

PEVAL3:

Existen 4 tipos de combatientes (podrá elegir el usuario qué tipo/s de combatientes pueden pelear, pudiendo existir dos guerreros por ejemplo en el mismo campo de batalla) (los distintos combatientes serán hilos): guerrero (más vida, más defensa, menos ataque), chamán (vida normal, se cura, ataque normal), cazador (menos vida, defensa normal, mucho ataque) y sacerdote (vida alta, se cura, menos ataque). Se enfrentarán entre 2 y 4 de ellos (elegido por el usuario), requiriendo un tiempo de ataque (Thread.sleep).

En primer lugar, los combatientes intentarán entrar al campo de batalla, pero solo entrarán los que lleguen hasta completar el aforo, el resto se irán (finalizarán los hilos). Se le dará un identificador a cada combatiente que entre al campo de batalla, que permitirá que sean seleccionados por los atacantes. Los combatientes accederán a la “taquilla” del campo de batalla en orden, haciendo cola (sincronización).

Los combatientes se “pelearán” por tomar el turno, pudiendo tomarlo solo uno y durmiendo el resto (sincronización, bloqueos de los hilos), se tendrá en cuenta que el último que ha atacado no puede volver a atacar en el siguiente turno. Al entrar al método atacar, elegirá el objetivo de su ataque, restándole vida a este (comunicación de datos). El último que quede con vida, o con más vida si la energía se agota, ganará. Las clases que posean curación, si su vida no está al máximo, se curarán al atacar. Las clases que poseen resistencia, recibirán menos daño al ser atacadas. Si dos o más combatientes empatan en la victoria, se mostrarán como ganadores.

Todos necesitarán absorber energía del planeta para poder atacar. Solo uno podrá acceder a esta energía (sincronización de objetos), en el momento en el que atacan (resta energía) o se curan (suma energía). La energía se recupera al finalizar el turno de ataque, por lo que un combatiente no puede recuperar energía para atacar después. Si la energía es 0 al finalizar un turno, nadie podrá atacar y el combate se dará por finalizado.