T-106.5600 Concurrent Programming Assignment 1 - Plan

by: Hannu Huhtanen 291288

--Goal--
Implement a Hangman game server that supports multiple concurrent users over TCP/IP.

--Style guidelines--
All the code will be written cleanly and according to industry best practises as far as the assignment requirements allow.

--Implementation plan--
Task A: Event queue

Blocking queue has two main methods get and put which should block until they have completed their action. To implement this get will invoke Object.wait() until the queue isn't empty. The put method will similarly wait until the queue has less items than the queue's maximum size. Once get method manages to retrieve a item from the queue it will notify all threads that may be waiting for queue to have room for new items. Similarly the put method will notify all the threads waiting for queue to be non-empty.

Task B: Reactor

The main part of the reactor implementation is the Dispatcher class. It's task is to keep a list of registered EventHandlers and pass events from Handles to their corresponding Handlers. This will be implemented by spawning a new WorkerThread to read a Handle when its Handler is registered. The WorkerThread will pass everything from the Handle to a BlockingEventQueue. The Dispatcher.handleEvents() method starts a process of passing messages from Handles to Handlers and it will use the Dispatcher.select() to wait a message from any of the handles and then pass it forward. The Dispatcher.select can simply ask for a message from the same BlockingEventQueue that the WorkerThreads use. If a null is read from a handle or Dispatcher.removeHandler() is called the corresponding WorkerThread will be cancelled so that no more messages are read from the Handle.

Task C: Hangman

The Hangman part of the project manages the game part of the game server. It will use a Dispatcher instance to handle all the player connections.