

Ingeniería del Software II – Enunciado de la práctica del primer parcial

Se desea desarrollar un sistema para jugar al *Ultimate Tic-Tac-Toe*¹ (el *Tres en raya definitivo*). Este juego se juega entre dos jugadores y comienza en un tablero vacío como el que se muestra en el lado izquierdo de la Figura 1: como se ve, el tablero está compuesto por 9 tableros del juego habitual de las *Tres en raya*. A cada pequeño tablero lo identificamos por una letra (*A*, *B* o *C*) para identificar la columna y por un número (*1*, *2*, o *3*) para identificar la fila. Dentro de cada pequeño tablero usamos la misma técnica de identificación. Así, por ejemplo, el aspa que hay colocada en el lado derecho de la figura está en la posición (*C*, *1*) del tablero (*A*, *2*).

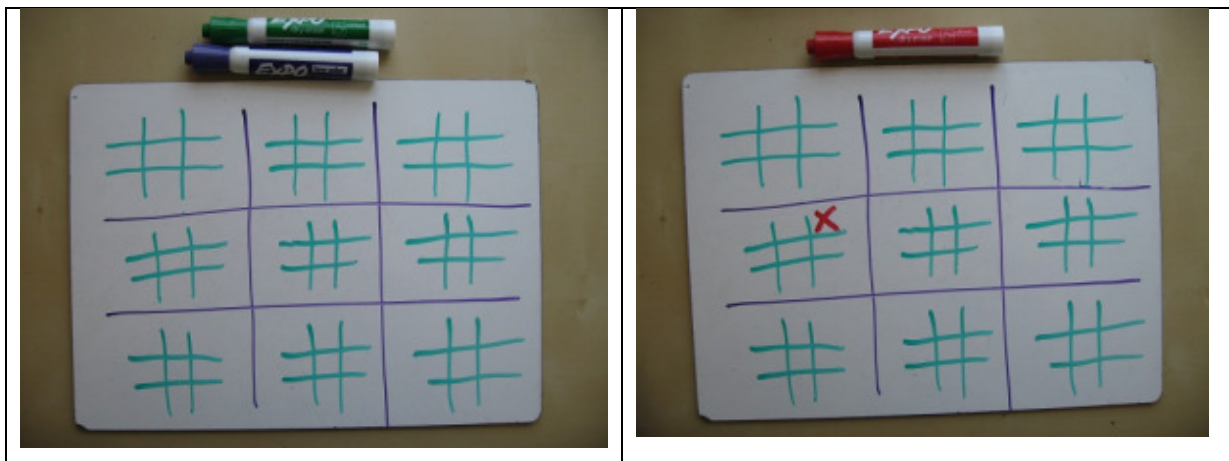


Figura 1

Supongamos que la partida la ha comenzado el jugador *X* con el movimiento mostrado en el lado derecho de la figura anterior. El siguiente jugador, *Y*, está obligado a colocar su “piedra” (marca, ficha, etc.) en el tablero (*C*, *1*): es decir, en el pequeño tablero cuyas coordenadas coincidan con la posición de la piedra colocada por el jugador anterior dentro del pequeño tablero. Es decir: como *X* ha puesto su piedra en (*C*, *1*), *Y* debe colocar la suya en el tablero (*C*, *1*). Eso sí: en la posición que prefiera.

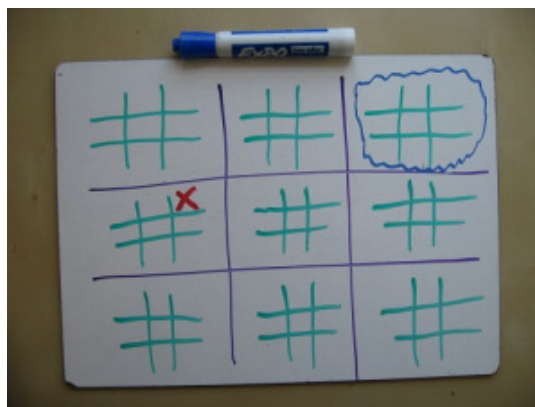


Figura 2

¹ En <http://mathwithbaddrawings.com/2013/06/16/ultimate-tic-tac-toe/> hay una descripción completa del juego.

La partida se gana cuando un jugador hace tres en raya en tres pequeños tableros.

En alguna circunstancia es posible que el pequeño tablero en el que debe ponerse ficha esté ya ocupado: en este caso, el jugador puede colocar su piedra en el lugar que desee.

Para el primer parcial, se debe desarrollar un sistema cliente-servidor que, utilizando *rmi*, permita jugar a este juego. Un posible escenario de uso es el siguiente:

- 1) El jugador *X*, usando su *Cliente*, se identifica mediante su login y password en el *Servidor*.
- 2) El *Servidor* valida sus credenciales en una base de datos y le da acceso, mostrándole una lista de los jugadores disponibles en ese momento.
- 3) *X*, mediante su *Cliente*, reta al jugador *Y*, e *Y* acepta jugar. La petición de reto le llega a *Y* a través del *Servidor*: es decir, hay un tráfico $X \rightarrow \text{Servidor} \rightarrow Y \rightarrow \text{Servidor} \rightarrow X$.
- 4) El *Servidor* crea un tablero vacío para que *X* e *Y* jueguen.
- 5) *X* e *Y* van jugando. El servidor toma nota de todos los movimientos que se van produciendo y los guarda en la base de datos.
- 6) Cuando la partida termina, anota la victoria al jugador que corresponda.

La legalidad de los movimientos se comprueba en el cliente.

Para el segundo parcial, se dará la opción de desarrollar el cliente en versión *web* o para teléfono móvil.

Para este primer parcial, se pide:

- 1) Descripción detallada de todos los requisitos funcionales del sistema.
- 2) Descripción de cada requisito funcional en forma de máquina de estados.
- 3) Casos de prueba funcionales que pueden derivarse de cada requisito funcional.
- 4) Diseño arquitectónico.
- 5) Para cada requisito funcional, detalle de sus correspondientes etapas de diseño, implementación y pruebas, incluyendo los resultados de la ejecución de las pruebas.
- 6) Diseño final detallado del cliente y del servidor.
- 7) Implementación de ambos.

Todos los miembros de cada grupo deberán hacer una presentación y defensa del trabajo.

La fecha tope de entrega de la práctica es el 7 de febrero de 2014.