

ESCUELA COLOMBIANA DE INGENIERIA

PROGRAMACIÓN ORIENTADA A OBJETOS

POOBConquista 2010-02

[Colaboración de Cesar Dueñas y Santiago Alzate]

DESCRIPCIÓN

El objetivo del proyecto es desarrollar una aplicación para que dos jugadores (un usuario y el computador) puedan jugar **POOBConquista**: un nuevo juego inspirado en el Risk. En **POOBConquista** los jugadores, que representan dos reinos, realizan sus movimientos por turnos y la conquista termina cuando un reino derrota a todos los ejércitos de su oponente o toma el territorio correspondiente a la base central enemiga.

Campo de juego

El campo de juego es un espacio rectangular de celdas que pueden estar ocupadas por un territorio, otro espacio rectangular, o estar libre. Los territorios, en esta versión, pueden ser de 4 diferentes tipos:

1. Territorios base: son la base central de los jugadores, están cerca a cada extremo del tablero (NE,SO) y tienen el símbolo correspondiente al reino. Cuando un grupo de combate retorna a su base se convierte en tantos grupos de combate como escuadrones tenga y recupera su fortaleza. La unión de escuadrones sólo es válida fuera de este territorio.
2. Territorios especiales: poseen un símbolo que los distingue en el tablero, no permiten la entrada de ejércitos que tengan una fortaleza menor a la exigida para su toma. Cuando un reino toma un territorio especial recibe escuadrones
3. Territorios normales: poseen un símbolo que los distingue y cualquier ejército puede intentar tomarlo.
4. Territorios montañosos: esos territorios no pueden ser atravesados ni ocupados por la artillería.
5. Territorios acuáticos: estos territorios no pueden ser ocupados ni atravesados por ninguno de los ejércitos actuales de este juego.

Reinos

Cada reino, para la conquista, cuenta con un ejército compuesto por grupos de combate. Cada grupo de combate está compuesto por escuadrones del mismo tipo. Los tipos de escuadrones pueden ser:

- Escuadrón de infantería el ejército más débil.
- Escuadrón de caballería que equivale a 5 infanterías.
- Escuadrón de **artillería** que equivale a 2 caballerías.

Inicialmente cada jugador recibe en su territorio base dos grupos de combate cada uno compuesto por una infantería. Adicionalmente, al comienzo de cada turno cada jugador recibe en su territorio base un grupo de combate compuesto por un escuadrón de infantería.

El reino dirigido por el computador puede ser de uno de los siguientes perfiles:

- ✓ Azaroso: decide sus actuaciones al azar
- ✓ Destructor: su principal interés es dar batalla, sin importar si pierda o gana.

Turnos y movimientos

El movimiento básico de un reino es avanzar un grupo de combate un número de posiciones en una dirección indicada (diagonal, vertical u horizontalmente).

Si al terminar el movimiento el grupo de combate queda en territorio enemigo, se declara una batalla. En este caso el reino victorioso es aquel cuyos ejércitos tenga una fortaleza mayor a la de los ejércitos que defienden quedando su fortaleza reducida considerando la fortaleza del enemigo.

Si al terminar el movimiento un grupo de combate queda en la misma posición de otro grupo de combate del mismo tipo, se arma un único grupo de combate con los escuadrones de los dos tipos. Este es el único caso que dos grupos de combate pueden ocupar temporalmente una misma posición.

En un turno un jugador hace los movimientos que desee siempre y cuando no supere el total de posiciones de que dispone en cada jugada. Este número depende de la cantidad de territorios especiales que domina y la fortaleza de todos sus grupos de combate.

Adicionalmente, cada vez que se tome un territorio especial, el reino ganador puede intercambiar sus escuadrones por escuadrones más fuertes equivalentes con la condición que ellos estén en su territorio base.

REQUISITOS FUNCIONALES

La aplicación debe:

- ➔ Permitir generar aleatoriamente un campo dado ancho, largo, número total de territorios.
- ➔ Permitir leer un campo de batalla dado el nombre de un archivo (sólo para la competencia)
- ➔ Permitir identificar al jugador oponente con su nombre e icono.
- ➔ Permitir seleccionar el tipo de perfil del reino del computador.
- ➔ Permitir indicar cual reino va a realizar el primer movimiento.
- ➔ Presentar permanentemente el estado del campo de juego.
- ➔ Presentar permanentemente la información completa del estado de cada uno de los reinos indicando la fortaleza de sus ejércitos.
- ➔ Permitir que el usuario realice sus movimientos indicando posibles errores. Si el usuario se equivoca pierde el turno.
- ➔ Presentar el movimiento del reino computador de manera "inmediata" (Máximo 5 segundos)
- ➔ Decidir cuando termina el juego y comunicar la causa.
- ➔ Permitir terminar el juego en cualquier momento.

REQUISITOS DE DISEÑO

De extensión

Permitir generar nuevas versiones de la aplicación que incluyan:

- Nuevos terrenos (por ejemplo, en un futuro, islas a las que sólo puede llegar la fuerza aérea)
- Nuevos ejércitos (por ejemplo, fuerza aérea que puede pasar por lagos)
- Nuevos perfiles de tanque computador (por ejemplo, en la competencia, el nuevo perfil con la estrategia)

De visualización

- ✓ Todos los elementos del juego deben ofrecer una representación texto detallada que permita explorar el estado detallado de cada uno de ellos [toString] .
- ✓ Cada ubicación del campo de juego deben implementar una interfaz **CVisible** con dos métodos: uno para la representación texto y otro para la representación gráfica de ellas en el campo de juego.

De manejo de errores

- ➔ Deben definir mínimo una nueva clase excepción para manejar las excepciones propias.
- ➔ Los métodos correspondientes a las acciones deben lanzar una excepción si la acción solicitada no es válida explicando claramente su causa.
- ➔ Cuando ocurra una excepción no esperada o una propia grave se debe escribir esta información en el log de errores para los programadores y terminar la ejecución del mismo.

REQUISITOS DE DOCUMENTACIÓN

- ➔ Las fuentes deben estar documentadas siguiendo el estándar java (objetivos, parametros, retorno y excepciones)
- ➔ Para la entrega de revisión se deben presentar el diagrama de clases¹ de la capa de aplicación y el diagrama de colaboración² de un métodos : 1. un ejercito se mueve

1 Para todos los diagramas de clase, únicamente métodos públicos

2 Para todos los diagramas de colaboración, desde la capa de aplicación.

- ➔ Para la primera entrega se deben presentar el diagrama de clases de la capa de aplicación y los diagramas de colaboración de dos métodos : 1. un ejercito se mueve, 3. el reino azaroso decide su próxima jugada completa 4. un reino hace una jugada ya definida
- ➔ Para la segunda entrega se deben presentar el diagrama de clases de la capa de aplicación, el diagrama de clases de la capa de presentación y los diagramas de colaboración correspondientes al método: 1. un ejercito se mueve, 2. un reino decide cambiar escuadrones 3. el reino azaroso decide su próxima jugada completa 4. un reino hace una jugada ya definida 5. el reino destructor decide su próxima jugada
- ➔ Para la competencia se debe presentar el diagrama de colaboración del nuevo tanque computador realizando su movimiento

REQUISITOS DE ENTREGA

Primera revisión [11 de noviembre a par]

- ➔ **Diseño:** diagrama de clases

Primera entrega [22 de noviembre inicial a par – 29 de noviembre final a par]

- ➔ **Interfaz:** texto
- ➔ **Perfiles computador:** azaroso, solo avanzar
- ➔ **Armas :** infantería
- ➔ **Campo de batalla:** generación aleatoria.
- ➔ **Parámetros: POOBConquista1**

Segunda entrega [6 de diciembre a par - 13 de diciembre final]

- ➔ **Interfaz:** texto y gráfica
- ➔ **Perfiles computador:** azaroso, destructor avanzar y cambiar escuadrones
- ➔ **Campo de batalla:** generación aleatoria.
- ➔ **Armas :** infantería, caballería y artillería
- ➔ **Parámetros: POOBConquista2**

Competencia [15 de diciembre]

- ➔ **Interfaz:** texto y gráfica
- ➔ **Perfiles computador:** azaroso, destructor y ?nombre del nuevo perfil
- ➔ **Campo de batalla:** generación aleatoria y lectura de archivo
- ➔ **Ejercitos :** infantería, caballería y artillería
- ➔ **Parámetros: POOBConquista3**