

La creación de efectos de las cartas consiste en leer el código introducido por el usuario, ser interpretado correctamente y ajustado a estructuras de datos disponibles para su uso en las dependencias del juego.

El efecto debe ser escrito por el usuario en un archivo .txt perteneciente al proyecto, luego este será leído, convertido en una lista de string con cada palabra o token en una posición distinta de la lista, lo cual facilitará su análisis lexicográfico, análisis tales como comprobar si el código estará escrito correctamente.

Un ejemplo de un efecto creado en código sería:

```
"Effect { Target = Ally(1); Action = Heal(800); FaithCost=9;
MilitaryCost=12; CapitalCost=120; KnowledgeCost=85; }";
```

Acá el usuario deja explícito cada campo necesario del efecto en cuestión.

Este efecto se interpretará como:

Un aliado será curado con 800 puntos. El coste de Fe será de 9, el Militar de 12, el Capital de 120 y el de Conocimiento de 85.

Las propiedades FaithCost, MilitaryCost, CapitalCost, KnowledgeCost solo aceptarán valores numéricos, mientras que Target solo aceptará dos palabras: "Ally" o "Enemy" acompañadas de paréntesis con un valor numérico dentro, esto indica el tipo de objetivo: aliado o enemigo y la cantidad de objetivos; por otra parte Action acepta un conjunto de acciones tales como:

```
"Heal", "Atk", "DefUp", "AtkUp", "DefDown", "VitDown", "AtkDown"
```

Estas acciones también estarán acompañadas de un valor numérico entre paréntesis que indica la cantidad de puntos de acción que realizará el efecto.

Se pueden agregar fácilmente nuevas acciones al programa sin tener que hacer muchas modificaciones al código.

La clase estática LexicalAnalyzer contiene un conjunto de métodos que se utilizan para transformar el código introducido por el usuario para su posterior interpretación, así como tener métodos que comprueban si el código es correcto en escritura.

La clase estática ReservadeWords contiene información y colecciona todas las palabras reservadas del lenguaje.

La clase estática AuxiliarMethods contiene un conjunto de métodos auxiliares y necesarios para el funcionamiento de otros métodos en otras clases.

La clase ObjectEffect representa una estructura de datos coherente con la interpretación del código del efecto, cada propiedad de dicha clase es un campo necesario del efecto, y a esta se le agregan como propiedad un objeto del tipo SubjectWord el cual representa una acción inmediata algo así como: " Ally(1)Heal(800) ; " donde se interpreta de la siguiente forma: "Un aliado será curado con 800 puntos de vida". Las palabras de tipo acción como "Heal" crean un objeto de tipo "ActionWord" donde tiene como propiedad el nombre de la Acción y su cantidad. Luego es creado un objeto de tipo "SubjectWord" que recibe una ActionWord, una palabra de tipo sujeto como "Ally" y su cantidad. Se hace todo un proceso después para ejecutar estas acciones y modificar las instancias del tipo "Player".

Deben ser rellenas todos los campos de efecto para que se interprete correctamente, de lo contrario dará error.

Existen otras clases dentro del proyecto que no son mentadas, ya que son partes inacabadas de una diseño de lenguaje más robusto, muchas de estas ideas se están trabajando o vinculando al proyecto que ya tenemos.