Master Software Craftmanship

2018/2019

Memoria Práctica 2

Implementación de pruebas automáticas

Autores

* Javier Antonio Valdez Palomino
* Francisco Javier Jiménez Olivares

Contents

[Pruebas Automáticas 2](#_Toc7131819)

[Pruebas unitarias de la clase Board 2](#_Toc7131820)

[Objetivo 2](#_Toc7131821)

[Implementación 2](#_Toc7131822)

[Pruebas con dobles de la clase TicTacToeGame 3](#_Toc7131823)

[Objetivo 3](#_Toc7131824)

[Implementación 3](#_Toc7131825)

[Pruebas de sistema de la aplicación 4](#_Toc7131826)

[Objetivo 4](#_Toc7131827)

[Implementación 4](#_Toc7131828)

# Pruebas Automáticas

## Pruebas unitarias de la clase Board

### Objetivo

En estas pruebas se comprueba que la clase Board implementa correctamente la detección de que un jugador ha ganado o de que se ha empatado.

Se implementan varias pruebas que cambian el estado de las celdas del tablero y se verifican los métodos getCellsIfWinner() y checkDraw().

### Implementación

Se ha implementado un caso de prueba unitario para cada una de las combinaciones de jugadas en las que se puede obtener tres-en-raya (8 en total), alternando entre el jugador “X” y el jugador “0”.

Adicionalmente se ha implementado un caso de prueba en el que se verifica una posible situación de “empate”.

En cada una las pruebas se verifican que:

* El procedimiento getCellsIfWinner() devuelve los valores correctos de acuerdo al jugador que se indica como parámetro de entrada. Esto es:
  + La lista de celdas que componen la jugada ganadora.
  + O Null en caso de que el jugador no se el ganador.
* El procedimiento checkDraw()detecta correctamente la situación de empate, devolviendo:
  + False si no hay empate.
  + True si hay empate.

La función siguiente permite la modificación sencilla de las celdas del tablero de acuerdo al valor indicado como parámetro:

**public** **void** setCell(Board board, **int** cellNumber, String value)

Haciendo uso de esta función, los casos de prueba quedan sencillos de codificar y sencillos de entender lo que facilita el futuro mantenimiento de las pruebas.

## Pruebas con dobles de la clase TicTacToeGame

### Objetivo

En estas pruebas se comprueba que la clase TicTacToeGame implementa de forma adecuada el juego. Se llama a los distintos métodos de la clase simulando los mensajes que llegan de los navegadores durante el uso normal de la aplicación.

Además, se deberán simular las conexiones creado dobles de los objetos Connection (que representan una comunicación WebSockets). Estas conexiones simuladas se utilizarán para verificar si los eventos que envía TicTacToeGame a los navegadores web (representados por los dobles de Connection) son los esperados.

### Implementación

Se han implementado tres casos distintos en los que el jugador que empieza la partida gana, otros tres casos en los que gana el segundo jugador y otros tres en distintos en los que ocurre un empate.

La siguiente función permite la generalización de los casos de prueba, ejecutando cada una de los pasos necesarios para el procedimiento completo de cada ejecución:

**public** **void** ticTacToeRunTest(**int**[] cells, **int** winner)

Nuevos casos de prueba se pueden añadir fácilmente ejecutando esta función con distintos parámetros de entrada.

## Pruebas de sistema de la aplicación

### Objetivo

Se implementan pruebas de sistema con Selenium. Para simular una partida la prueba inicia dos navegadores web de forma simultánea e interactúa con ellos de forma alternativa. De esta forma, se simula un partida por turnos.

### Implementación

Se han implementado tres casos distintos en los que el jugador que empieza la partida gana, otros tres casos en los que gana el segundo jugador y otros tres en distintos en los que ocurre un empate.

Las siguientes funciones permiten generar nuevos casos de prueba de manera sencilla:

**public** **void** startGame()

Arranca los navegadores e introduce los jugadores para comenzar la partida.

**public** **void** clickOnCell(WebDriver webDriver, **int** cellNumber)

Ejecuta el click de cada jugador en la celda seleccionada por cada jugador.

**public** **void** verifyMessages(String expectedMessage)

Comprueba los mensajes finales de final de parida.