, **King Frost** y **Frost Ace**. Adicionalmente, consideremos que el segundo jugador escogió al samurai **Kei** y a los monstruos **Pyro Jack** y **Jack Ripper**. Una vez terminada la selección de ambos equipos sucederá lo siguiente:

1. En el puesto de más a la izquierda del primer jugador se posicionará a **Flynn**. En el puesto equivalente del segundo jugador se posicionará a **Kei**.
2. En los siguientes tres puestos de izquierda a derecha del primer jugador se posicionarán, en orden, a **Jack Frost**, **Black Frost** y a **King Frost**. El monstruo **Frost Ace** quedará en la reserva del jugador.
3. En los siguientes dos puestos del segundo jugador se posicionará a **Pyro Jack** y a **Jack Ripper**. El puesto de más a la derecha quedará vacío y este jugador no tendrá unidades en reserva.

Podemos ver esto ilustrado en la siguiente figura:

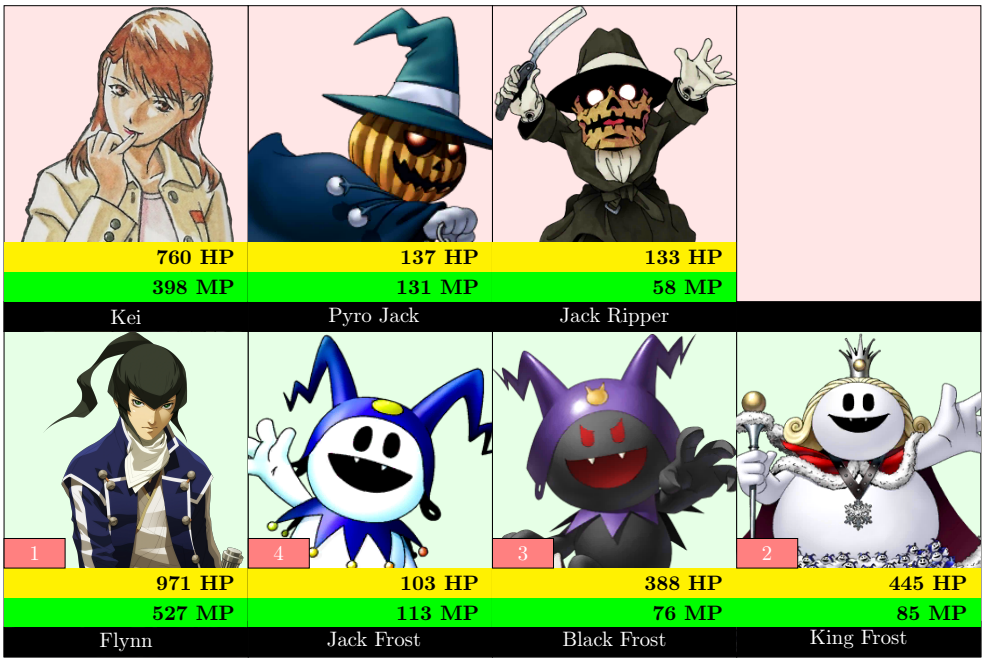


Figura 2: Tablero con los dos equipos. Abajo se encuentra el equipo del primer jugador y arriba el del segundo jugador. Se omite a **Frost Ace** dado que se encuentra en la reserva.

3. Unidades

Antes de ahondar en el flujo del juego, revisaremos los distintos atributos de las unidades. La diferencia entre samurai y monstruo solo se vuelve importante cuando el juego inicia, por lo que las características de las que se hablará en esta sección aplican tanto a ambos tipos de unidad.

Cada unidad del juego cuenta con distintas características, las cuales serán fundamentales al momento de determinar el flujo del juego. Los atributos de las unidades corresponden a:



(a) Jack Frost de Shin Megami Tensei IV

- **Nombre**: Es el nombre de la unidad, es su identificador único.
- **Stats**: Son una serie de números que determinarán qué tan buena es la unidad en distintos roles.
- **Habilidades**: Son habilidades que la unidades pueden usar para atacar o provocar efectos especiales en el juego
- **Afinidad**: Determina si la unidad será resistente o débil contra algún tipo de ataque.

En cuanto a estos atributos, la única diferencia entre un monstruo y un samurai es que los primeros tienen un set de habilidades pre-definido, mientras que los últimos pueden ser equipados con habilidades por el jugador.

3.1. Stats

Los stats determinan distintos elementos del juego, siendo muy relevantes al momento de determinar como sucederá el flujo del juego en su totalidad. Los stats son personales para cada unidad y son los siguientes:



(a) Demi-Fiend de Shin Megami Tensei III: Nocturne

- **HP máximo**: Determina la vitalidad máxima de la unidad
- **HP actual**: Determina la vitalidad actual de la unidad. Si este valor llega a 0, la unidad muere. Este valor se encuentra entre 0 y HP máximo. Este stat suele llamarse simplemente HP.
- **MP máximo**: Determina el Mana máximos de la unidad
- **MP actual**: Determina el Mana actuales de la unidad. Estos puntos pueden ser utilizados para utilizar habilidades. Este valor se encuentra entre 0 y MP máximo. Este stat suele llamarse simplemente MP.
- **Str**: Determina la potencia de los ataques físicos.
- **Skl**: Determina la potencia de los disparos.
- **Mag**: Determina la potencia de los ataques mágicos.
- **Spd**: Determina el orden de los ataques.
- **Lck**: Determina el grado de efectividad de ciertas habilidades.

3.2. Afinidades

En el juego existen distintos tipos de ataques, los cuales pueden resultar útiles en diversas situaciones. Cada unidad tiene un una afinidad asociada a cada uno de los siguientes tipos de ataque, también llamados elementos: **Phys**, **Gun**, **Fire**, **Ice**, **Elec**, **Force**, **Light** y **Dark**.

Las afinidades definen que tipo de reacción tiene cada unidad a los ataques que recibe del rival. Existen 6 tipos de afinidad en el juego:



(a) Pyro Jack de Shin Megami Tensei IV

- **Neutral (-)**: La unidad no es particularmente débil o resistente al elemento de ataque
- **Weak (Wk)**: La unidad es particularmente débil contra el elemento del ataque
- **Resist (Rs)**: La unidad es resistente al elemento del ataque
- **Null (Nu)**: La unidad nulifica todos los ataques del elemento
- **Repel (Rp)**: La unidad devuelve el ataque al rival que lo lanzó
- **Drain (Dr)**: El ataque no dañará a la unidad, sino que la curará

4. Sistemas de juego

Una vez conformado los equipos, los jugadores deberán tomar distintas decisiones que definirán como sucederá cada ronda.

Antes de ahondar en las mecánicas del combate, el jugador debe comprender dos elementos esenciales del juego: el sistema de turnos y las acciones que pueden realizar las unidades.

4.1. Acciones

Las unidades podrán realizar diversas acciones cuando sea su turno de actuar, las que pueden tener diversos efectos en el juego.

En este punto hay una diferencia entre samurai y monstruos. Los samurai en general tienen más acciones entre las que pueden escoger en comparación a los monstruos, además de que algunas acciones tienen efectos distintos si la realiza un samurai o un monstruo. En particular, las acciones que cada tipo de unidad puede tomar son las siguientes:

	Samurai	Monstruo
Atacar	Realiza un ataque básico tipo **Phys**	
Disparar	Realiza un ataque básico tipo **Gun**	No tiene la opción
Usar Habilidad	Selecciona una habilidad para ser utilizada	
Invocar	Invoca un monstruo vivo de la reserva. Este puede colocarse en un puesto vacío o ser intercambiado por un monstruo en los puestos activos, pero no por el samurai	Invoca un monstruo vivo de la reserva. Este será intercambiado por este monstruo que lo invocó
Pasar Turno	Termina el turno automáticamente	
Rendirse	Termina el juego y pierde automáticamente	No tiene la opción

Cuadro 1: Acciones del juego separadas por samurai y monstruo

4.2. Turnos

El elemento más importante el sistema de combate es el manejo de turnos, el cual incentiva a los jugadores a tomar decisiones de tal forma que maximice la cantidad de turnos que tiene disponible y minimice la cantidad de turnos disponibles del rival.

En el juego, un turno corresponde a una acción que puede realizar una unidad, por lo que tener más turnos significa tener la posibilidad de atacar más veces, reorganizar al equipo más cómodamente o tomar más acciones que perjudiquen al rival. Existen dos tipos de turnos:

Full Turn: permite a la unidad realizar una acción. La característica principal de este tipo de turnos es que una vez usados pueden convertirse en **Bliking Turns**, permitiendo al jugador realizar una acción adicional.

Blinking Turn: este tipo de turnos es básicamente lo mismo que un **Full Turn**, sin embargo, estos no pueden convertirse en otro tipo de turno.

El proceso de conseguir o perder más turnos de lo normal dependerá de diversos factores, pero el principal será la afinidad que tenga el enemigo a los ataques realizados por el jugador. La siguiente tabla detalla el flujo de los turnos para distintas acciones:

Acción		Efecto
Atacar al rival	Repel / Drain	Consume todos los turnos
	Null	Consume 2 **Blinking Turns**. Si no hay suficientes, consume lo que falte en **Full Turns**
	Miss	Consume un **Blinking Turn**. Si no hay, consume un **Full Turn**
	Weak	Consume un **Full Turn** y otorga un **Blinking Turn**. Si no hay, consume un **Blinking Turn**
	Neutral/ Resist	Consume un **Blinking Turn**. Si no hay, consume un **Full Turn**
Invocar/Pasar Turno		Consume un **Blinking Turn**. Si no hay, consume un **Full Turn** y otorga un **Blinking Turn**
Usar habilidad no ofensiva		Consume un **Blinking Turn**. Si no hay, consume un **Full Turn**

Cuadro 2: Efecto de ciertas acciones sobre los turnos del jugador. Las diversas acciones de la tabla serán explicadas más adelante

Si es que un ataque tiene como objetivo a más de una unidad simultáneamente y estas unidades tienen distintas afinidades sobre el elemento del ataque, entonces solo tendrá efecto en los turnos la afinidad con mayor prioridad.

La prioridad de las afinidades se encuentran en orden descendiente en el Cuadro 2, es decir, las afinidades **Repel** y **Drain** tienen prioridad sobre **Null**, el que tiene prioridad sobre **Miss**[1], el que a su vez es más prioritario que **Weak**; todas esas afinidades son más prioritarias que **Neutral** y **Resist**.

5. Flujo del juego

Ahora que conocemos los elementos y los sistemas del juego, ahondaremos en el flujo general del programa.

En situaciones normales, el flujo del juego será el siguiente:

- Inicia la ronda del jugador 1. El jugador comenzará con una cantidad de **Full Turns** igual a la cantidad de unidades vivas que tenga en sus puestos activos en el tablero.
- Por cada uno de sus turnos, el jugador podrá hacer que una de sus unidades realice una acción. El resultado de la acción tendrá un efecto en los turnos del jugador igual al señalado en el cuadro 2.
- El orden en que las unidades actuarán dependerá de su Spd. La unidad con mayor Spd actuará de los primeros y la con menos Spd actuará al final. Si dos unidades tienen el mismo Spd, entonces actuará primero la que esté más a la izquierda en el tablero.
- Si en cualquier punto del combate un monstruo muere, este será enviado a la reserva y el puesto que ocupaba en el tablero quedará vacío.
- Si en cualquier punto del combate el samurai muere, este permanecerá muerto en la misma posición del tablero en la que se encuentra y no podrá actuar.
- Una vez que se consuman todos los turnos del jugador comenzará la ronda de su rival. El cual sucederá de la misma forma y bajo las mismas reglas.

[1]Más adelante se explicará que es **Miss**

- Si en cualquier punto del juego un jugador se queda sin unidades activas en sus puestos activos, aunque tenga unidades vivas en la reserva, el juego terminará y ese jugador será declarado el perdedor.

5.1. Orden de acciones

Como se mencionó en el listado anterior, el orden en que las unidades podrán actuar dependerá de cómo se comparen sus stats. A continuación ahondaremos en cómo distintas interacciones pueden hacer que este orden cambie.

Como se mencionó anteriormente, al iniciar la ronda de un jugador se definirá el orden en que actuarán las unidades. El orden será descendente en relación al Spd de las unidades; sin embargo, si dos unidades tienen el mismo Spd, entonces actuará primero la que esté más a la izquierda en el tablero.

Veamos esto con un ejemplo. Supongamos que al iniciar la ronda del jugador 1 este tiene en sus puestos activos, ordenados de izquierda a derecha, a las siguientes unidades:

1. **Joker**, con $Spd = 54$
2. **Agathion**, con $Spd = 15$
3. **Leanan Sidhe**, con $Spd = 15$
4. **Nue**, con $Spd = 32$

En la situación anterior, al iniciar la ronda, el orden en que actuarán las unidades será **Joker**, **Nue**, **Agathion** y, por último, **Leanan Sidhe**. Podemos ver esto ilustrado en la siguiente figura:

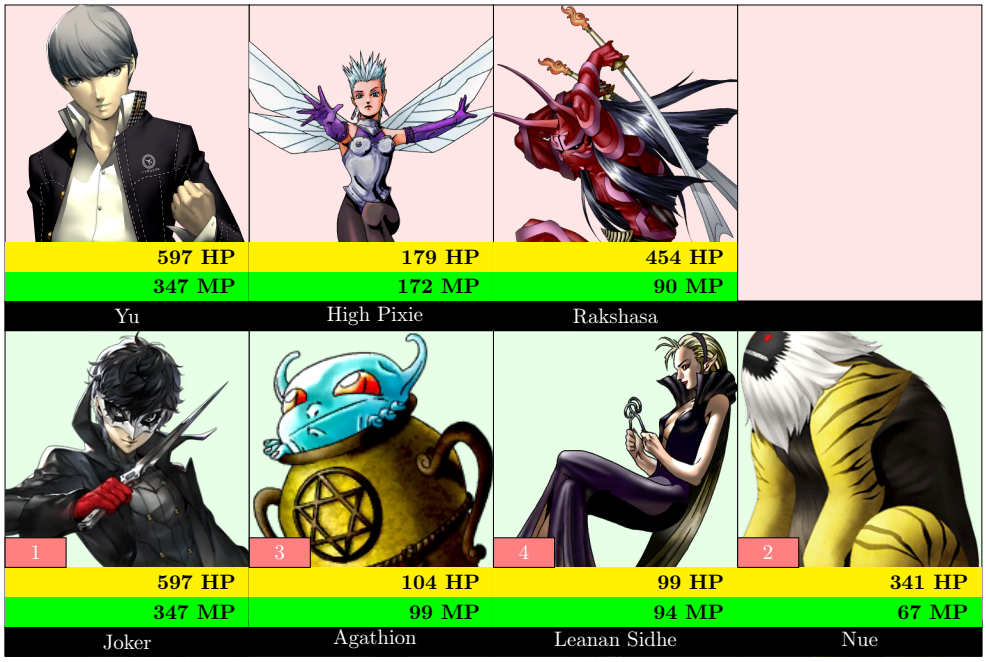


Figura 6: Tablero con el orden del equipo de **Joker**. El número dentro de los cuaros rojos indica el orden en que actuará la unidad.

Luego de que una unidad actué, el orden avanzará. Es decir, la unidad que acaba de actuar se colocará al final de la fila, dejando a actuar al resto de unidades. Tomando el ejemplo anterior, si **Joker** realiza una acción,

entonces el orden cambiará a **Nue**, **Agathion**, **Leanan Sidhe**, **Joker** y **Nue** podrá actuar. Podemos ver este cambio ilustrado en la siguiente figura:

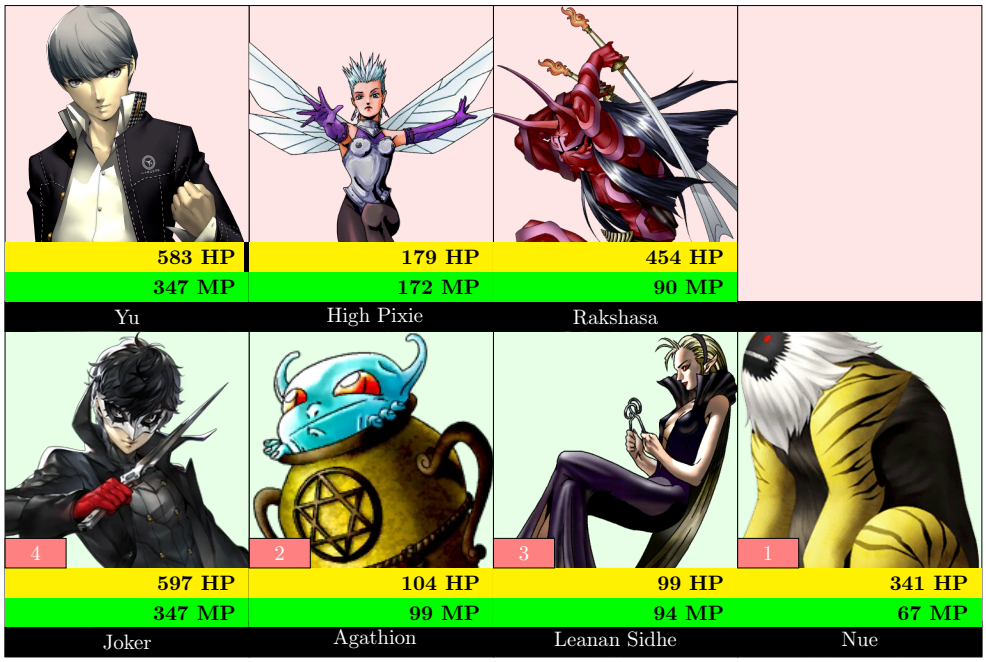


Figura 7: Tablero con el orden del equipo de **Joker** luego de realizar una acción.

Si una unidad es intercambiada por otra, entonces el orden se mantendrá, pero la unidad que ha sido invocada tomará el lugar de la unidad que se ha retirado, independiente de su Spd. Volviendo al ejemplo anterior, supongamos que **Nue** tiene una habilidad que le permite intercambiar a **Agathion** por **Melchom**, quien tiene $Spd = 10$. Si **Nue** utiliza la habilidad, entonces el orden del juego pasará a ser **Melchom**, **Leanan Sidhe**, **Joker**, **Nue**, esto porque **Melchom** a tomado el lugar de **Agathion** como la siguiente unidad en la lista. Podemos ver esto en la siguiente figura:

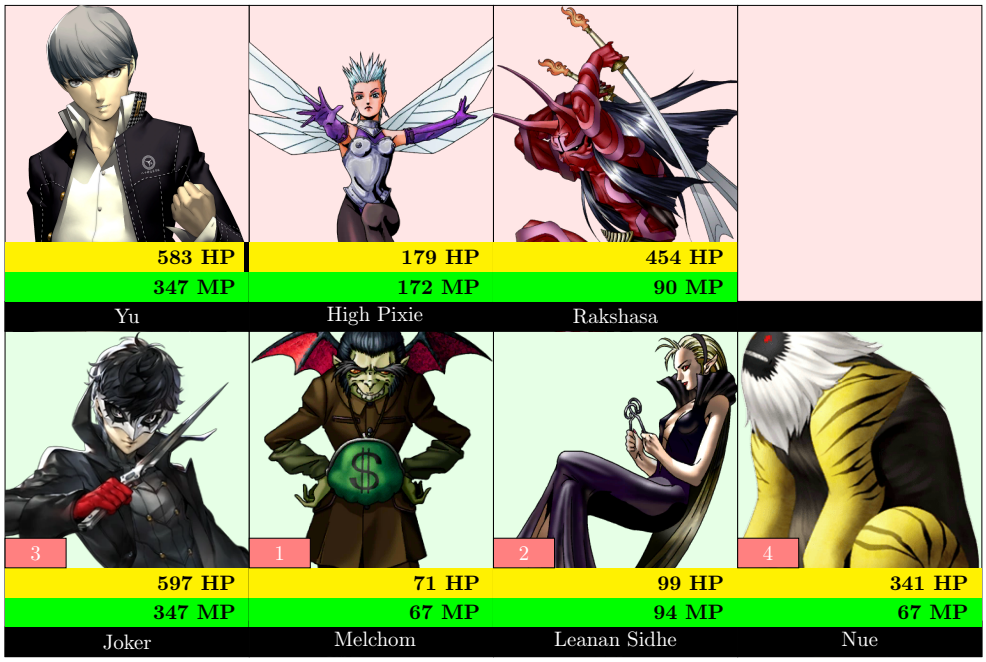


Figura 8: Tablero con el orden del equipo de **Joker** luego de intercambias a un monstruo por otro

Si un monstruo es invocado a un espacio vacío, entonces este se colocará al final de la lista del orden de acciones y luego se moverá la lista a la siguiente unidad, independiente de su Spd. Veamos esto con un ejemplo: supongamos que para el ejemplo anterior ha terminado la ronda del jugador 1 y ha comenzado la del jugador 2. Apenas inicie la ronda del jugador 2 el orden en que actuarán las unidades será **Yu**, **Rakshasa** y, finalmente, **High Pixie**. Esto lo podemos ver en la siguiente figura

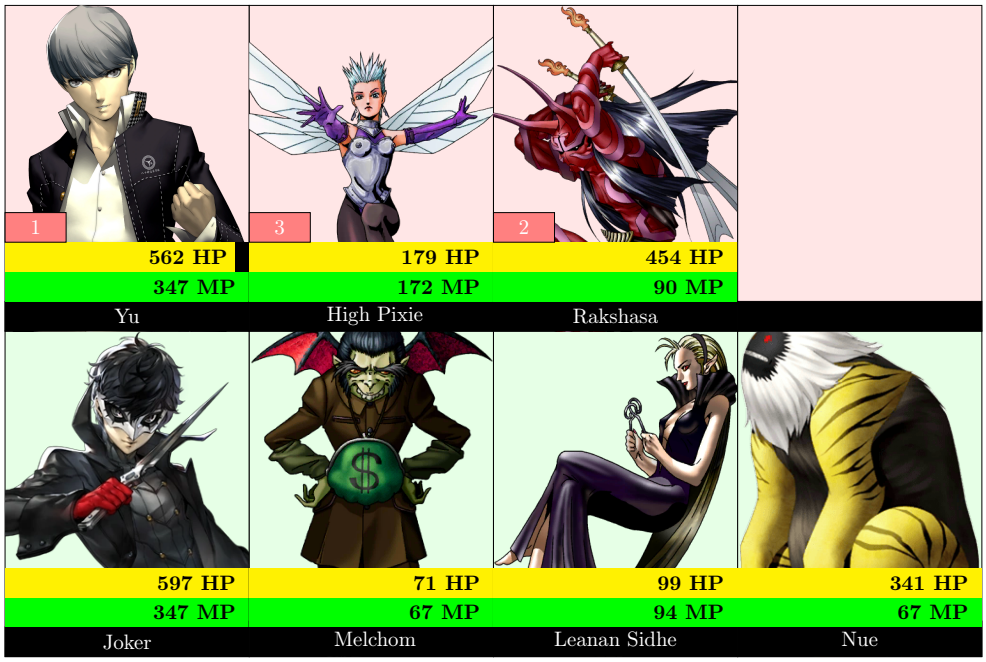


Figura 9: Tablero con el orden del equipo de **Yu** al partir su ronda.

Si **Yu** invoca al monstruo **Yamata-no-Orochi** al puesto vacío, entonces este se pondrá al final de la lista, es decir, inmediatamente despues de **High Pixie**. Como **Yu** acaba de actuar, la lista avanzará y el nuevo orden será **Rakshasa**, **High Pixie**, **Yamata-no-Orochi** y, por último, **Yu**. Esto se ve ilustrado en la siguiente figura:

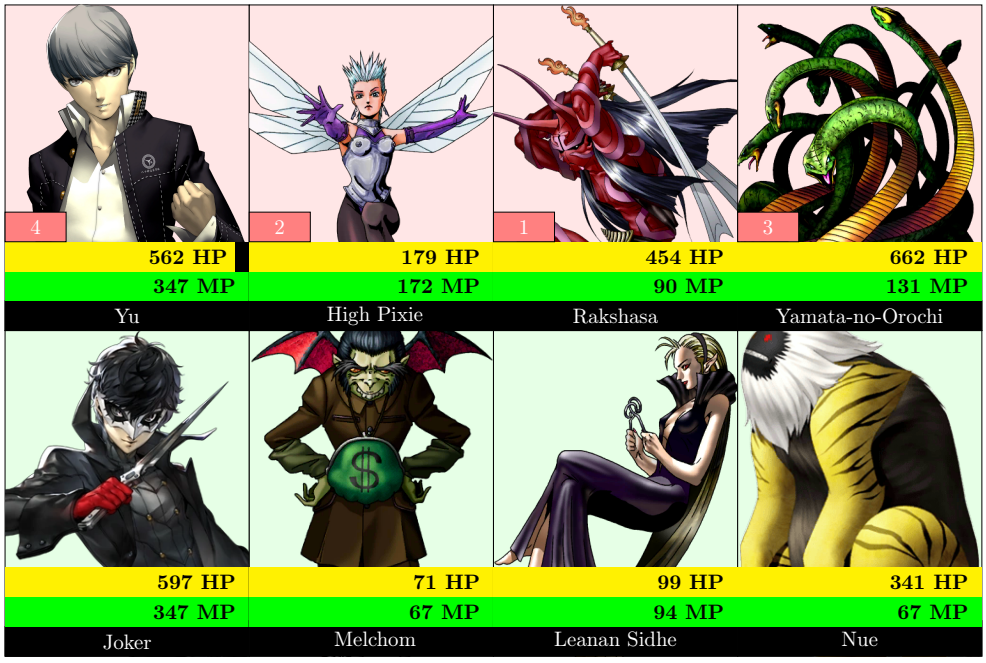


Figura 10: Tablero con el orden del equipo de **Yu** luego de invocar a **Yamata-no-Orochi**

Por último, el orden en que actuarán las unidades se reiniciará al iniciar una nueva ronda, es decir, cuando inicie una nueva ronda el orden que las unidades actuarán dependerá de como se comparen sus Spd. Supongamos que ha terminado el turno del segundo jugador y comienza nuevamente el del primer jugador. En esta situación el orden en que actuarán las unidades, producto de la comparación de sus Spd, será **Joker**, **Nue**, **Leanan Sidhe** y, por último, **Melchor**. Esto lo podemos ver en la siguiente figura:

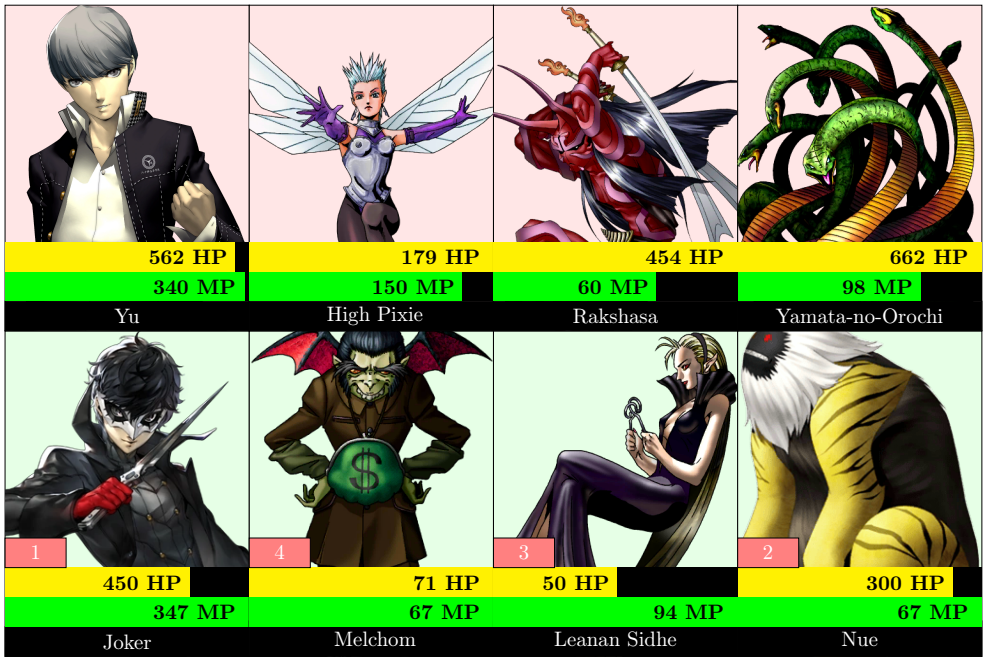


Figura 11: Tablero con el orden del equipo de **Joker** luego de que inicia una nueva ronda del jugador 1

6. Combate

Ahora que conocemos los distintos elementos del juego y como estos afectan el flujo del juego, revisaremos más en detalle los aspectos más especifico del combate.

6.1. Cálculo del daño

El daño que realizará una unidad será calculado en base a sus stats y de que forma se realice el ataque. Este daño es un número que se le resta al HP de la unidad que es atacada, Adicionalmente, otros elementos del juego pueden afectar el daño que realizará una unidad. Primero analicemos el caso más simple.

El stat de la unidad que se utilizará para los cálculos dependerá del tipo de daño del ataque. En particular, el stat se seleccionará de la siguiente manera:

- Si el ataque es de tipo **Phys**, entonces se utilizará la Str de la unidad
- Si el ataque es de tipo **Gun**, entonce se utilizará la Skl de la unidad
- Si el ataque es mágico, es decir, es de tipo **Fire**, **Ice**, **Elec**, **Force** o **Almighty**, entonces se utilizará la Mag de la unidad.

Si es que el ataque se realiza con las acciones atacar o disparar, entonces el daño se calculará con la siguiente formula:

$$Daño = [Stat] \cdot [Modificador] \cdot 0{,}0114$$

Donde $[Modificador]$ tomará un valor de 54 para la acción de atacar y un valor de 80 para la acción de disparar.

Por otro lado, si el ataque es realizado mediante el uso de una habilidad, entonces el daño se calculará como:

$$Daño = \sqrt{[Stat] \cdot [SkillPower]}$$

Donde $[SkillPower]$ es un valor propio de cada habilidad que será explicado más adelante.

El daño nunca podrá ser negativo, teniendo un valor mínimo de 0.

Veamos esto con un ejemplo. Tomemos al samurai **Nahobino**, quien tiene los siguientes stats:



(a) Nahobino de Shin Megami Tensei V

- **HP**: $HP = 453$
- **MP**: $MP = 389$
- **Str**: $Str = 48$
- **Skl**: $Skl = 92$
- **Mag**: $Mag = 111$
- **Spd**: $Spd = 52$
- **Lck**: $Lck = 55$

Si es que el jugador escoge la acción Atacar para **Nahobino**, dado que esta acción hace daño **Phys**, se seleccionará el Str de **Nahobino**, que tiene un valor de 48. Luego el daño se calculará como:

$$Daño = 48 \cdot 54 \cdot 0{,}0114 \approx 29$$

Si el jugador escoge la acción Disparar para **Nahobino**, dado que esta acción hace daño **Gun**, se seleccionará el Skl de la unidad, que tiene un valor de 92. Luego el daño se calculará como:

$$Daño = 92 \cdot 80 \cdot 0{,}0114 \approx 83$$

Por último, asumamos que el jugador selecciona la habilidad **Agi** para que **Nahobino** use. **Agi** tiene hace daño del elemento **Fire** y tiene un Skill Power de 80. Si **Nahobino** usa esa habilidad, se selecionará la Mag de la unidad, la que tiene un valor de 111. Entonces el calculo el daño se calculará como

$$Daño = \sqrt{111 \cdot 80} \approx 94$$

6.2. Efecto de las afinidades en el calculo del daño

Las afinidades no solo afectan cuantos turnos consumirán los ataques, sino que también afectarán cuanto daño recibirán las unidades. En particular, las afinidades tienen los siguientes efectos en el daño recibido:

- **Neutral**: Si la unidad tiene esta afinidad al elemento del ataque, no recibirá ni más ni menos daño.
- **Resist**: Si la unidad tiene esta afinidad al elemento del ataque, el daño que reciba se multiplicará por un factor de 0,5.
- **Weak**: Si tiene esta afinidad, el daño recibido se multiplicará por un factor de 1,5
- **Null**: Si tiene esta afinidad, el daño recibido será nulificado y no afectará a la unidad

- **Repel**: Si tiene esta afinidad, la unidad devolverá el daño que se le hubiese realizado al rival que la atacó. Este rival recibirá ese daño ignorando cualquier afinidad que esta unidad pueda tener.
- **Drain**: Si tiene esta afinidad al elemento del ataque, en vez de recibir daño, la unidad curará una cantidad de HP igual al daño que hubiese realizado el ataque.

6.3. Manejo de decimales

Para facilitar el testeo automático del juego, evitaremos usar números decimales al momento de realizar cálculos. En particular los distintos cálculos del juego producen resultados decimales, los cuales **deben ser truncados** al entero más cercano por debajo. El valor que será truncado será el resultado final del calculo del daño luego de haberle aplicado cualquier tipo de modificador de daño.

Por ejemplo, supongamos que una unidad usa una habilidad con $SkillPower = X$, que esta tiene $Mag = Y$ y que su rival tiene afinidad que multiplica el daño por un valor Z, entonces el resultado será:

$$Daño = \lfloor \sqrt{X \cdot Y} \cdot Z \rfloor$$

Ahora supongamos que además de lo anterior, el rival puede reducir el daño multiplicándolo por un factor W, entonces el resultado final será:

$$Daño = \lfloor \sqrt{X \cdot Y} \cdot Z \cdot W \rfloor$$

En resumen, se truncará el valor final del daño luego de aplicar cualquier tipo de modificador.

7. Habilidades

Las habilidades tienen la capacidad de ayudar al jugador a explotar debilidades de sus rivales, potenciar a sus unidades o perjudicar al equipo del rival. Todas las habilidades tienen distintas caracteriticas que determinan su efectividad en el juego. Estas son:



(a) Jack Ripper de Shin Megami Tensei IV

- **Nombre**: Es el nombre de la habilidad, funciona como identificador único.
- **Tipo**: Determina que tipo de efecto la habilidad tendrá en el juego. Los distintos tipo se explicarán más adelante.
- **Costo**: Las habilidades tienen un costo de MP asociado que deberá pagarse al ser usada. Una unidad no podrá utilizar una habilidad si no tiene suficiente MP para pagar su costo. El costo de MP se consume antes de aplicar el efecto de la habilidad.
- **Skill Power**: Es un valor que determina la potencia y efectividad de algunas habilidades.
- **Objetivo**: Determina sobre quien o quienes tendrá efectividad la habilidad.
- **Hits**: Determina cuantas veces se aplicará el efecto de la habilidad al ser utilizada
- **Efectos**: Es una descripción de la habilidad. Algunas habilidades pueden tener efectos secundarios detallados en esta descripción.

7.1. Hits

La gran mayoría de las habilidades cuenta con $hits = 1$ o algún otro número entero, sin embargo, hay habilidades que rompen esta regla. Estas habilidades cuentan con un conjunto de valores que puede tomar el atributo *hits*. Para entender esto veamos dos ejemplos esto con dos ejemplo:

- El atributo *hits* de **Fire Breath** tiene un valor de $[1, 4]$, lo que quiere decir que la habilidad, al ser utilizada, realizará daño mágico de fuego X veces, donde X es un número entero entre 1 y 4.
- El atributo *hits* de **Bouncing Claw** tiene un valor de $[1, 3]$, lo que quiere decir que la habilidad realizará daño físico X veces, donde X es un número entero entre 1 y 3

El valor de este número X para una habilidad con $hits = [A, B]$ se determinará con el siguiente algoritmo:

- Para cada jugador, definiremos un número entero al que llamaremos K, el cual tendrá un valor de 0 al inicio del juego. Cada vez que el jugador decida utilizar una habilidad (independiente de su tipo) el valor de K aumentará en 1 luego de que se apliquen los efectos de la habilidad.
- Definiremos un número $offset$ como $offset = K \mod (B - A + 1)$
- Finalmente, el número de veces que se aplicará el efecto de la habilidad será $hits = A + offset$

Veamos un ejemplo usando la habilidad **Myriad Arrows**, que cuenta con $hits = [2, 4]$. Supongamos que el jugador ha usado 5 habilidades durante el juego, entonces el número de veces que se dañará con la habilidad se determinará de la siguiente manera:

- Definiremos $K = 5$ ya que el jugador ha usado 5 habilidades.
- Definiremos $offset = 5 \mod (4 - 2 + 1) = 5 \mod 3 = 2$
- Finalmente, la habilidad realizará daño $2 + 2 = 4$ veces.
- Luego de que la habilidad haga daño, K terminará con un valor de $K = 6$

7.2. Objetivos

Como discutimos, el objetivo determinará quiénes se verán afectados cuando una unidad utilice cierta habilidad. Existen 8 tipos de objetivos:

- **Single**: La habilidad afectará a una unidad de los puestos activos del rival. Esta será escogida por el jugador.
- **All**: La habilidad afecta a todas las unidades de los puestos activos del rival.
- **Multi**: La habilidad afecta a algunas de las unidades de los puestos activos del rival. Quienes se ven afectadas dependerá de un algoritmo que se explicará a continuación.
- **Ally**: La habilidad afectará a un aliado del tablero escogido por el jugador. Si la habilidad tienen la capacidad de revivir a un aliado, entones el jugador podrá escoger a aliados de la reserva.
- **Party**: La habilidad afectará a todas las unidades en los puestos activos del jugador, incluyendo a quien utilizó la habilidad. Si la habilidad tiene la capacidad de revivir a los aliados, entonces afectará a todo el equipo activo y en reserva del jugador.
- **Self**: La habilidad afectará a la unidad que la utilizó.
- **Universal**: La habilidad afectará a todas las unidades del tablero, sean aliados o rivales.

El algoritmo para determinar a que unidades afectarán las habilidades con objetivo **Multi** es bastante similar al algoritmo para *hits* que revisamos la sección anterior. El algoritmo es el siguiente:

- Para cada jugador, definiremos un número entero K de la misma forma que en el algoritmo anterior, es decir, cada vez que el jugador decida utilizar una habilidad (independiente de su tipo) el valor de K, que se inicializa como 0 al inicio del juego, aumentará su valor en una unidad. Este valor aumentará luego de que se apliquen los efectos de la habilidad.
- Definiremos A como la cantidad de unidades vivas en los puestos activos del rival.
- Definiremos un valor $i = K \mod A$
- Definiremos la dirección D. Esta dirección será derecha si i es un número par, en caso contrario será izquierda.
- Definiremos R como el conjunto de unidades vivas en los puestos activos del rival ordenadas de izquierda a derecha.
- Como punto de partida seleccionaremos el puesto en la posición i del conjunto R ($R[i]$), donde el puesto en la posición 0 es el de más a la izquierda y el de la posición $A - 1$ es el de más a la derecha.
- Desde el punto de partida seleccionado, nos moveremos en la dirección D una cantidad de veces igual a $hits - 1$ entre los elementos de R. Si es que en algún momento llegamos a los limites del conjunto R, entonces continuaremos moviéndonos por la misma dirección desde el extremo contrario.
- Finalmente, todas las unidades encontradas mientras recorríamos en conjunto R, incluyendo a la unidad del punto de partida, recibirán el efecto de la habilidad

Notar que es posible que una unidad sea seleccionada más de una vez. Si esto ocurre, entonces la unidad recibirá los efectos de la habilidad todas las veces que haya sido seleccionada.

Para que esto quede más claro, continuaremos desarrollando el ejemplo de la sección anterior. Consideremos que el jugador 1 ha usado 5 habilidades durante el juego y que ahora quiere que su samurai use **Myriad Arrows**, que como vimos anteriormente tendrá $hits = 4$. Adicionalmente, supongamos que el tablero se encuentra en el siguiente estado:



Figura 14: Tablero de ejemplo. El equipo rival tiene 3 unidades en sus puestos activos

Se determinará que unidades serán atacadas de la siguiente manera:

- Definiremos $K = 5$ ya que el jugador ha usado 5 habilidades
- Definiremos $A = 3$ ya que el rival tiene 3 unidades en el tablero.
- Definiremos $i = 5 \mod 3 = 2$
- Definiremos $D = derecha$ ya que i es un número par.
- Definiremos $R = [\textbf{Kotone}, \textbf{Black Rider}, \textbf{Matador}]$
- Seleccionaremos como punto de partida la posición $R[i] = R[2] = \textbf{Matador}$
- Nos moveremos $4 - 1 = 3$ veces hacia la derecha desde el puesto donde se encuentra **Matador**. De esta forma encontraremos a **Kotone**, luego a **Black Rider** y, finalmente, a **Matador**
- Finalmente, las unidades que encontramos con este algoritmo fueron **Matador**, **Kotone**, **Black Rider** y **Matador**, por lo que la habilidad dañará 2 veces a **Matador** y 1 vez tanto a **Kotone** como a **Black Rider**

7.3. Habilidades ofensivas

Las habilidades ofensivas son, principalmente, habilidades que le permiten a la unidas hacer daño. Usarlas suele ser más efectivo que utilizar las acciones de atacar o disparar, pero tienen la desventaja que la unidad debe consumir MP para poder utilizarlas y que los rivales pueden tener afinidades que les permitan resistirlas.

Un tipo de habilidades ofensivas que resaltan son las habilidades de tipo **Almighty**. Estas realizan daño mágico todo poderoso, tipo de ataque para el que todas las unidades tienen afinidad **Neutral**. Esto quiere decir que, bajo situaciones normales, estos ataques no serán particularmente efectivos contra ninguna unidad, pero no correrán el riesgo de afectar negativamente al usuario de la habilidad.

7.4. Habilidades que drenan HP o MP

Dentro de las habilidades **Almighty** existe una sub-categoría que tienen interacciones especiales en el juego. Estas habilidades no realizarán daño de forma normal, sino que le robarán HP o MP al rival para que el usuario de la habilidad pueda recuperar estos stats.

La cantidad de HP o MP que robará la habilidad al rival se calculará con la formula de daño normal, sin embargo, dado que estas habilidades roban stats, el valor no podrá superar la cantidad de ese stat que la unidad tiene disponible al momento de recibir el ataque. Por ejemplo, si el ataque puede robar 100 MP, pero el rival solo tiene 5 MP, la habilidad robará esos 5 MP y se los recuperará al usuario de la habilidad.

7.5. Habilidades Light y Dark

Las habilidades tipo **Light** y **Dark** son habilidades ofensivas que no hacen daño, sino que tienen la capacidad de, bajo ciertas condiciones, matar instantáneamente al rival, independiente de su HP actual.

La efectividad de estas habilidades dependerá de la afinidad que tenga el rival sobre el elemento asociado. La siguiente tabla muestra esta relación:

Afinidad	Efectividad
Neutral	La habilidad matará al rival si $Lck_{Usuario} + SkillPower \geq Lck_{Rival}$. Si no se cumple la condición, el ataque hará **Miss**
Weak	La habilidad siempre matará al rival
Resist	La habilidad matará al rival si $Lck_{Usuario} + SkillPower \geq 2 \cdot Lck_{Rival}$. Si no se cumple la condición, el ataque hará **Miss**
Null	La habilidad siempre será bloqueada
Repel	La habilidad siempre matará a su usuario. Esto solo es posible con ciertas habilidades
Drain	No existen casos donde una unidad pueda drenar ataques de tipo **Light** o **Dark**

Cuadro 3: Efectividad según la afinidad del rival para habilidades **Light** y **Dark**

Si un ataque hace **Miss**, esto quiere decir que el ataque fallará y el rival no se verá afectado por el.

7.6. Habilidades Support

Las habilidades **Support**, como indica su nombre, son habilidades que no realizan ataques, sino que aplican efectos que pueden beneficiar al jugador o perjudicar al rival.

Existen múltiples tipos de habilidades **Support**, que pueden aplicar distintos efectos en el juego. En particular, los posibles efectos que estas habilidades pueden aplicar son los siguientes:

- **Charge**: Incrementa el daño del siguiente ataque físico de la unidad en un factor de 2,5. Los ataques físicos son aquellos con elemento **Phys** o **Gun**. No se puede acumular, es decir, si una unidad utiliza dos veces seguida esta habilidad, el incremento no será por un factor de $2{,}5 \cdot 2{,}5$ o 5,0, sigue manteniéndose en 2,5. Si una unidad usa una habilidad de ataque físico con más de un objetivo, se incrementa el daño solo del primer ataque.
- **Concentrate**: Incrementa el daño del siguiente ataque mágico de la unidad en un factor de 2,5. Los ataques mágicos son aquellos con elemento **Fire**, **Ice**, **Elec**, **Force** y **Almighty**, incluyendo los ataques que drenan HP o MP. No se puede acumular
- **Tetraja**: Previene un ataque de tipo **Light** y **Dark**, pero solo si este hubiese matado a la unidad. Este efecto no bloqueará el ataque, sino que hará que el ataque falle. No se puede acumular
- **Tetrakarn**: Repele ataques físicos, es decir **Phys** y **Gun**, por un turno. Tiene el mismo efecto en los turnos que la afinidad Repel. No se puede acumular
- **Makarakarn**: Repele ataques mágicos, es decir **Fire**, **Ice**, **Elec**, **Force**, **Light**, **Dark** y **Almighty**, por un turno. Tiene el mismo efecto en los turnos que la afinidad Repel. No se puede acumular. No afecta a habilidades que drenan HP o MP.
- **Tetrakowas**: Cancela los efectos de **Tetrakarn** en al unidad afectada
- **Makarakowas**: Cancela los efectos de **Makarakarn** en al unidad afectada
- **Doping**: Aumenta el HP máximo, pero no el actual, de la unidad afectada en 30 % hasta que esta muere. No se puede acumular.

Aparte de estos efectos, existe efectos que alteran los cálculos del daño, aumentando o disminuyendo la ofensiva o defensa de una unidad. Existen 2 tipos de buffs y 2 tipos de debuffs, los cuales afectarán unos valores propios de cada unidad llamados grado ofensivo y grado defensivo. Primero, revisemos estos buffs y debuffs:

- **Tarukaja**: Aumenta el ataque de la unidad, es decir, aumenta su grado ofensivo en 1

- **Tarunda**: Disminuye el ataque de la unidad, es decir, disminuye su grado ofensivo en 1
- **Rakukaja**: Aumenta la defensa de la unidad, es decir, aumenta su grado defensivo en 1
- **Rakunda**: Disminuye la defensa de la unidad, es decir, disminuye su grado defensivo en 1

Los grados ofensivos y defensivos son valores enteros que siempre empiezan en 0 y pueden existir en un rango de $[-3, 3]$. El grado ofensivo de una unidad hará que el daño que inflija sobre sus rivales sea multiplicado por un factor dado por la siguiente tabla:

Grado ofensivo	Multiplicador
-3	0.625
-2	0.75
-1	0.875
0	1
1	1.25
2	1.5
3	1.75

Cuadro 4: Relación entre el grado ofensivo y el multiplicador de daño

El grado defensivo de una unidad hará que el daño que esta reciba por ataques del rival sea multiplicado por un factor dado por la siguiente tabla:

Grado defensivo	Multiplicador
-3	1.75
-2	1.5
-1	1.25
0	1
1	0.875
2	0.75
3	0.625

Cuadro 5: Relación entre el grado defensivo y el multiplicador de daño

Los multiplicadores de ambos tipos no tendrán efecto en los cálculos del daño si es que el rival tiene afinidad **Drain** o **Repel** sobre el ataque o si el ataque drena el HP o MP del rival.

Finalmente, existen otros 2 efectos que pueden eliminar buffs o debuffs de las unidades:

- **Dekaja**: Elimina los buffs de la unidad, es decir, convierte en 0 el grado ofensivo o defensivo de la unidad si es que estos tenían un valor positivo
- **Dekunda**: Elimina los debuffs de la unidad, es decir, convierte en 0 el grado ofensivo o defensivo de la unidad si es que estos tenían un valor negativo

7.7. Habilidades Heal

Las habilidades **Heal**, como indica su nombre, son habilidades que pueden recuperar el HP a alguna unidad. Estas habilidades pueden tener los siguientes efectos:

- **Dia**: Cura una cantidad de HP igual a un X % de su HP máximo, donde X = [Skill Power]. Este efecto solo afecta a unidades vivas.

- **Recarm**: Revive a la unidad afectada con una cantidad de HP especificada en los efectos de la habilidad. No tiene efecto sobre unidades vivas. No puede devolver a monstruos de la reserva a no ser que lo especifique el efecto de la habilidad.

7.8. Habilidades Passive

Las habilidades **Passive**, como indica su nombre, son habilidades que el jugador no podrá decidir utilizar, sino que estas tomarán efecto automáticamente y sin injerencia del usuario. Estas habilidades pueden tener los siguientes efectos:

- **Null**: Otorga la afinidad **Null** a la unidad para cierto tipo de habilidad especificado en los efectos de la habilidad. Esto sobrescribe la afinidad original
- **Resist**: Otorga la afinidad **Resist** a la unidad para cierto tipo de habilidad especificado en los efectos de la habilidad. Esto sobrescribe la afinidad original
- **Repel**: Otorga la afinidad **Repel** a la unidad para cierto tipo de habilidad especificado en los efectos de la habilidad. Esto sobrescribe la afinidad original
- **Drain**: Otorga la afinidad **Drain** a la unidad para cierto tipo de habilidad especificado en los efectos de la habilidad. Esto sobrescribe la afinidad original
- **Lesson**: Otorga un bonus a algún stat de la unidad. Tanto el stat como el valor están especificados en el efecto de la habilidad. Este bonus afecta todos los valores y comparaciones del juego que usen este stat. Se puede acumular
- **Gain**: Otorga un bonus al HP o el MP máximos de la unidad por un porcentaje descrito en el efecto de la habilidad. No se ve afectado por efectos de tipo **Lesson**.
- **Pleroma**: La unidad realizara un 25 % más de daño al utilizar un ataque del elemento señalado en el efecto de la habilidad. Se acumula de forma aditiva.
- **Counter**: Si la unidad recibe un ataque **Phys** o **Gun** y sobrevive, entonces podrá contraatacar inmediatamente al rival apenas este termine de atacar. El daño que realizará será de un $[SkillPower]\,\%$ del daño recibido. No tendrá efecto en los turnos. No se acumula. Solo aplica la habilidad con mayor Skill Power.
- **Ally Counter**: Si un aliado de la unidad recibe un ataque **Phys** o **Gun** y sobrevive, entonces podrá contraatacar inmediatamente al rival apenas este termine de atacar. El daño que realizará será de un $[SkillPower]\,\%$ del daño recibido. No tendrá efecto en los turnos. No se acumula. Solo aplica la habilidad con mayor Skill Power. Si esta unidad muere, entonces no se activará el efecto.
- **Penetrate**: Al atacar con un elemento especificado en el efecto de la habilidad, la unidad ignorará las afinidades **Drain (Dr)**, **Null (Nu)** y/o **Resist (Rs)** del rival.

7.9. Habilidades Special

Las habilidades **Special** no caen dentro de alguna categoría anterior y no siguen ningún patrón que las haga similares entre si. Estas habilidades pueden tener efectos arbitrarios sobre el juego, los cuales pueden o no ser similares a otras mecánicas.

Las más importantes de estas habilidades son aquellas que permiten a los monstruos realizar acciones que normalmente solo puede hacer un samurai.

7.10. Efectos secundarios

Finalmente, en el juego existen habilidades ofensivas que tienen efectos secundarios, es decir, efectos adicionales aplicados sobre el rival. Algunos ejemplos de estas habilidades son:

- **Fang Breaker**: Un ataque que también aplica **Tarunda** al rival.
- **Cold World**: Un ataque que también lanza un ataque de muerte instantánea con $SkillPower = 10$ de tipo **Almighty**.

Para discutir los elementos especiales de estas habilidades, las separaremos en efecto principal y efectos secundarios, donde el efecto principal será el ataque correspondiente al tipo de la habilidad y los efectos secundarios serán todos los efectos adicionales de la habilidad. El como se aplicarán estos efectos dependerá de de cual sea la afinidad del rival al efecto principal de la habilidad. En la siguiente tabla podemos ver que pasará para cada afinidad que el rival puede tener sobre el ataque principal:

Afinidad	Efectividad
Neutral/ Weak/ Resist	La unidad recibirá el daño del efecto principal de forma normal. Lo que suceda con los efectos secundarios dependerá de cual sea la afinidad de la unidad sobre estos. Pudiendo ser reflejados, nulificados o absorbidos. El efecto que esto tendrá en los turnos será el que tenga prioridad.
Null	La unidad nulificará tanto el efecto principal como todos los efectos secundarios
Repel	La unidad devolverá tanto el efecto principal como todos los efectos secundarios
Drain	La unidad absorberá el daño del efecto principal Se nulificarán todos los efectos secundarios

Cuadro 6: Relación entre afinidad al efecto principal y lo que sucederá con el efecto secundario en habilidades con efectos secundarios.