

Risk: Entrega 2

Jorge Esteban Martinez Clavijo
Ingeniería de sistemas
Pontificia Universidad Javeriana
Bogotá, Colombia
Je.martinezc@javeriana.edu.co

I. DESARROLLO

Funcionalidades pendientes

TAD Carta, el TAD Carta ha cambiado su implementación con relación a la entrega pasada, ya que ahora la carta solo posee el nombre de un territorio y la tropa que posee, las cartas del juego se encuentran en un archivo.txt en el directorio raíz del proyecto, este archivo es leído y se crean las diferentes combinaciones de cartas. Al momento de asignar una carta a un jugador que ha conquistado un territorio, hay una función que dado un numero aleatorio, siempre dentro del rango de la cantidad de cartas, busca esa carta en el vector, la da al jugador y es eliminada del vector de cartas, para asegurarse que no hay cartas repetidas.

El comando Turno, Obtener unidades: En la segunda entrega se realizó la implementación del comando turno, que recibe un nombre de jugador y valida si es el jugador en turno o indica si a ese jugador no le corresponde ese turno o de lleno, el jugador no existe. Se validan 3 casos en los que un jugador obtiene unidades, haciendo uso de las siguientes funciones

Risk::nuevasTropasTerritorio(), que agrega un número de tropas con relación a la cantidad de territorios conquistados dividido entre 3, se aproxima a la cifra más cercana usando la función round(). Risk:: nuevasTropasContiente(), busca en cada continente, si todos sus territorios están gobernados por un jugador, si lo están, se agregan las fichas correspondientes a cada caso.

El comando de turno se implementó, inicialmente se le dan unidades adicionales al jugador en turno basados en primero, los territorios que ocupa, segundo los continentes que ocupa por completo y, por último, las combinaciones de cartas que puede hacer el jugador. También se cambió la forma en la que se pobla el tablero de juego con los continentes y sus territorios, donde se tiene un archivo aparte con los continentes, otro archivo con los territorios y otro archivo con los territorios vecinos, que indica los territorios colindantes a un territorio, esto con el fin de saber cuales territorios se pueden atacar.

Comando Guardar <nombreArchivo>

Comando que ejecuta una función que crea un archivo de texto con toda la información presente en el juego como lo es, jugadores, territorios con el total de fichas de cada jugador y las cartas asignadas, posee una estructura con serie de etiquetas que permite al compilador identificar el valor que debe ser ingresado al árbol, se divide en tres bloques, primer bloque indica todos los jugadores, segundo bloque indica los territorios con la cantidad de fichas de cada jugador y por ultimo las cartas de cada uno. las etiquetas usadas son las siguientes:

1. TRR: indica que a partir de ese punto empiezan los territorios de un jugador.
2. CAR: indica que a partir de ese punto empiezan las cartas del jugador, es importante, un jugador puede o no tener cartas.
3. FJ: indica el fin de un jugador y el comienzo de otro para poblar las estructuras.

La estructura usada para el archivo es la siguiente

1. Primer Bloque, Jugadores
 - a. Están en fila los nombres de los jugadores, por ejemplo, hay 3 jugadores en el juego, Carlos, Camilo y Julián, por ende se guarda de esta manera
 - i. Carlos
 - ii. Camilo
 - iii. Julián
2. Segundo bloque, territorios
 - a. Se encuentran dentro de etiquetas TRR que hacen la división de territorios para cada jugador, como se organizan en relación con el índice del jugador, el primer grupo después de TRR corresponde a los territorios del jugador Camilo y así sucesivamente.
3. Tercer bloque, cartas
 - a. Siguiendo la misma idea de que se encuentran por orden de índice, las cartas se asignan al jugador correspondiente al índice presente.

```

# partida.txt
1 jorge
2 esteban
3 martinez
4 TTR
5 alaska 22
6 estados_unidos_orientales 1
7 argentina 1
8 brasil 1
9 colombia 5
10 europa_del_norte 1
11 ucrania 1
12 africa_oriental 1
13 africa_del_norte 1
14 china 1
15 japon 1
16 mongolia 1
17 ural 1
18 indonesia 1
19 TTR
20 alberta 22
21 groenlandia 1

```

Diseño de TADs



miro

Ilustración 1 Diseño de Diagrama de TADs: Elaboración propia

Plan de pruebas comando guardar_comprimido

Objetivo

Verificar el funcionamiento correcto de cada uno de los procesos esenciales en el programa, revisar posibles fallos o problemas algorítmicos y validar cada uno de estos procesos.

Requisitos

El programa debe cumplir con los requisitos esenciales para poder ejecutar este comando

1. Debe haber una partida inicializada.
2. La cantidad de jugadores debe estar entre 3 y 6
3. Cada jugador debe tener un determinado numero de territorios con sus fichas
4. Deben existir las frecuencias de cada letra del archivo generado por la función guardar

Casos de prueba

Guardar_comprimido <nombrearchivo>

Resultado esperado

Al ejecutar el comando Guardar_comprimido <nombrearchivo> debería hacer uso de la función que crea el texto que se guarda en el archivo creado por el comando guardar, anteriormente descrito, dentro del juego, se toma esta cadena, se borran los saltos de línea “\n” para quedar en una cadena completa y se empieza a recorrer calculando la frecuencia de cada letra, luego, se empieza a formar el árbol usando el nodoHuffman, que se va relacionando con los demás nodos que conforman el árbol, luego se debe recorrer este árbol y según la ruta (derecha / izquierda) se va colocando un 1 o un 0 eso asegura que no hay combinaciones repetidas

Resultado obtenido

Se trató de implementar la función que ayuda a determinar la frecuencia de los caracteres dentro de la cadena, pero queda en un bucle infinito, revisar para la próxima entrega.

Casos de error

Hay una serie de eventos que conducen a un error dentro de la ejecución del comando

1. Cuando la partida no ha sido inicializada correctamente.
2. Cuando hay un error en la escritura del archivo
3. Cuando hay un error en la creación de la cadena con a la información del juego