

Trabajo Práctico Especial– Primera Parte – 2017

Grupos

Para la realización del trabajo práctico especial se deberán formar grupos de 2 personas. Aquellos que por diferentes razones no puedan formar un grupo y decidan realizar el trabajo práctico de forma individual, enviar un mail a la cátedra informando la situación.

Juego De Cartas

Defina las clases (nombre, superclase, atributos y métodos) para implementar una solución orientada a objetos para el siguiente problema e implemente en Java.

Un mazo de cartas consiste en una colección de cartas con iguales características. Cada carta consiste en un personaje y varios atributos numéricos que la definen. A continuación se muestra un ejemplo de 3 cartas pertenecientes a un mazo de cartas de Minions. En el ejemplo se puede ver cómo cada carta posee un nombre, y 3 atributos que lo definen: Valentía, Liderazgo y Ternura.



El siguiente ejemplo muestra un ejemplo de 3 cartas pertenecientes a un mazo de cartas de Transformers. En este ejemplo, los atributos que se definen en cada carta son 5: Altura, Peso, Fuerza, Peleas Ganadas y Velocidad.



Las cartas de un mazo siempre poseen la misma cantidad de atributos, y todas sus cartas deben poseer los mismos atributos. La cantidad de atributos puede oscilar entre un mínimo de 3 y un máximo de 6. Todos los atributos tienen un nombre y un valor numérico.

Se debe proveer funcionalidad que permita la **creación de mazos** y que un mazo de cartas pueda **verificar que todas sus cartas son correctas** (mismos atributos y cantidad). Adicionalmente, se debe proveer la funcionalidad para **jugar una partida de cartas**.

La mecánica del juego es la siguiente: Se reparten las cartas del mazo en dos montones iguales (si las cartas son impares, la última carta del mazo no se utiliza). Cada jugador toma la primera carta de su montón. El primer jugador selecciona un atributo de los disponibles. El jugador que posea la carta con mayor valor para ese atributo gana la ronda. El ganador de la ronda se lleva las dos cartas (las debe colocar al final de su montón) y también tiene el turno en la siguiente ronda (es decir, es quien selecciona el atributo a utilizar en la siguiente ronda).

En caso de que ambas cartas posean el mismo valor para el atributo seleccionado, ocurre un empate. En el caso de los empates cada jugador se queda con su carta, la cual va a parar al fondo de su montón. El turno lo tiene el jugador que tenía el turno anteriormente

Si uno de los jugadores se queda sin cartas, el juego termina y gana el otro jugador. También es posible jugar en la modalidad de Rondas, en dicho caso se configura el número de rondas y cuando se llega a ese número gana el jugador que tenga más cartas. Si los dos posee la misma cantidad se decreta un empate.

Alcance del práctico especial

Diseño de las clases involucradas con sus atributos y responsabilidades.

Implementar el juego por línea de comando (se crean los objetos de forma estática en el main). Para jugar, se crean jugadores que eligen de forma aleatoria el atributo por el cual se van a enfrentarse.