Bases de Dades Avançades

PRÀCTICA 1

HIBERNATE

Gestió d’una aerolínia

Karen Samsó

Judit Domènech

2016

Universitat de Barcelona

**Índex**

* **1.** Introducció .................................................................................................................... pàg. 3
* **2.** Funcionament i implementació .................................................................................... pàg. 3
  + **2.1.** Joc ................................................................................................................. pàg. 3
    - **2.1.1.** Elements bàsics ............................................................................. pàg. 3
    - **2.1.2.** Estructura de les classes ............................................................... pàg. 4
  + **2.2.** Client ............................................................................................................. pàg.
  + **2.3.** Servidor ......................................................................................................... pàg.
    - **2.3.1.** Servidor amb *threads* .................................................................... pàg.
    - **2.3.2.** Servidor amb Selector ................................................................... pàg.
* **3.** Problemes sorgits .......................................................................................................... pàg.
* **4.** Javadoc .......................................................................................................................... pàg.
* **5.** Conclusions ................................................................................................................... pàg.

**1. Introducció**

L’objectiu final d’aquesta primera pràctica de Software Distribuït és aprendre a utilitzar els mecanismes de programació Client-Servidor en Java, concretament, aprendre a programar amb Sockets, i servidors amb i sense *threads*.

Per tal d’aconseguir aquest objectiu, es demana que programem un client i un servidor que implementin el protocol per el joc “Enfonsar la flota”, seguint les especificacions de l’enunciat.

“Enfonsar la flota” és un joc per torns de dos jugadors, on cada un ha d’intentar descobrir la posició (coordenades) dels vaixells del contrincant i enfonsar-los tots per guanyar. De manera que, es tenen dos panells per cada jugador (el panell on el jugador situa els seus vaixells i registra els dispars del contrincant contra la seva flota, i el panell on va registrant els dispars encertats i fallits a la flota del contrincant).

Així doncs, seguint les instruccions de protocol sobre com es desenvolupen les accions en la partida i com és la comunicació entre el client (jugador o màquina segons s’esculli) i el servidor (màquina), s’ha prosseguit a implementar el joc en el següent ordre: primer les seves funcionalitats bàsiques, després el client i, finalment, el servidor (tant amb *threads* com amb Selector).

**2. Funcionament i implementació**

A continuació s’explicarà, a grans trets, el funcionament del joc i la comunicació entre Client i Servidor, així com la seva implementació (a nivell d’organització i orientació a objectes).

La informació detallada sobre la implementació del codi es troba a l’apartat *4. Javadoc*, on s’explica cada un dels diferents mètodes de les classes del nostre programa.

**2.1. Joc**

2.1.1. Elements bàsics

Els elements físics bàsics necessaris per jugar a “Enfonsar la flota” són els panells i els vaixells, a partir d’aquí, s’han creat les classes *Grid* i *Ship* que els representaran.

Dins del panell, cada vaixell estarà situat en una sèrie de coordenades adjacents, és a dir, cada un dels trossos del vaixell estarà en una posició que correspondrà a una lletra seguida d’un número. Aquesta posició està representada per la classe *Position*.

Les situacions dels vaixells en el panell poden ser introduïdes via teclat (en el cas del client), o via fitxer (en un format concret) o generades aleatòriament en ambdós casos (client i servidor), segons s’indiqui en executar la partida.

Finalment, es necessita aquesta partida on tenir i inicialitzar tots aquests elements, que es representa amb la classe *Game*.

2.1.2. Estructura de les classes

Les classes definides anteriorment estan relacionades entre elles per a un bon ús de la programació orientada a objectes i delegació de responsabilitats.

**2.2. Client**

**2.3. Servidor**

2.3.1. Servidor amb *threads*

2.3.2. Servidor amb Selector

**3. Problemes sorgits**

**4. Javadoc**

**5. Conclusions**