

Trabalho do Jogo em Java com Rede.

Estou disponibilizando dois jogos: o jogo da Luta e o Chinelada Na Batata. Caso você precise de um exemplo de como fazer um programa usando rede, veja os programas exemplo nesta atividade.

Jogo de Luta

- Os arquivos contidos no JogoServidor.zip tem o servidor, inicie a execução pelo arquivo JogoServidor.java. Este programa deve ser executado antes dos clientes.
- Os arquivos contidos no JogoCliente.zip tem o cliente, enquanto o JogoServidor estiver executando, execute em outras duas janelas de comandos (cmd ou terminal) o programa JogoCliente.java duas vezes. Nos clientes, para jogar, use as teclas <seta para direita> para andar, "p" para parar e "s" para socar.

Jogo da Chinelada na Barata

- Os arquivos contidos no ServidorBarataJogo.zip tem o servidor, inicie a execução pelo arquivo JogoServidor.java. Este programa deve ser executado antes dos clientes.
- Os arquivos contidos no ChineladaNaBarata.zip tem o cliente, enquanto o JogoServidor estiver executando, execute em outras duas janelas de comandos (cmd ou terminal) o programa ChineladaNaBarata.java duas vezes. Nests clientes, para jogar, use o mouse, posicione o cursor sobre a barata e clique com o botão esquerdo, se acertar ganhará pontos.

Estude os programas. Para você entender melhor as alterações que precisa fazer no seu programa apresentado na atividade de jogo anterior, analise o código contido no JogoCliente.zip especialmente a classe em Rede.java e, no arquivo JogoCliente.java, a classe anônima que descende de KeyAdapter e a classe anônima que implementa Runnable. Para ver outras possibilidades, analise os programas contidos no ChineladaNaBarata.zip.

Requisitos que os seus programas devem seguir

- Ser um jogo em tempo real, jogos de turnos não serão aceitos;
- Ter um programa servidor independente do programa cliente;
- Ter um programa cliente que será executado um para cada jogador, este programa deve se conectar ao servidor para o jogo funcionar;
- A lógica do jogo **DEVERÁ** estar no servidor. Veja os exemplos;
- Não use *package* no seu arquivo, faça todos os programas na mesma pasta.

Entrega do trabalho

O trabalho deve ser enviado em dois arquivos .zip:

- Um arquivo chamado Servidor.zip com os arquivos para o servidor funcionar, contendo uma classe chamada Servidor com o método main() que inicia o servidor;
- Um segundo arquivo Cliente.zip com os arquivos para o cliente, contendo uma classe chamada Cliente com o método main() que inicie o cliente. Este programa cliente será executado duas vezes.

Nos arquivos .zip, para entregar o trabalho, coloque apenas os arquivos para que o seu programa possa ser compilado e executado. Não coloque arquivos de configuração da sua IDE, nem arquivos

compilados como .class ou .jar. Não use *package* no seus programas, mas separe o cliente e o servidor em duas pastas uma chamada Servidor e outra chamada Cliente.

Alguns outros detalhes sobre o chinelada na Barata

Este jogo funciona com mouse.

O Cliente envia:

- Sempre que o mouse for movimentado, o cliente envia ao servidor: uma String com o texto "MM" seguida pelas coordenadas x e y no mouse;
- Quando o botão mouse for pressionado, ou seja "clicado", o cliente envia ao servidor: uma String com o texto "MP" seguida pelas coordenadas x e y no mouse.

O Cliente recebe: O cliente fica constantemente esperando por Strings com comandos enviados pelo servidor, estes comandos são: "ACERTOU", "ADVERSARIO ACERTOU", "ERROU", "ADVERSARIO ERROU", "PERDEU", "GANHOU", "ADVERSARIO GANHOU", "SOH_COORDENADAS".

Depois de cada um destes comandos chegam também as coordenadas da bara, do chinela A e do chinelo B. Em seguida os comandos são testados e ações são feitas pelo cliente para atualizar a janela do jogo, estas ações podem necessitar a recepção de mais dados. Por exemplo o comando "SOH_COORDENADAS", recebe apenas as coordenadas, não recebe nenhum dado extra e em seguida chama a função repaint() para desenhar a barata e os chinelos nas suas posições atuais do servidor.

Ultimas considerações

Os programas foram feitos com classes em arquivos separados, para compilar todas as classes envolvidas no programa de uma só vez, basta compilar o programa que cria as instâncias das outras classes, dessa forma o compilador javac irá procurar na pasta os arquivos .java e os compilará também. Para que isso funcione as classes tem que ter o mesmo nome dos arquivos .java onde elas estão definidas.

Qualquer dúvida estou a disposição.