Trabajo Práctico

Juego de cartas intercambiables

[75.10] Técnicas de diseño

Alumno	Padrón
Apaldetti, Tomás	105157
Di Matteo, Carolina	103963
Andrade, Agustin	104046
Facundo Monpelat	92716
Santiago Tissoni	103856

Agregar cartas nuevas

- *Pez: [Criatura] 4HP, 1 Agua. Ataque: Quita 2HP*
- *Cese al fuego: [Reacción] Ante un ataque, ignorar todo daño y efecto de dicho ataque*
- *Vampiro: [Criatura] 5HP, 1 Planta. Ataque: Quita 3HP, cura a sí mismo por 1HP con un límite máximo de sus puntos de vida iniciales*

  Agregar una nueva clase para cada carta que llame al builder de cartas, en donde se le agreguen todas estas características. Para el caso de la reacción, se agrega un nuevo efecto en la parte de efectos y este se agrega desde el builder. Para la cura de vampiro ya existe el efecto 'Cura'; solo faltara agregar un ataque que agregue este mismo efecto a la 'OriginalAction' para en el momento de resolución, aplicarlo a la carta.

  Además, cada carta deberá agregarse a los enums de nombres de cartas y también a las opciones de creación en la factory

Agregar una nueva regla

- *Si un jugador tiene que tomar cartas de su mazo y no hay cartas suficientes, pierde inmediatamente*

  Para agregar esta nueva funcionalidad es necesario agregar una nueva regla de aviso de que una condición se cumplió en el jugador. El modo de juego deberá realizar la acción que considere apropiada para ese evento.

  Se deberá agregar un enum a la firma del *subscriber* para distinguir entre eventos de pérdida en caso de ser necesario.
- *Modo combinado que trackea puntos de victoria y vida (Mismas condiciones para generar cada uno que sus modos originales)*

  Crear una nueva clase que combina la funcionalidad de 'PointsCounter' y 'PlainHP', y a su vez implementa la interfaz de 'IMatchEndCondition'. Luego de desarrollar el funcionamiento entre ambas condiciones, se deberá pasar como argumento al momento de construir el player.

  Si es necesario identificar cuál condición se cumplió, se deberá agregar un Enum o alguna otra forma de distinguir los dos eventos.

Energía nueva

- *Metal*

  Agregar al enum de energía esta nueva.

  Agregar una carta artefacto para este tipo de energía.

Etapa nueva

- *Para simplificar la regla de no activar artefactos en el mismo turno que se invocan, queremos separar la etapa principal en dos:*
    - *Etapa de activación de artefactos (Solo activación de artefactos)*
    - *Etapa de invocación (Toda la etapa principal excepto activar artefactos)*

Agregar una nueva clase que implemente IPhase y sus respectivos enums para los parseos.

Trasladar la lógica de activación de artefactos a esta nueva clase, removiendolo de la actual. Todo los otros métodos deben lanzar una excepción.

Cambiar las clases para que los cambios de estados en el turno tengan sentido dentro del turno.

Recompensa

- *Al final de la partida, se le regala alguna cantidad de dinero a ambos jugadores.*

Una vez determinado el fin de la partida, dado que existe un ganador, cada jugador tiene una referencia al usuario subyacente. Utilizando esto, dado que un usuario tiene métodos de agregar dinero, se puede dar la recompensa; cada modo de juego o partida puede determinar la cantidad de dinero en base a ciertas estadísticas, o un monto plano.