

A priori o projeto é assim delineado:

1. Haverá um servidor sempre ativo que será responsável por:

1.1. Esperar por novos jogadores;

*Isso significa que deve haver uma classe que será a classe escutadora. Ela vai ficar sempre ativa num looping infinito enquanto o servidor estiver ativo. Essa classe vai receber as novas conexões (via ServerSocket) e passar o Socket criado a partir do ServerSocket para uma nova thread que será responsável pela comunicação entre o cliente e o servidor. Nesta parte vamos ter que manter o estado de cada conexão. Vamos ter que criar uma lista de usuários para saber como se comunicar com cada usuário específico.*

1.2. Aglutinar todos os jogadores numa área comum, que deverá:

1.2.1. Possuir uma lista de usuários visível a todos;

*Aqui, usando aquela lista que a classe escutadora vai criar, agente vai ter que preencher um determinado dispositivo de Java que lista os usuários como no mIRC com os nicknames dos usuários que deverá ser preenchido por eles antes de se conectar com o servidor. Teremos que configurar o que vai acontecer com cada evento de mouse que houver nesta lista. Duplo-Clique, Clique-Simples, Botão direito... essas coisas.*

1.2.2. Possuir uma lista de jogos ativos;

*Quando cada usuário criar um jogo, o servidor deverá atualizar sua lista de jogos. E preencher da mesma forma como ocorre com os usuários. E configurar as mesmas coisas que no item anterior.*

1.2.3. Possuir um chat via multicast entre todos os usuários.

*Este chat vai ser composto, no cliente, de uma TextArea e um TextField. Na textArea agente escreve o que chegar pelo multicast tipo textArea.append(mensagem) e no textField agente pega o que estiver escrito depois que o user der um enter e envia pelo multicast.*

1.3. Manter o estado dos jogos que estejam acontecendo;

*Não sei ainda ao certo como vai ficar isso. Vamo conversar com castor pra ver como fica. A principio o cliente manda uma mensagem ao servidor dizendo que quer abrir um jogo. Ai não sei ao certo como fazer, se o cliente ou o servidor que vai manter o jogo. Se for o servidor ele deve criar o objeto jogo, responder pro cliente: "ok, jogo criado e esperando jogador.", atualizar a lista de jogos e deixar o jogo em espera. Se for o cliente, basta criar o jogo no cliente, mandar uma mensagem pro servidor dizendo que criou um jogo e o servidor atualizar a lista de jogos e deixar em espera por outro jogador.*

1.4. Manter o estado dos jogadores conectados;

*Aqui é com o servidor. Voce se conecta, ele pega teus dados e coloca você numa lista de jogadores, atualiza a lista de jogadores ativos e pronto.*

2. Haverá um cliente que se valerá de uma interface gráfica para que o jogador possa:

2.1. Conectar-se com a área comum do servidor e saber quais jogadores estão ativos;

*Isso aqui significa dizer que vai haver uma salinha de chat inicial que vai mostrar a lista de jogadores, jogos e o chat.*

2.2. Permitir que o jogador estabeleça um jogo;

*Aqui nós caímos na mesma conversa do item 1.3. Depois de decidir como fazer agente vê.*

2.3. Permitir que o jogador se conecte a jogos já existentes;

*Isso agente faz via ações de mouse na lista de jogos. Com botão direito ou duplo clique.*

2.4. Prover um subrotina que o permita:

*Isso aqui é a gui do jogo. Aquela clássica do jogo da velha e um textField e uma textArea pra exibir as conversas do chat. Neste caso da pra fazer o chat peer-to-peer. Basta enviar pro cliente o ip do outro jogador.*

2.4.1. Interagir e executar suas jogadas;

2.4.2. Perceber o estado atual do jogo;

2.4.3. Permitir que os dois participantes do jogo atual se comuniquem de forma privada.