

Trabalho Prático N°2 – Desenho e implementação de um jogo distribuído na Internet

Jorge Ferreira, José Pereira, and Pedro Cunha

Universidade do Minho, Departamento de Informática, 4710-057 Braga, Portugal
{a64293,a67677,a67680}@alunos.uminho.pt

Abstract. Neste relatório iremos apresentar a nossa resolução do trabalho prático de Comunicação por Computadores. Vamos descrever o essencial do projecto assim como a sua implementação e algumas decisões tomadas. No fim vamos apresentar alguns testes da aplicação.

1 Introdução

Este trabalho requer que seja desenvolvido um jogo baseando-se na distribuição de conteúdos com interação dos utilizadores. A aplicação não pode basear-se em tecnologia baseada em http logo deverá implementar comunicações entre servidores e comunicações UDP entre clientes e servidores. O objetivo do jogo é vários jogadores jogarem um desafio de perguntas de escolha múltipla onde o vencedor é o utilizador que acertar mais perguntas sendo que o critério de desempate é o tempo de resposta às questões certas. No fim do jogo a pontuação é integrada no score global do utilizador.

2 Implementação

O ponto fundamental neste trabalho é a comunicação entre cliente e servidor. Para a mesma foram criados PDU's para regular a comunicação entre os intervenientes. Os PDU's implementados são uma versão alterada dos apresentados no enunciado, de forma a incluir por exemplo o nome do utilizador ao criar um desafio.

2.1 Bibliotecas Utilizadas

As bibliotecas/classes externas que foram utilizadas para a realização do projecto foram as seguintes:

- *Apache.IOUtils* - Foi utilizada para utilizar o método que transforma um ficheiro num array de Bytes. As restantes classes da biblioteca Apache foram incluídas para evitar possíveis situações de incompatibilidade.
- *JLGui.BasicPlayer* - Foi utilizada para reproduzir a Música, todas as restantes bibliotecas dos packages jlgui foram incluídos para evitar possíveis situações de incompatibilidade.

- *Business.Picture* - Foi utilizada para apresentar Imagem sobre uma janela de Java Swing.
- *Business.Crono* - Foi utilizada para contar os segundos entre o start() e stop().
- *Business.Input* - Foi utilizada para possuir métodos robustos e protegidos de Excepções para Input do teclado.
- *Business.Menu* - Foi utilizada para apresentar menus em formato texto.

3 Modo de Funcionamento

3.1 Servidor

Ao iniciar o servidor, este irá carregar os dados de imagens, texto e perguntas existentes (dos ficheiros *Imagens.txt*, *Musicas.txt* e *Questoes.txt*) para uma estrutura interna que servirá de base para todos os Clientes. De seguida entra em ciclo infinito à espera de mensagens vindas de clientes.

Ao receber um pacote de um cliente, o servidor irá primeiro verificar o seu cabeçalho para saber qual é o tipo de ação que deve realizar e, de seguida irá processar o pedido e enviar a resposta correspondente.

Para gerar um desafio é necessário que existam mais de 10 perguntas na estrutura e os desafios serão gerados com 10 perguntas aleatórias da base de dados.

3.2 Cliente

Um cliente ao iniciar irá enviar logo um pacote 'Hello', ficando à espera que o servidor responda com OK. Caso isso não aconteça, um novo pacote Hello é enviado até ser recebido um Ok. De seguida irá ser apresentado o menu de início de sessão/registo. Caso o utilizador pretenda registar-se deverá colocar o seu nome, alcunha e palavra passe. Será enviado um pacote 'Register' com os três campos referidos, ficando à espera de um OK por parte do servidor. Caso haja erro o utilizador deverá inserir uma alcunha diferente. Caso o utilizador pretenda realizar o login no jogo deverá inserir a alcunha e a palavra passe. Será enviado um pacote 'Login' com os campos referidos, ficando à espera que o servidor envie uma resposta com o nome e a pontuação actual. Neste momento encontra-se o menu inicial:

- "Criar Desafio" - Irá pedir ao utilizador que escolha o nome para o desafio, envia um MakeChallenge com o nome de utilizador e nome de desafio, espera um OK por parte do servidor e executa o menu de Desafio.
- "Ver Classificações" - Envia um List Rankings, espera pela resposta por parte do servidor e apresenta a resposta.
- "Listar Desafios" - Envia um List Challenges, espera pela resposta do servidor e apresenta a mesma.
- "Realizar Desafio" - Irá pedir ao utilizador que escolha o nome do desafio que pretende realizar, será enviado um AcceptChallenge, esperando um OK do servidor e executa o Menu de Desafio.

- "Sair" - Irá enviar um pacote Logout ao servidor e espera um OK.

Caso se encontre no menu de Desafio, as seguintes opções são possíveis:

- "Modo Jogo" - Entrará no modo de jogo, em que irão ser carregadas e apresentadas, uma a uma as perguntas que perfazem o desafio (Música e Imagem incluídas). O utilizador terá sempre a opção de desistir do desafio, premindo em 0, e será enviado um pacote Quit e a pontuação obtida até ao momento será descartada. No caso de se tratar de uma resposta a uma pergunta irá ser enviado um pacote com a alcunha do utilizador, o índice da pergunta no desafio, o nome do desafio e a opção escolhida.
- "Terminar Desafio" - Para terminar o desafio é necessário que o utilizador que o termina seja o utilizador que o criou. Será enviado um pacote EndChallenge com o nome do Utilizador actual e o nome do Desafio a terminar, esperando um OK.
- "Eliminar Desafio" - Será enviado um pacote Delete com o nome do Desafio, para o poder fazer é necessário que o desafio a eliminar se encontre nos desafios já terminados.

4 Conclusão