Implementación de Juego de Cartas “Guerra”

Introducción

Este informe actualiza la implementación del juego de cartas "Guerra", orientado a objetos y utilizando estructuras de datos más definidas. La nueva versión se enfoca en una representación más modular y una estructura de datos más sofisticada para manejar las cartas, los jugadores y las operaciones del juego.

Objetivos

Los objetivos del proyecto continúan siendo los mismos, pero con un enfoque más orientado a objetos:

1. **Representación Modular de las Cartas y el Mazo:** Se ha implementado la estructura de datos basada en clases como **Nodo** y **Mazo** para una representación más coherente y ordenada.
2. **Operaciones Mejor Estructuradas:** Se han redefinido las operaciones de repartición de cartas, comparación y resolución de conflictos (guerras) utilizando las nuevas estructuras de datos.
3. **Mantenimiento de la Aleatoriedad:** Se asegura la aleatoriedad del mazo mediante la utilización de algoritmos de mezcla en la creación del mazo.
4. **Manejo de Excepciones y Pruebas Unitarias:** La nueva implementación continúa garantizando la robustez mediante el manejo de excepciones y la realización de pruebas unitarias.

Operaciones Principales

Las operaciones clave del juego se mantienen, aunque redefinidas con las nuevas estructuras de datos:

- **Creación y Mezcla del Mazo de Cartas:** Se ha introducido la clase **Mazo** con operaciones de poner arriba, sacar arriba, poner abajo y obtener cabeza/cola para manipular el mazo de manera más eficiente.
- **Repartición de Cartas a los Jugadores:** Se ha modificado la lógica para repartir las cartas a los jugadores, aprovechando las funcionalidades del nuevo diseño.
- **Juego de Rondas:** Las operaciones de comparación de cartas, manejo de guerras y actualización de manos se han actualizado para adaptarse a la nueva estructura.

Complejidad Algorítmica

Complejidad Algorítmica

- **Creación y Mezcla del Mazo de Cartas**:
    - Creación del mazo: Tiempo O(1).
    - Mezcla del mazo: Tiempo O(n) en el peor caso (donde 'n' es el número de cartas en el mazo).

- **Repartición de Cartas a los Jugadores**:
    - Repartir cartas a cada jugador: Tiempo O(n) en el peor caso (donde 'n' es el número de cartas a repartir).
- **Juego de Rondas**:
    - Jugar ronda: Tiempo O(1) en el caso común, pero puede llegar a O(n) en el peor caso si se produce una guerra y es necesario mover múltiples cartas.
- **Obtener Mazo**:
    - Obtener la cabeza del mazo: Tiempo O(1).
- **Jugar**:
    - Jugar el juego en su totalidad: Tiempo O(n) en el peor caso (donde 'n' es el número máximo de turnos permitidos en la partida).

Resultados Exitosos

Los resultados exitosos se han mantenido con la nueva implementación:

- **Mantenimiento de Funcionalidad Central:** La reestructuración no afecta la funcionalidad central del juego, manteniendo la comparación de cartas, resolución de guerras y el final del juego.
- **Mejora en la Estructura y Mantenibilidad:** La nueva estructura de clases **Nodo** y **Mazo** proporciona una base más sólida y mantenible para la gestión de las cartas y los jugadores.
- **Mantenimiento del Manejo de Excepciones y Pruebas Unitarias:** Se mantiene el enfoque en la robustez y calidad del código con el manejo de excepciones y pruebas unitarias.

Conclusiones

La reestructuración del código ha permitido una representación más clara y modular del juego de cartas "Guerra", manteniendo las operaciones y la lógica esencial. La introducción de clases ha mejorado la estructura de datos y la mantención del juego, manteniendo el enfoque en el manejo de excepciones, pruebas unitarias y la eficiencia en la manipulación de las cartas.

Estos cambios han mejorado la mantención del código, ofreciendo una estructura más coherente y mejorada en comparación con la versión anterior.