

Sistemas Tolerantes a Falhas e Tempo-Real 2001-2002

Notas sobre o protótipo para o trabalho

Ricardo Almeida

O código base do projecto foi criado de modo a ser o mais modular possível. Desse modo, foram criados vários “packages” que são explicados de seguida. A explicação é curta, pois pretende apenas dar uma visão geral do protótipo (o código encontra-se documentado).

Package stftr

Pacote “raiz”. Aqui apenas se encontra uma classe, mas todos os outros packages estão dentro deste.

Board.java – Responsável por guardar a representação do jogo. Internamente guarda uma matriz de inteiros em que os primeiros 28 bits para o identificador (só usado para o jogador), 2 bits para o tipo (jogador ou parede) e finalmente os últimos 2 para a direcção. Existem métodos para não ser necessário mexer directamente com os bits.

Package stftr.player

Neste package encontra-se a aplicação do jogador. A aplicação é composta por duas classes:

GameBoardComponent.java – Componente que mostra o estado de uma board. Nesta classe existe a constante `IMAGES_PATH` que deve ser alterada para apontar para o directório onde se encontram as imagens.

GamePlayer.java – Onde se encontra o main. Verifica os parametros da linha de comandos e lança a aplicação.

Package stftr.server

Neste package encontra-se a aplicação servidor. Também esta está dividida em duas classes:

GameServer.java – Contém a lógica do servidor. É esta classe que altera a board (ex: adiciona jogador, move jogador, etc).

LaunchGameServer.java – Onde se encontra o mais. Verifica os parametros da linha de comandos e lança o servidor. Antes de lançar o servidor vai ler de um ficheiro, especificado na linha de comandos, a board que vai ser utilizada. Existe um ficheiro de exemplo no directório stftr.

Package stftr.protocols

Neste package encontram-se os protocolos em APPIA que permitem fazer a comunicação entre o cliente e o servidor. Este pacote em particular não tem classes, apenas sub-packages.

Package stftr.protocols.Player

Neste package encontra-se o protocolo do jogador. É a camada de topo e abaixo dela apenas existe o Tcp. O jogador informa esta camada para enviar uma accção (movimento ou disparo) e quando esta camada recebe uma board, informa a aplicação que o estado da board alterou e que esta deve ser redesenhada.

Package stftr.protocols.Game

Neste package encontra-se o protocolo que é a camada de topo do servidor. Por cada pedido de accção que recebe, executa-a e envia a board resultante para baixo.

Package stftr.protocols.Distribute

Neste package encontra-se o protocolo que é responsável por gerir os endereços dos jogadores e de distribuir as boards por eles. Abaixo desta camada está o Tcp.

Resumo

Na tabela ao lado encontra-se o resumo das várias camadas APPIA existentes nas duas aplicações (jogador e servidor).

Player	<table border="1"><tr><td>Player</td></tr><tr><td>TcpComplete</td></tr></table>	Player	TcpComplete	
Player				
TcpComplete				
Server	<table border="1"><tr><td>Game</td></tr><tr><td>Distribute</td></tr><tr><td>TcpComplete</td></tr></table>	Game	Distribute	TcpComplete
Game				
Distribute				
TcpComplete				

Nota: Não esquecer de alterar a variável de ambiente CLASSPATH para apontar para o directório do APPIA e de stftr.