

ISW1-BattleField

El juego de “Campo minado” fue todo un éxito y hay que convertirlo en un juego de “Campo de Batalla”, es por eso que el juego pasó a ser para dos equipos y con varios jugadores por equipo, sin embargo se debe mantener la posibilidad de que haya un solo jugador de un solo equipo como hasta ahora, en cuyo caso será un Soldado con Bayoneta (más adelante se detallan los tipos de jugadores).

La cantidad de jugadores por equipo puede ser variable, pero ambos equipos deben tener la misma cantidad de jugadores y la suma de vidas de los jugadores de cada equipo debe ser igual.

Los jugadores del equipo 1 deben empezar en columnas configurables de la fila 1.

Los jugadores del equipo 2 deben empezar en columnas configurables de la fila superior.

Para ambos casos, deben empezar en posiciones vacías, no solo de elementos del campo sino también de jugadores.

Se deben seguir usando los mensajes de movimiento ya existentes, pero se irá moviendo un jugador de cada equipo a la vez, empezando por el primer jugador del equipo 1. Por ejemplo, si cada equipo tiene un jugador y se envía el mensaje **#movePlayerRight** dos veces, el primero moverá el jugador del equipo 1 y el segundo el jugador del equipo 2, en ambos casos a la derecha. Cuando se mueve el último jugador de un equipo, la próxima vez que le toque mover a ese equipo será el primer jugador del mismo.

El equipo 1 gana cuando alguno de sus jugadores llega a la fila superior (como era hasta ahora). El equipo 2 gana cuando alguno de sus jugadores llega a la fila 1.

El juego se acaba cuando algún equipo gana o cuando todos los jugadores se quedan sin vida.

Hay tres tipos de jugadores.

- **Soldado con pistola**: No puede usar armadura contra las bombas. Si cae en una posición con armadura no se la puede poner y la armadura debe quedar ahí.

- **Soldado con ametralladora**: Al igual que el soldado con pistola, no puede usar armadura.

- **Soldado con bayoneta**: Si puede usar cualquier tipo de armadura.

Solo puede haber un jugador por posición. Si un jugador quiere moverse a una posición ocupada por un jugador del mismo equipo, no puede y vuelve a su posición inicial.

Si un jugador se mueve a una posición ocupada por un jugador del otro equipo y si ambos jugadores poseen la misma arma (por ejemplo ambos con bayoneta), pierde la vida el jugador que llega a la posición si los segundos de la hora es par, pero si es impar pierde la vida el jugador que estaba en la posición.

Cuando los jugadores poseen distintas armas y se encuentran en una posición, quién pierde una vida se decide de la siguiente manera:

- Si uno tiene bayoneta, es el otro jugador el que pierde la vida

- Si uno tiene pistola y otro ametralladora, pierde la vida el que entra en la posición y mantiene la vida el que estaba en la posición.

Cuando un jugador pierde una vida en un enfrentamiento, debe volver a su posición inicial.

Se puede asumir que la posición inicial siempre estará vacía.

Si un jugador se queda sin vidas, no participa más del juego.

Tener en cuenta que:

- Siempre se analiza si hay algún elemento en la posición a la que llega un jugador y después si hay jugador. Por ejemplo, si hay un soldado en la salida de un túnel y cae otro soldado de otro equipo, el primero se mueve por el túnel.
- Si a la salida de un túnel hay un jugador, se debe hacer lo mismo que si el jugador se hubiese movido a esa posición.
- Siguen valiendo las mismas reglas que para el problema del parcial, o sea todos los tests que hay deben seguir pasando. Para esto hay que manejar el caso especial del primer equipo con un jugador y el segundo equipo sin jugadores.

Implementarlo usando TDD y teniendo en cuenta las heurísticas de diseño vistas a lo largo del cuatrimestre.

Los que tengan que recuperar el 1er parcial deben:

- Hacer lo necesario para que se pueda crear el juego con las nuevas condiciones
- Implementar solo el Soldado con Bayoneta
- Que se puedan mover los jugadores de cada equipo (que solo serán Soldados con Bayoneta) sin tener en cuenta que se pueden encontrar jugadores en una posición.

Recomendaciones y Protocolo a tener en cuenta:

- Para obtener la hora actual se envía el mensaje **#now** a **GregorianCalendar**.
- Para obtener los segundos de la hora actual se debe enviar los mensajes **#timeOfDay** y luego **#seconds** a un **GregorianCalendar**.
- Existe la clase **CircularReadStream** que representa un stream que nunca se acaba y que cuando se llega al final vuelve al principio.
Cuando se instancia, **#current** devuelve el primer elemento.
El mensaje **#next** mueve el stream al siguiente elemento.
El mensaje **#previous** mueve el stream al elemento anterior.
Ejemplo de instanciación: **CircularReadStream on: #(1 2 3) moving: NullUnit new**
- Se recomienda seguir el siguiente proceso al principio:
 - Empezar haciendo que los tests actuales funcionen para el tipo de jugador de Soldado con Bayoneta, pasando a dicha abstracción lo que está esparcido del "player" en **MineField**.
 - Crear una abstracción para representar el equipo de jugadores y hacer pasar todos los tests con esa nueva abstracción
 - Modificar el juego para que pueda recibir la configuración de los dos equipos.
- No hay problema que no se creen de manera completa los equipos y/o jugadores si así lo requiere la solución que hacen.
- Las recomendaciones acá dadas no implican que vayan a aprobar o desaprobado. Son recomendaciones que puede seguir o no. La nota dependerá de la solución que hagan, hayan seguido o no estas recomendaciones.

Entrega:

1. Entregar por mail el fileout de la categoría de clase **ISW1-2023-2C-Recuperatorio** que debe incluir toda la solución (modelo y tests). El archivo de fileout se debe llamar: **ISW1-2023-2C-Recuperatorio.st**

2. Entregar también por mail el archivo que se llama **CuisUniversity-nnnn.user.changes**
3. Probar que el archivo generado en 1) se cargue correctamente en una imagen “limpia” (o sea, sin la solución que crearon) y que todo funcione correctamente. Esto es fundamental para que no haya problemas de que falten clases/métodos/objetos en la entrega.
4. Realizar la entrega enviando mail a la lista de Docentes: ingsoft1-doc@dc.uba.ar con el **Subject: LU nnn-aa - Solución Recuperatorio 2c2023**
5. Subir a sus repos grupales los archivos **CuisUniversity-nnnn.image** y **CuisUniversity-nnnn.changes**. Debe **zippearlos** previamente para reducir su tamaño o podría dejar sin espacio disponible a sus compañeros. **Pueden eliminar las imágenes del 1er parcial para liberar espacio.**
6. Deberán subirlos al main branch de sus respectivos repos (tenga en cuenta hacer pull antes de ser necesario), y al subdirectorio **/Recuperatorio/LUnnn-aa/**
7. **RECOMENDACIÓN IMPORTANTE: Salvar la imagen de manera frecuente o con el autosave**

IMPORTANTE:

No retirarse sin tener el ok de los docentes de haber recibido el mail con la resolución.