ESCUELA COLOMBIANA DE INGENIERIA
PROGRAMACIÓN ORIENTADA A OBJETOS
Poober 2011-02
[Colaboración de David y Diana]

DESCRIPCIÓN

Bomber es una historia que se desarrolla en un lejano planeta donde Bomberman Blanco (o Shirbon), un robot defensor de la justicia creado por el Dr. Mimori, debe luchar contra Bomberman Negro, un robot que se convirtió en su enemigo por error de progamación.

El objetivo del proyecto es desarrollar **Poober,** una aplicación inspirada en **Boomber** que simula batallas entre Bomberman Blanco (el usuario) y Bomberman Negro(el computador). En estas batallas los robots inician en esquinas opuestas del tablero (N-O y S-E), realizan sus movimientos por turnos y terminan cuando un robot gana porque su oponente muere (pierde toda su energía) o huye (se ubica en una celda de salida del campo de batalla).

–--

Robots

Cada robot tiene un nivel de energía, bombas para atacar y algunos elementos de poder. El robot computador puede adoptar diferentes personalidades; entre ellas:

- Agresiva: su principal interés es matar a su oponente.
- Cobarde : su principal interés es huir del campo de batalla.

Campo de batalla

El campo de batalla es un espacio cuadrado de celdas en el que se mueven los robots y que pueden contener alguno de los elementos especiales:

- Bombas: se tienen tres tipos de bombas dependiendo de su alcance (en número de celdas) y su potencia (nivel de energía que hacen perder): *debil*, 2 y 30; *grande*, 4 y 50; y *fuerte*, 6 y 70.
- Monedas: de oro y de plata, permiten al robot hacer compras en el campo de batalla.
- Baterías: alimentan al robot aumentando su nivel de energía, se debe tener el dinero suficiente para comprarlas.
- Escudos: un robot sólo puede cubrirse con un escudo a la vez que lo protegerá del impacto de cualquier bomba y se destruirá.
- Piedras: impiden el paso de los robots. Actualmente existen tres tipos de piedras: *frágiles,* las destruye cualquier bomba, *fuertes,* se van debilitando econ cada impacto de las bomba y *fijas,* son indestructubles. Las celdas con piedras puede esconder monedas, baterías o escudos.
- Salidas: túneles que permiten salir del campo de batalla.

Movimientos

Las acciones básicas de un jugador son:

- ➔ Avanzar:un número de posiciones en una dirección indicada (norte, sur, oeste, este). El jugador puede pasar sobre las celdas con elementos especiales, salvo las piedras; pero, si los quiere tomar debe parar allí. No se permite que un robot se mueva a la celda donde está su oponente.
- ➔ Lanzar bomba: el robot puede lanzar la última bomba recogida con una fuerza (número de cuadros) y en una dirección (norte, sur, oeste, este) determinada. No se pueden lanzar bombas desde una celda con elementos especiales.

En un turno un jugador puede avanzar y lanzar bomba, en ese orden. Para que se puedan realizar las acciones el jugador debe tener la energía suficiente.

Nivel de energía

Cuando un jugador recibe el impacto de una bomba pierde parte de su nivel de energía, este valor depende de la potencia de las bombas y de distancia a su agresor. Adicionalmente, al avanzar o lanzar bombas también pierde energía dependiendo del peso que lleva a sus hombros (los elementos de poder pesan), en el primer caso, y de la fuerza y tipo de bomba en el segundo.

REQUISITOS FUNCIONALES

La aplicación debe:

- ➔ Permitir generar aleatoriamente un campo dado longitud y número de cada uno de los elementos especiales.
- ➔ Permitir seleccionar el tipo de perfil del robot del computador.
- ➔ Permitir indicar cual robot va a realizar el primer movimiento: blanco o negro. El robot que inicia se situa en la esquina noroeste.
- ➔ Presentar permanentemente el estado del campo de juego.
- ➔ Presentar, si se pide, la información completa del estado de cada uno de los robots: los elementos que lleva y su fortaleza.
- ➔ Ilustrar el máximo de energía y su gasto en cada movimiento.
- ➔ Permitir que el usuario realice sus movimientos.
- ➔ Presentar el movimiento del robot computador de manera "inmediata"
- ➔ Decidir cuando termina el juego y comunicar la causa.
- ➔ Permitir terminar el juego en cualquier momento.
- ➔ Permitir leer un campo de batalla dado el nombre de un archivo (sólo para la competencia)

REQUISITOS DE DISEÑO

De extensión

Permitir generar nuevas versiones de la aplicación que incluyan:

- Nuevos elementos (por ejemplo, un hospital donde el robot puede ir a curarse sus heridas)
- Nuevos perfiles (por ejemplo, en la competencia, el perfil con la estrategia del equipo para ganar)

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

- ➔ Diseño general: diagrama de paquetes
- ➔ Diseño de interfaz: diagrama de clases.
- ➔ Diseño de aplicación: diagrama de clases[1] y diagrama de secuencia [2] desde los métodos de la fachada del juego.
- ➔ Las fuentes deben estar documentadas siguiendo el estándar java (objetivos, parametros, retorno y excepciones)

1 Para todos los diagramas de clase, únicamente métodos públicos

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

REQUISITOS DE ENTREGA

Primera revisión [11 noviembre a par, publica el 10]

- ➔ **Diseño:** diagrama de estructura de clases

Segunda revisión [28 noviembre a par, publica el 27]

- ➔ **Diseño:** diagrama de estructura de clases
  esquema de ventanas y navegación (papel)
  métodos de comunicación interfaz-aplicación
- ➔ **Interfaz:** prototipo de la interfaz

Tercera revisión [9 diciembre a par, publica el 8]

- ➔ **Perfiles computador**: cobarde
- ➔ **Campo de batalla:** generación aleatoria.
- ➔ **Elementos :** bombas, piedras y salidas.
- ➔ **Parametros:** por equipo

Entrega final[12 de diciembre, publica el 11]

- ➔ **Campo de batalla:** generación aleatoria.
- ➔ **Parametros:** por equipo

Competencia[14 de diciembre]

- ➔ **Perfiles computador:** ?nuevo perfil con la estrategia propia
- ➔ **Campo de batalla:** lectura de archivo
- ➔ **Parametros:** acordados con los equipos seleccionados