



Estructura de Datos

Entrega 02 Proyecto RISK

Miguel Angel Rico Llanos

Mateo Maldonado Diaz

Edward Andres Quintero Palma

John Corredor

17/10/2023

1.1. Interfaz

El primer apartado del proyecto hace referencia a la interfaz, es decir al inicio de todo el código, al inicializar el código se obtiene por pantalla lo siguiente:

[illegible]

Este apartado recibe por consola la palabra 'ayuda' lo cual muestra el siguiente menú.

Comando	Descripcion
inicializar	Realiza las operaciones necesarias para inicializar el juego
turno	Realiza las operaciones descritas dentro del turno de un jugador. Por ejemplo, si se desea acceder al turno del jugador 5, se debe digitar de esta manera: turno.
guardar <nombre_archivo>	Guarda el estado actual del juego en un archivo de texto. Por ejemplo, si se desea guardar la partida bajo el nombre de 'juego', se debe digitar el comando de esta manera: guardar juego.
guardar_comprimido <nombre_archivo>	Guarda el estado actual del juego en un archivo binario comprimido. Por ejemplo, si se desea guardar la partida bajo el nombre de 'juego', se debe digitar el comando de esta manera: guardar_comprimido juego.
inicializar <nombre_archivo>	Inicializa el juego con los datos contenidos en el archivo identificado por <nombre_archivo>. Por ejemplo, si se desea leer el archivo denominado 'juego' para continuar con la partida guardada, se debe digitar: inicializar juego.
costo_conquista <territorio>	Calcula el costo y la secuencia de territorios a ser conquistados para lograr controlar el territorio dado por el usuario. Por ejemplo, para el territorio 'colombia': costo_conquista colombia.
conquista_mas_barata	Calcula el territorio que implique un menor número de unidades de ejército perdidas.
ayuda	Muestra la lista de comandos disponibles.
salir	Sale del juego Risk.

En este menú se muestran todos los comandos y operaciones que realizará el programa. Este menú recibe por consola el nombre del comando a realizar, y posteriormente se reciben y muestran distintos datos y menús dependiendo el comando seleccionado, a continuación se hará la explicación de una serie de comandos.

1.2.Comando Inicializar

Este comando realiza las operaciones necesarias para inicializar el juego. Inicialmente, el comando pregunta la cantidad de jugadores, para cada jugador su nombre, y luego, por turnos, preguntar a cada jugador en qué territorio desea ubicar sus unidades de ejército. En cada turno, el jugador sólo puede indicar un único territorio a ocupar.

```
$inicializar
Digite la cantidad de jugadores
1
Digite el nombre del jugador No.1
Miguel
Miguel, jugador No.1, Digita el continente en el cual deseas poner tus unidades de inicio.(id del continente)
+-----+
| 1.America del Norte |
| 2.America del sur  |
| 3.Europa            |
| 4.Africa            |
| 5.Asia              |
| 6.Australia         |
+-----+
3
Ahora digita el pais en el cual deseas poner tus unidades de inicio.(id del pais)
+-----+
| 1.Gran Bretaña      |
| 2.Islandia          |
| 3.Europa del Norte  |
| 4.Escandinavia      |
| 5.Europa del Sur    |
| 6.Ucrania           |
| 7.Europa Occidental |
+-----+
1
Miguel, Jugador No.1, con pais de inicio Gran Bretaña
El juego se ha inicializado correctamente.
```

Como se puede evidenciar, el comando inicializar muestra en pantalla “Digite la cantidad de jugadores”, esto recibe por consola un entero, posteriormente se pide que se ingresen los nombres de los jugadores, esto recibe un string (Una cadena de caracteres), continuando con el comando, le solicita a cada jugador que ingrese el id del continente que desea escoger para inicializar su juego y posteriormente el pais que desea escoger en dicho continente, estos dos apartados reciben por pantalla un entero, para acabar con el comando se finaliza con un mensaje recapitulando la información recibida y arroja un mensaje de éxito de que el juego se inicializo correctamente.

Es decir, al final la salida de este comando es un valor booleano que indica si la inicialización del juego se realizó con éxito o no. Retorna `true` si la inicialización se realiza correctamente, y `false` si hay algún error en el proceso.

1.3.Comando Turno

Este comando, se encarga de realiza las operaciones de un jugador (obtener nuevas unidades, atacar y fortificar). De esta forma, el comando informa al jugador cuántas unidades adicionales puede reclamar, para luego preguntarle en cuáles de sus territorios las quiere asignar y en qué cantidad. A continuación, pregunta la configuración del ataque, desde cuál territorio y hacia cuál territorio, verificando las condiciones ya descritas. Luego informa los valores obtenidos con los dados, y la cantidad de unidades que se ganan o pierden. Este proceso se repite hasta que alguno de los dos territorios se quede sin unidades, o hasta que el atacante decida detenerse. Finalmente, el comando pregunta al jugador los territorios vecinos

que desea seleccionar para la fortificación, así como la cantidad de unidades que se trasladarán de uno al otro.

```
$turno
Digite el id del jugador
1
Jugador No.1, tienes un total de 1 territorios en tu poder, por lo tanto puedes reclamar un total de 0 unidades
países en posesion: Argentina
```

Al ingresar el comando “turno” solicita el id del jugador, por lo tanto, recibe un entero por consola, posteriormente muestra la información de cuantos territorios tiene en su poder el jugador y cuantas unidades puede reclamar.

1.4.Comando Salir

Este último comando simplemente se encarga de termina la ejecución de la aplicación.

```
$salir
Gracias por usar nuestro programa
```

Este comando no recibe nada por consola y simplemente muestra un mensaje por pantalla agradeciendo por usar el programa.

2. Descripción de TADs.

- **TAD Jugador:**

Atributos:

Nombre: String

ID: Entero

País: String

CantPais: Entero

Países: Lista de Strings

Operaciones:

ObtenerNombre():

ObtenerID():

ObtenerPais():

ObtenerCantPais():

ObtenerPaises():

FijarNombre(n_dato):

FijarID(n_dato):

FijarPais(n_dato):

FijarCantPais(n_dato):

FijarPaises(n_dato):

Jugador		
Nombre: String		
ID: Entero		
Pais: String		
CantPais: Entero		
Países: Lista de Strings		
ObtenerNombre():		
ObtenerID():		
ObtenerPais():		
ObtenerCantPais():		
ObtenerPaises():		
FijarNombre(n_dato)		
FijarID(n_dato)		
FijarPais(n_dato)		
FijarCantPais(n_dato)		
FijarPaises(n_dato)		

- TAD País:

Atributos:

Nombre: String

PaísesVecinos: Lista de Strings

Operaciones:

ObtenerNombre():

ObtenerPaísesVecinos():

FijarNombre(n_dato):

FijarPaísesVecinos(n_dato):

Pais	
Nombre: String	
PaísesVecinos: Lista de Strings	
ObtenerNombre():	
ObtenerPaísesVecinos():	
FijarNombre(n_dato):	
FijarPaísesVecinos(n_dato):	

- Fortificar:

Atributos:

TerritorioX: String

Unidades: Entero

Operaciones:

ObtenerTerritorioX():

ObtenerUnidades():

FijarTerritorioX(n_dato):

FijarUnidades(n_dato):

Fortificar	
TerritorioX: String	
Unidades: Entero	
ObtenerTerritorioX():	
ObtenerUnidades():	
FijarTerritorioX(n_dato):	
FijarUnidades(n_dato):	

- **TAD Territorio:**

Atributos:

nombre: Cadena

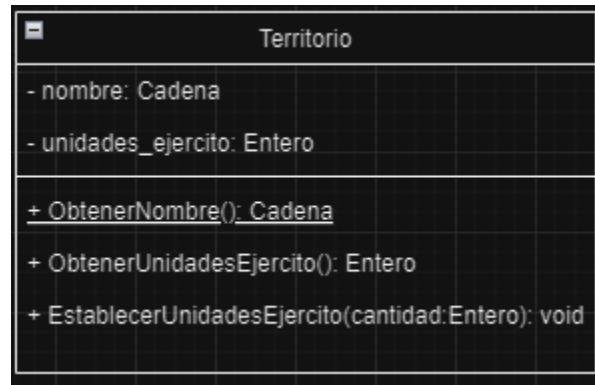
unidades_ejercito: Entero

Operaciones:

ObtenerNombre(): Cadena

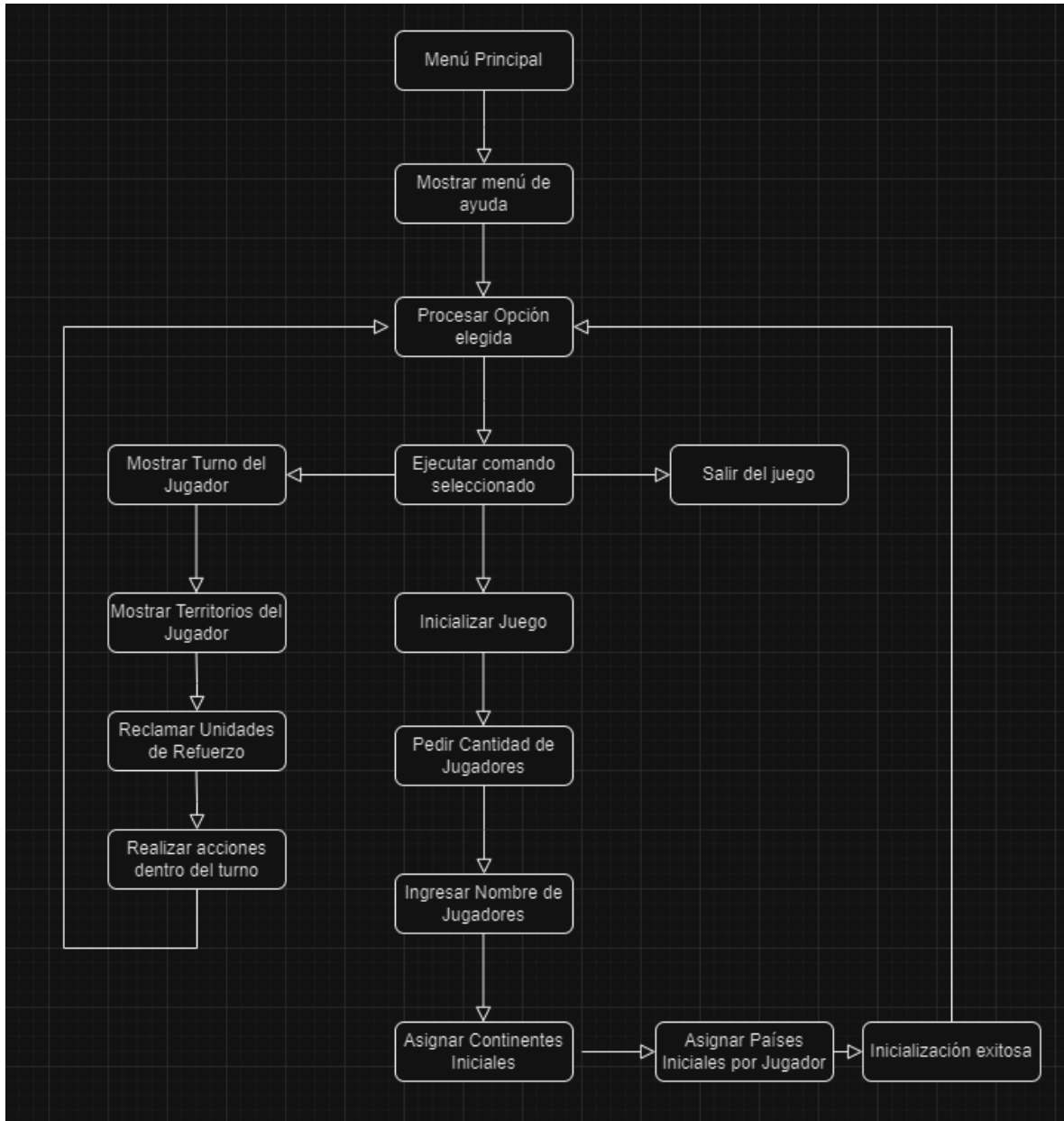
ObtenerUnidadesEjercito(): Entero

EstablecerUnidadesEjercito(cantidad:Entero): void



3. Funcionamiento General Comandos

A continuación, se muestra un diagrama el cual es bastante claro respecto al funcionamiento de los comandos, esta muestra un paso a paso de las acciones que se realizan durante la ejecución de cada operación.



4. Plan de Pruebas

A continuación, se adjuntarán imágenes del funcionamiento del comando “Iniciar”, se mostrarán todas las opciones disponibles en el momento y el resultado por pantalla.

Así se vería una correcta inicialización del juego

```
$inicializar
Digite la cantidad de jugadores
2
Digite el nombre del jugador No.1
Miguel
Digite el nombre del jugador No.2
Pepito
Miguel, jugador No.1, Digita el continente en el cual deseas poner tus unidades de inicio.(id del continente)
+-----+
| 1.America del Norte |
| 2.America del sur   |
| 3.Europa            |
| 4.Africa            |
| 5.Asia              |
| 6.Australia         |
+-----+
1
Ahora digita el pais en el cual deseas poner tus unidades de inicio.(id del pais)
+-----+
| 1.Alaska            |
| 2.Alberta           |
| 3.America Central   |
| 4.Estados Unidos Orientales |
| 5.Groenlandia       |
| 6.Territorio Noroccidental |
| 7.Ontario           |
| 8.Quebec            |
| 9.Estados Unidos Occidentales |
+-----+
1
Pepito, jugador No.2, Digita el continente en el cual deseas poner tus unidades de inicio.(id del continente)
+-----+
| 1.America del Norte |
| 2.America del sur   |
| 3.Europa            |
| 4.Africa            |
| 5.Asia              |
| 6.Australia         |
+-----+
2
Ahora digita el pais en el cual deseas poner tus unidades de inicio.(id del pais)
+-----+
| 1.Argentina         |
| 2.Brasil            |
| 3.Peru              |
| 4.Venezuela         |
+-----+
1
Miguel, Jugador No.1, con pais de inicio Alaska
Pepito, Jugador No.2, con pais de inicio Argentina
El juego se ha inicializado correctamente.
```

Si se ingresa una opción de continente no valida, muestra un mensaje de continente no valido y vuelve a mostrar el menú preguntando el continente nuevamente.

```
Miguel, jugador No.1, Digita el continente en el cual deseas poner tus unidades de inicio.(id del continente)
+-----+
| 1.America del Norte |
| 2.America del sur   |
| 3.Europa            |
| 4.Africa            |
| 5.Asia              |
| 6.Australia         |
+-----+
7
Continente no valido
Miguel, jugador No.1, Digita el continente en el cual deseas poner tus unidades de inicio.(id del continente)
```

Si se ingresa una opción de país no valida, muestra un mensaje de país no valido y vuelve a mostrar el menú preguntando el país nuevamente.

```
Ahora digita el pais en el cual deseas poner tus unidades de inicio.(id del pais)
+-----+
| 1.Afghanistan |
| 2.China        |
| 3.India        |
| 4.Irkutsk      |
| 5.Japon        |
| 6.Kamchatka    |
| 7.Medio Oriente|
| 8.Mongolia     |
| 9.Siam         |
| 10.Siberia     |
| 11.Ural        |
| 12.Yakutsk     |
+-----+
13
Pais no valido
```

Operación Guardar:

```
$guardar
Digite el nombre del archivo terminado en '.txt'
archivo.txt
La partida ha sido guardada correctamente.
$
```

```
archivo.txt

1  2
2  Pepe  1 Alberta,
3  Grillo  2 Escandinavia,
```

Error al querer guardar una partida que no ha sido inicializada:

```
$guardar
Esta partida no ha sido inicializada correctamente.
$
```

Carga correcta del archivo:

```
$inicializarA
Digite el nombre del archivo terminado en '.txt'
archivo.txt
Cantidad de jugadores: 2
Nombre: Pepe
ID: 1
Países: Alberta,

Nombre: Grillo
ID: 2
Países: Escandinavia,

$
```

Error al intentar iniciar un juego al momento de haber cargado otro:

```
$inicializarA
Digite el nombre del archivo terminado en '.txt'
asfsaf.txt
asfsaf.txt no contiene información válida para inicializar el juego.
$
```

Error al momento de intentar abrir un archivo que no existe:

```
$inicializarA
Digite el nombre del archivo terminado en '.txt'
asfsaf.txt
asfsaf.txt no contiene información válida para inicializar el juego.
$
```

No fue posible llevar a cabo la implementación de "guardar_comprimido" ya que no pudimos entender como implementar la codificación de Hauffman en código en c++, de igual manera se tendrá en cuenta para la tercera entrega.

ACTUALIZACIÓN DE LA ENTREGA 1:

Se corrigió el retorno booleano que presentaba fallas al momento de validar si la partida ya había sido inicializada, además de añadir las funciones correspondientes para la funcionalidad total de la función turno (Dichas funciones fueron aplicadas, pero aún no son implementadas)