**Especificação do Primeiro Trabalho**

|  |
| --- |
| Descrição e Objetivo  Esse trabalho consiste na implementação de um protótipo de jogo de *Truco multiplayer online*. O jogo será dividido em duas aplicações: um cliente e um servidor. Cada jogador disparará uma instância da aplicação cliente, especificando o endereço do servidor que deseja utilizar. A aplicação cliente, então, abrirá uma conexão TCP com esse servidor. O lado servidor, por sua vez, quando executado, abrirá um *socket* TCP e aguardará conexões dos clientes. Quando quatro (número típico de jogadores em uma partida de Truco) conexões cliente forem recebidas, o servidor dará início à partida. A partir deste ponto, o servidor controlará todas as ações e eventos do jogo. Neste sentido, as aplicações clientes funcionarão basicamente como uma interface da aplicação com o jogador.  O restante deste documento descreverá em maiores detalhes a implementação esperada, incluindo os requisitos e restrições.  Regras do Jogo  Como o enfoque deste trabalho se dará majoritariamente nos aspectos de comunicação e redes, consideraremos aqui um subconjunto simplificado das regras do jogo de Truco. É este subconjunto que deverá ser implementado nesse trabalho.  O jogo é jogado por duas equipes de 2 jogadores cada. Ao longo desta especificação, esses jogadores serão denotados por **jogador 1, jogador 2, jogador 3 e jogador 4.** **O jogador 1 será aquele que primeiro se conectar ao servidor, seguido do jogador 2 e assim sucessivamente.** A primeira dupla será sempre formada pelos jogadores 1 e 3, enquanto a segunda será formada pelos jogadores 2 e 4.  Uma partida de truco se divide em múltiplas **mãos**. Cada mão vale, inicialmente, 1 ponto. De acordo com a situação, no entanto, os jogadores podem aumentar o valor da mão para 3, 6, 9 ou 12 pontos. **Ganha a partida a equipe que chegar primeiro aos 12 pontos na soma das mãos jogadas.**  No início de cada mão, as cartas são embaralhadas e distribuídas aos jogadores. Cada jogador recebe três cartas. **A ordem em que as cartas são dadas para cada jogador não é importante nesse trabalho**. Após as 12 cartas terem sido distribuídas aos jogadores, retira-se a próxima carta ainda no baralho. Essa carta recebe o nome de **vira**, e deve ser exibida a todos os jogadores.  Uma vez dadas as cartas e o vira, a mão começa efetivamente. Neste trabalho, consideraremos que **o primeiro jogador a jogar na primeira mão de uma partida deverá ser sempre o jogador 1.**Para as mãos subsequentes, os jogadores se alternarão em ordem: a segunda mão será iniciada pelo jogador 2, a terceira pelo jogador 3, etc.  Cada mão é dividida em até 3 rodadas. Em cada rodada, os jogadores são percorridos em ordem. Sempre que um jogador se encontra na sua vez de jogar, ele deverá escolher uma das cartas na sua mão e jogá-la. Ao jogar a carta, **o jogador pode optar por jogá-la aberta (*i.e.*, visível aos demais jogadores) ou fechada (*i.e.*, invisível aos demais jogadores)**. A vez é, então, passada ao próximo jogador na ordem. Isso se repete até que todos os jogadores tenham jogado suas cartas naquela rodada.  Ao final de uma rodada, deve-se determinar a equipe vencedora. Será declarada vencedora daquela rodada a equipe do jogador que houver jogado a carta mais alta. **Repare que cartas jogadas fechadas são desconsideradas, não tendo qualquer valor.** Uma equipe que vence duas das três rodadas é considerada a vencedora de uma mão. No caso de uma mesma equipe vencer as duas primeiras rodadas de uma mão, não há necessidade de se jogar a terceira. A próxima rodada de uma mão é sempre iniciada pelo jogador que jogou a carta mais alta na rodada anterior.  Note que rodadas podem ser consideradas empatadas (vide explicação do valor das cartas a seguir). Isso acaba desencadeando algumas exceções à regra supracitada. Se a primeira rodada empata, a dupla que vence a segunda é declarada vencedora da mão. Se as duas primeiras rodadas empatam, vence a mão a equipe vitoriosa na terceira rodada. Se todas as rodadas empatarem, a mão é considerada um empate e nenhuma das duplas pontua. Finalmente, se houver um vencedor da primeira rodada e um empate na segunda ou terceira, vence a mão a dupla que ganhou a primeira rodada. Em cado de empate em uma rodada, a rodada seguinte será iniciada pelo último jogador a jogar a carta de valor mais alto.  Nesse conjunto de regras, consideraremos a seguinte ordem (crescente) de valor das cartas: 4, 5, 6, 7, 8, 9, 10, Dama, Valete, Rei, Ás, 2, 3. Em outras palavras, o 4 é a carta de valor mais baixo, enquanto o 3 é a mais alta. Há, no entanto, uma exceção para essa ordem. Como explicado anteriormente, a cada mão sorteia-se uma carta chamada de vira. Naquela mão, as cartas de valor subsequente ao do vira recebem o nome de **manilha**. Por exemplo, se o vira é um valete, as manilhas são os reis de cada um dos quatro naipes. **As manilhas sempre são as cartas de mais alto valor dentro de uma mão. Para cartas normais, o naipe é irrelevante (*i.e.*, todas possuem o mesmo valor), mas para as manilhas há a seguinte ordem crescente de valor: ouros, espadas, copas e paus.** Isso significa que nunca é possível um empate em uma rodada na qual tenha sido jogada uma manilha.  A última regra que consideraremos para efeito desse trabalho é a do truco. Como dito no início dessa explicação, cada mão tem valor inicial de 1 ponto. No entanto, qualquer jogador durante a sua vez (e antes de jogar sua carta) pode **pedir truco**. O truco é basicamente um aumento da aposta, elevando o valor da mão. Nesse instante, a dupla adversária tem as seguintes três opções:   * Aceitar o truco, fazendo com que a mão continue a ser jogada normalmente, mas com seu valor aumentado. * Fugir, fazendo com que a mão seja imediatamente interrompida com vitória da dupla que solicitou o truco (a dupla que solicitou o truco recebe a pontuação anterior da rodada, *i.e.*, sem o truco). * Retrucar, aceitando o truco proposto e propondo aumentar ainda mais o valor da mão.   No caso de um retruco, a decisão passa a ser da dupla que originalmente solicitou o truco que terá as mesmas três opções supracitadas. Esse processo de retruco pode levar o valor da mão de 1 para 3, 6, 9 e 12 pontos. Não é possível retrucar uma mão que já está em 12 pontos. Finalmente, note que depois de uma situação de truco aceito a mão continua transcorrendo normalmente e, caso a mão ainda esteja valendo menos que 12 pontos, jogadores podem solicitar um novo truco para elevar ainda mais a pontuação. No entanto, um jogador só pode solicitar truco se o jogo ainda não está trucado ou se o último pedido de truco foi feito pela dupla adversária.  O Lado Servidor  Quando inicialmente executado, o servidor deve criar um ***socket* TCP para recepção de conexões de clientes**. O servidor deve realizar um *bind* desse *socket*à porta especificada pelo usuário no momento da inicialização do servidor.  Uma vez inicializado o *socket*de recepção, o servidor deve aguardar conexões dos clientes. Como já explicado, uma partida necessita de exatamente 4 jogadores. Portanto, o servidor deve aguardar até que a quarta conexão seja estabelecida. Neste momento, o servidor deve parar (temporariamente) de aceitar novas conexões e iniciar a partida.  O servidor é responsável por gerenciar toda a lógica do jogo. Ele deve manipular o embaralhamento das cartas, e a distribuição das mesmas pelos jogadores (*i.e.*, o envio pelo *socket* correspondente da informação de quais cartas foram dadas a cada jogador). O servidor também é responsável pelo sorteio da carta usada como vira a cada nova mão. Como essa carta deve ser conhecida por todos os jogadores, ao sortear um vira, o servidor deve enviar a informação dessa carta para todos os quatro jogadores através dos seus *sockets*.  O servidor é também responsável por informar a cada cliente quando é a sua vez de jogar. Nesse caso, o servidor deverá aguardar a resposta daquele cliente específico que pode ser uma das seguintes ações:   * Jogar uma determinada carta aberta. * Jogar uma determinada carta fechada. * Pedir truco.   Nos dois primeiros casos, o servidor deve repassar a informação aos demais jogadores. No segundo caso, o servidor deve verificar se aquele jogador pode, de fato, pedir truco naquele momento (*i.e.*, se a mão já não vale 12 pontos e se o último truco pedido naquela mão não foi pela mesma dupla). Caso o truco seja impossível, o servidor deve enviar uma mensagem avisando o cliente correspondente sobre essa impossibilidade e deve voltar a aguardar uma ação desse cliente.  Por outro lado, se o pedido de truco for possível naquele momento, o servidor deve enviar mensagens aos demais clientes avisando sobre esse evento. O servidor deve, então, aguardar por respostas de**ambos os clientes correspondentes aos jogares da outra dupla.**A resposta de cada um desses jogadores deverá conter uma das seguintes ações:   * Fugir. * Aceitar o truco. * Retrucar.   Se ao menos um dos jogadores responder com "fugir", o servidor deverá encerrar a mão e declarar a dupla que pediu o truco como vencedora. Caso contrário, se ao menos um dos jogadores responder com "aceitar o truco", o servidor deverá considerar o truco aceito, avisar todos os jogadores sobre isso e aguardar que o jogador que tem a vez jogue sua carta. Por fim, se ambos os jogadores responderem com "retrucar", o servidor deverá executar as mesmas ações relativas ao pedido de truco normal.  Ao final de cada rodada, o servidor deverá determinar a dupla vencedora da rodada (ou se houve empate). O servidor deverá ainda determinar se houve (e quem foi o) vencedor da mão. Terminada uma mão, o servidor deverá atualizar o placar da partida. De toda maneira, estes resultados deverão ser enviados a todos os clientes para serem exibidos aos jogadores.  Ao final de uma partida, o servidor deverá enviar uma mensagem para cada cliente avisando sobre o final do jogo. **O servidor deverá, então, voltar ao estado inicial que em aguarda por conexões para o estabelecimento de uma nova partida.**  O Lado Cliente  O lado cliente é mais simples que o servidor, servindo basicamente como uma interface entre o usuário humano e o servidor. **O cliente deverá receber como entrada um endereço ip (ou nome de *host*) e um número de porta**. Este par corresponde ao endereço no qual o servidor se encontra. O cliente deve, então, abrir uma conexão TCP com o servidor.  Uma vez aberta a conexão, o cliente deverá esperar mensagens do servidor e reagir de acordo. **Enquanto o servidor aguarda completar quatro jogadores, o cliente deverá exibir uma mensagem na tela informando o jogador dessa situação.**  Conforme descrito na seção anterior, **a maior parte das mensagens diz respeito simplesmente a informações a serem exibidas para o jogador**(*e.g.*, quais cartas foram dadas, qual carta acabou de ser jogada por outro jogador). As duas principais exceções são:   * Mensagem informando que agora é a vez do jogador. * Mensagem informando que a dupla adversária solicitou truco.   No primeiro caso, o cliente deve aguardar a ação do jogador. Essa ação poderá ser uma dentre as seguintes três:   * Jogar uma das cartas fechada. * Jogar uma das cartas aberta. * Pedir truco.   **A interface fornecida ao jogador deverá permitir essas três ações**. Note que não é responsabilidade do lado cliente determinar se o jogador pode ou não pedir truco nesse momento: isso é uma atribuição do servidor (vide seção anterior). O cliente deve apenas enviar a mensagem ao servidor que testará as condições adequadas e responderá com um erro (que deve ser exibido ao jogador) caso o truco não seja possível. Nesse último caso, o lado cliente mostrará novamente as opções ao jogador.  Já no segundo caso, o lado cliente deve informar ao jogador sobre o pedido de truco e aguardar que o jogador selecione uma das seguintes três ações:   * Fugir. * Aceitar. * Retrucar.   Novamente, é de responsabilidade do cliente fornecer uma interface que permita a seleção destas opções, mas não é responsabilidade do cliente verificar se é possível retrucar nesse momento.  Relatório  Como parte deste trabalho, cada grupo deverá entregar também um pequeno relatório documentando sua implementação. São itens obrigatórios desse relatório:   * Instruções de compilação e uso. * Uma descrição geral dos módulos que compõem a implementação (*e.g.*, classes, arquivos fonte). * Uma descrição do protocolo de comunicação usado (*e.g.*, quais são as mensagens definidas pelo protocolo, qual é o formato dessas mensagens).   Restrições  A implementação do trabalho pode ser feita em qualquer uma das seguintes linguagens: C, C++, C#, Java, Python, Pascal. Devem ser observadas as seguintes restrições adicionais:   * todas as funcionalidades citadas nesta especificação devem ser implementadas por código desenvolvido pelos próprios membros do grupo (*i.e.*, não serão aceitos códigos copiados total ou parcialmente de terceiros e funcionalidades como a manipulação dos *sockets* e manipulação das mensagens devem ser originais do grupo); e * o código final deve ser compilável/executável sem a necessidade de bibliotecas, *frameworks*ou ferramentas pagas.   De maneira análoga, o relatório é de formato livre, sem limites inferiores ou superiores de páginas. Apenas como um guia geral, 3 páginas devem ser suficientes (embora não necessárias) para contemplar todos os itens listados na seção anterior. O relatório deve ser entregue preferencialmente em formato pdf.  Os grupos poderão ser formados com no mínimo 2 e no máximo 5 alunos.  Requisitos e Critério de Avaliação  Cada trabalho receberá uma nota variando de 0 a 10. Esta pontuação será dividida nos seguintes requisitos:   1. Até 3,0 pontos para a implementação do cliente, dos quais:    1. Até 0,5 pontos para a existência e qualidade de comentários ao longo do código-fonte.    2. Até 0,5 pontos para a leitura dos parâmetros de entrada (endereço e porta do servidor).    3. Até 0,5 pontos para a correta manipulação do *socket*.    4. Até 0,5 pontos para a correta manipulação (geração e leitura) das mensagens do protocolo.    5. Até 0,5 pontos para a correta interação com o jogador.    6. Até 0,5 pontos para outros aspectos relevantes de implementação. 2. Até 4,0 pontos para a implementação do servidor, dos quais:    1. Até 0,5 pontos para a existência e qualidade de comentários ao longo do código-fonte.    2. Até 0,5 pontos para a leitura dos parâmetros de entrada (porta na qual o servidor fará o *bind*).    3. Até 1,0 ponto para a correta manipulação do *socket*.    4. Até 1,0 ponto para a correta manipulação (geração e leitura) das mensagens do protocolo.    5. Até 0,5 pontos para a implementação correta da lógica do jogo.    6. Até 0,5 pontos para outros aspectos relevantes da implementação. 3. Até 3,0 pontos para o relatório, dos quais:    1. Até 0,5 pontos para as informações básicas sobre a implementação.    2. Até 0,5 pontos para a visão geral do código.    3. Até 2,0 pontos para a descrição do protocolo criado.   As implementações serão avaliadas apenas em relação à correção do código e aderência a esta especificação. Não haverá pontuação associada ao desempenho/eficiência da implementação.  Note também que não há exigência de qualquer tipo de interface gráfica. Uma simples interface texto que exiba as informações e permita a interação do jogador é suficiente. Não haverá pontuação (obrigatória ou opcional) associada à interface.  Entrega  A data limite para a entrega do trabalho está disponível no calendário da página da disciplina. A entrega deverá ser realizada por e-mail, através do endereço dpassos@ic.uff.br. O e-mail deverá conter:   * Identificador do trabalho (*e.g.*, “Primeiro Trabalho de Redes I”). * Lista dos integrantes do grupo. * Código fonte da implementação. * Relatório.   Serão aceitos, sem penalidade, e-mails enviados até as 19:59 da data limite (*i.e.*, até o horário da aula). Os e-mails de entrega de trabalho terão seus recebimentos devidamente confirmados. **É responsabilidade do grupo garantir que o trabalho seja recebido, aguardando pela confirmação e reenviando a mensagem caso não a recebam em tempo razoável.** Em caso de dúvidas ou correções relacionadas a esta especificação, também é responsabilidade de cada grupo entrar em contato (seja pessoalmente, ou através do mesmo endereço de e-mail) requisitando esclarecimentos dentro do prazo de entrega do trabalho.  Uma vez entregue o trabalho, não **serão aceitas alterações** (nem inclusões, nem remoções) na lista de integrantes do grupo **em nenhuma hipótese**. Por isso, sugere-se atenção no momento do envio da mensagem para que a lista contenha todos os integrantes do grupo. |