Anotaciones sobre el pia (tcg):

Pequeña descripcion de lo que hará cada clase: (IDEA AL INICIAR EL PROYECTO ESTO PUEDE CAMBIAR CON EL TIEMPO)

Clase cartas: Se van a crear una clase abstracta(carta) con atributo de nombre, descripción y costo. De la clase madre se creara dos clase hijas que serán las cartas de personaje y de acción. Con sus respectivos métodos las cuales serian sus 3 habilidades y las mecánicas de perder o ganar vida.

Clase de masos: Tendrá el listado del maso y tendrá tambien métodos básico de extraer tarjeta, añadir tarjeta, barajear.

Clase jugador : Genera el maso, genera una lista aleatoria de las fichas de los elementos, poner lógica lógica de acciones que el hace (cambiar personaje, usar fichas)

Clase juego: va hacer el inicializador de todo y controlara todo el juego (controlar el juego es generar las fichas de cada jugador, terminar rondas).

Clase dados: Esto es para crear los dados elementales para poder realizar todas las acciones.

Descripción avances 18 de Abril del 2025

Se creo una clase abstracta llamada carta\_accion en el que tiene todos los datos genéricos de una carta de acción con su único método llamado aplicar efecto en el cual según sea la sbuclase de la carta de acción aplica un efecto

Se creo una subclase de carta acción llamda carta\_accion\_Apoyo esta clase sera para los objetos que aplican ventajas de tipo (recarga de energía, coldown, acciones rápidas, modificar datos, curacion etc…)

Carta\_Accion\_arma es una subclase de la clase abstracta antes mencionada en cual solo se aplica a los personajes que tienen un cierto tipo de arma en especifico esto su efecto es aumentar el ataque de las habilidades.

Clase maso es la clase que se encarga de gestionar todo lo necesita el jugador para poder jugar (crea su propio maso de personajes y cartas de apoyo) cuenta con funciones básica de modificar la baraja que tiene y la baraja de las carta que tiene disponibles. La función barejear cartas se usara al inicio de cada juego y al iniciar una ronda para darle cartas al jugador. Tambien maneja los dados (seleccionar aleatoriamente y los elimina)

Clase carta\_personaje es una clase que implementa otra case que son las habilidades de cada personaje. La clase cuenta con las atributos básicos de cada personaje para el juego (posiblemente se va a agregar mas atributos pero por el momento son esos). Las funciones de ganar y perder vida son para cuando reciben curación o daño respectivamente. Cuentan con el método ataque básico en el cual es un método igual independientemente del personaje solo si el personaje usa catalizador ahí cambio un poco no requiere mucho cambio. Lo diferente es en los método que hereda pero eso se va hablar en el siguiente párrafo.

La clase habilidades personajes sera la encargada de modificar los ataques según el personaje (cada personaje tiene ataques diferente que varia desde el cooldown, daño, o buffos especiales en algunos casos )

Tambien contamos dos enum es el cual nos sirvira par listar las armas y los elementos de cada personaje.

Se creo la clase dados en el cual tiene

Se creo la clase jugador en cual se selecciona los personajes a usar al igual que las clase de apoyo. Tambien uso el método para seleccionar el personaje que se va a usar y tambien las cartas

En la clase de juego se incializo los jugadores 1 y 2 y con el método iniciar ronda generamos los dados elementales, los personajes y las cartas de apoyo y de los de uso (están incompletos porque no hay objetos)

Se estuvo probando si funcionaba lo que se lleva hasta ahora, se termino creando objetos para la carta seleccionada. Por el momento se esta viendo como programar todo lo que conlleva terminar e iniciar una ronda (reiniciar dados, agregar 2 cartas de apoyo a la varaja)

Posible avances de mañana: Usar los ataques(aun no tendrán efecto en los jugadores) y las cartas (sin aplicar efectos)

La clase habilidades personajes sera la encargada de modificar los ataques según el personaje (cada personaje tiene ataques diferente que varia desde el cooldown, daño, o buffos especiales en algunos casos )

Tambien contamos dos enum es el cual nos sirvira par listar las armas y los elementos de cada personaje.

Personajes a usar:

Diluc

Bennet

Keqing

Sara

Furina

Hydro Hilichurl

Xianyun

Maguu Kenki

Chiori

Zhongli

Alhaitham

YaoYao

Ayaka

Rosaria

Listado habilidades

**Diluc** (Pyro)

* **Ataque Normal**: *Tempered Sword* – 2 PHY DMG.
* **Habilidad Elemental**: *Searing Onslaught* – 3 Pyro DMG, puede usarse hasta 3 veces en una ronda (coste incremental).
* **Habilidad Definitiva**: *Dawn* – 8 Pyro DMG, inflige *Pyro* al enemigo y crea *The Flame* (al final de la ronda, 2 Pyro DMG adicional).

**Bennett** (Pyro)

* **Ataque Normal**: *Strike of Fortune* – 2 PHY DMG.
* **Habilidad Elemental**: *Passion Overload* – 3 Pyro DMG, cura 1 HP al personaje activo si Bennett tiene ≤6 HP.
* **Habilidad Definitiva**: *Fantastic Voyage* – 4 Pyro DMG, crea *Inspiration Field* (+1 DMG para aliados, cura 1 HP por ronda).

**Keqing** (Electro)

* **Ataque Normal**: *Yunlai Swordsmanship* – 2 PHY DMG.
* **Habilidad Elemental**: *Stellar Restoration* – 3 Electro DMG, otorga *Lightning Stiletto* (al cambiar, Keqing no consume dados).
* **Habilidad Definitiva**: *Starward Sword* – 5 Electro DMG, inflige *Electro* y anula *Lightning Stiletto*.

**Kujou Sara** (Electro)

* **Ataque Normal**: *Tengu Bowmanship* – 2 PHY DMG.
* **Habilidad Elemental**: *Tengu Stormcall* – 3 Electro DMG, otorga *Crowfeather Cover* (+1 DMG al próximo ataque).
* **Habilidad Definitiva**: *Subjugation: Koukou Sendou* – 5 Electro DMG, inflige *Electro* y derriba al oponente.

**Furina** (Hydro)

* **Ataque Normal**: *Elegant Blade* – 2 PHY DMG.
* **Habilidad Elemental**: *Salon Solitaire* – Invoca *Singers of the Waters* (1 Hydro DMG por ronda, cura o daño según alineación).
* **Habilidad Definitiva**: *Let the People Rejoice* – 4 Hydro DMG, otorga *Universal Revelry* (aumenta DMG basado en cambios de HP).

**Hydro Hilichurl Rogue** (Hydro)

* **Ataque Normal**: Water Slash – 1 Hydro DMG.
* **Habilidad Elemental**: Hydro Summon – Invoca un Hydro Mimic (1 Hydro DMG por ronda).
* **Habilidad Definitiva**: **Torrential Ambush** – 3 Hydro DMG.
  + **Efecto adicional**: Si hay un Hydro Mimic en el campo, el daño aumenta a **5 Hydro DMG** y el Mimic se consume.

**Xianyun** (Anemo)

* **Ataque Normal**: *Whispering Wind* – 1 Anemo DMG.
* **Habilidad Elemental**: *Cloud-Grazing Strike* – 3 Anemo DMG, cura 1 HP a un aliado.
* **Habilidad Definitiva**: *Stars Gather at Dusk* – 4 Anemo DMG, reduce coste de habilidades aliadas.

**Maguu Kenki** (Anemo/Cryo)

* **Ataque Normal**: *Iaido Slash* – 2 PHY DMG.
* **Habilidad Elemental**: *Frosty Assault* – 3 Cryo DMG o *Gale Blade* (3 Anemo DMG).
* **Habilidad Definitiva**: *Kyouka Fuushi* – 5 Anemo/Cryo DMG (doble elemento).

**Chiori** (Geo)

* **Ataque Normal**: *Weaving Blade* – 2 PHY DMG.
* **Habilidad Elemental**: *Fluttering Hasode* – 3 Geo DMG, invoca *Tailor-Made* (escudo 1 HP).
* **Habilidad Definitiva**: *Hiyoku: Twin Blades* – 5 Geo DMG, rompe escudos fácilmente.

**Zhongli** (Geo)

* **Ataque Normal**: *Rain of Stone* – 2 PHY DMG.
* **Habilidad Elemental**: *Dominus Lapidis* – 3 Geo DMG, crea *Jade Shield* (reduce 1 DMG recibido).
* **Habilidad Definitiva**: *Planet Befall* – 5 Geo DMG, petrifica (salta turno enemigo).

**Alhaitham** (Dendro)

* **Ataque Normal**: *Abductive Reasoning* – 2 PHY DMG.
* **Habilidad Elemental**: *Universus: Normal Attack* – 3 Dendro DMG, genera *Mirror* (aumenta DMG).
* **Habilidad Definitiva**: *Fetters of Phenomena* – 5 Dendro DMG, consume *Mirrors* para bonus.

**Yaoyao** (Dendro)

* **Ataque Normal**: *Toss ‘n’ Turn Spear* – 1 Dendro DMG.
* **Habilidad Elemental**: *Raphanus Sky Cluster* – Invoca *Yuegui* (cura o daño Dendro).
* **Habilidad Definitiva**: *Moonjade Descent* – 4 Dendro DMG, cura equipo.

**Ayaka** (Cryo)

* **Ataque Normal**: *Kamisato Art: Kabuki* – 2 PHY DMG.
* **Habilidad Elemental**: *Kamisato Art: Hyouka* – 3 Cryo DMG, congela con Hydro.
* **Habilidad Definitiva**: *Kamisato Art: Soumetsu* – 5 Cryo DMG, persiste 2 rondas.

**Rosaria** (Cryo)

* **Ataque Normal**: *Spear of the Church* – 2 PHY DMG.
* **Habilidad Elemental**: *Ravaging Confession* – 3 Cryo DMG, reduce coste del próximo ataque.
* **Habilidad Definitiva**: *Rites of Termination* – 5 Cryo DMG, inflige *Cryo* persistent.

19/04/2025

Se realizo la acción de atacar ahora los personajes pueden atacar los personajes con sus diferentes habilidades sin embargo aun el calculo de daño solo es con su daño base y no con buffos como las armas o cartas. Tambien se modifico el resorteo de dados para que se sortearan otra vez los dados que no son de los elementos omni o el elementos de los personajes para que tengan mas dados utilizables.

Posibles avances de mañana: Hacer que los dados usen dados elementales y hacer que las cartas de acción tengan buffos.

05/05/25

Ya se hizo el calculo de daño con las armas equipadas, las cartas de armas ya se pueden aplicar, solo falta: buffos de cartas de apoyo(aumento daño, mas dados, mas cartas, reducción de costos), conteo de turnos para cada personaje para las definitivas. Poner buffos escudo, mitigación, recuperar vida( personalidar cada habilidad del personaje)

Cartas de apoyo

+Vida: Adeptus Temptation, Chef Mao, Leave It to Me, Minty Meat Rolls

+Ataque: Abyssal Summons, Ellin, Guardians Oath, Master of Weaponry

DadosM: Chang the Ninth, Jade Chamber, NRE, The Bestest Travel Companion

DadosAlzar: Blessing of the Divine Relic, Starsigns , Knights of Favonius Library

* DadosOMNI: Calxs Arts ,Favonius Cathedral, Liben, Paimon, Strategize

Armas:

* Espadas: Aquila Favonia, Sacrificial Sword, Travelers Handy Sword, Skyward Blade
* Mandoble: Sacrificial Greatsword, Wolfs Gravestone, Skyward Pride, The Bell
* Arco: Raven Bow, Sacrificial Bow, Skyward Harp, Amos' Bow
* Catalizador: Magic Guide, Sacrificial Fragments, Skyward Atlas, A Thousand Floating Dreams
* Lanza: Lithic Spear, Skyward Spine, White Tassel, Engulfing Lightning

List<Dados>costoCAR = new ArrayList();//Costo carta accion rapida  
Dados d1 = new Dados();//Aqui se seleccionara los dados que el usario quiero pero por mientras es una simulacion  
d1.setTipo(Elementos.PYRO);  
costoCAR.add(d1);//La cantidad de dados dependera de la carta  
Dados d2 = new Dados();  
d2.setTipo(Elementos.PYRO);  
costoCAR.add(d2);  
int cantD = juego1.jugdor1.maso.dadosJuego.size();  
juego1.jugdor1.maso.eliminarDados(costoCAR);  
  
if(cantD == juego1.jugdor1.maso.dadosJuego.size()+ costoCAR.size())  
{  
 if(juego1.jugdor1.getCaaApSelecionado() == null)  
 {  
 juego1.jugdor1.getCaaSelecionado().aplicarEfecto();  
 }else {  
 juego1.jugdor1.getCaaApSelecionado().aplicarEfecto();  
 }  
}