

Manual de Uso do Sistema ServidorSimples

versão beta 1.0 – 17/11/2008

Luciano Antonio Digiampietri

1. Descrição:

Este sistema contém um gerenciador de jogos de tabuleiro desenvolvido em C#, incluindo as classes de comunicação. O sistema é composto por dois “pacotes”: *manager* e *netGameManager*. Enquanto o pacote *manager* contém as classes para a criação e gerenciamento dos jogos de tabuleiro o pacote *netGameManager* contém as classes responsáveis pela comunicação web e por criar as instâncias do servidor de jogos e dos jogos propriamente ditos. A Figura 1 contém o diagrama de classes deste sistema. A seguir, há uma breve descrição de cada uma das classes do sistema.

Program: classe principal que instancia e inicia um servidor de jogos (*GameServer*), no endereço e porta passados pelo usuário. Fica ativo enquanto o usuário não digitar *exit*, *halt*, *quit*, ou *stop*. Exibe na tela o log das mensagens recebidas e enviadas pelo servidor.

As seguintes classes pertencem ao “pacote” *netGameServer*, responsável por gerenciar um servidor de jogos web.

GameServer: classe responsável por gerenciar a lista de jogos ativos e por tratar e responder as mensagens enviadas utilizando-se os protocolos TCP ou UDP.

TCPServer: classe responsável por ouvir a uma porta de comunicação utilizando o protocolo TCP para receber mensagens das aplicações clientes, encaminhá-las ao servidor de jogos (*GameServer*) e responder às mensagens com a resposta produzida pelo *GameServer*.

UDPServer: classe responsável por ouvir a uma porta de comunicação utilizando o protocolo UDP para receber mensagens das aplicações clientes, encaminhá-las ao servidor de jogos (*GameServer*) e responder às mensagens com a resposta produzida pelo *GameServer*.

AddressPort: classe responsável por armazenar duas informações: endereço IP e porta de comunicação.

AnswerMessage: classe que contém as constantes (textos) utilizadas em cada mensagem XML das respostas dadas pelo servidor de jogos. Além disso, contém os métodos básicos para a geração das mensagens.

RequestMessage: classe responsável por traduzir (e estruturar) as requisições enviadas ao servidor por aplicações clientes, de forma a facilitar o acesso às informações recebidas.

Board: classe responsável por gerenciar o tabuleiro de um dado jogo. Não conhece as regras dos jogos e, assim, permite apenas operações básicas sobre tabuleiros (inserir peça, remover peça, etc).

Position: classe para armazenar um par de coordenadas (linha e coluna).

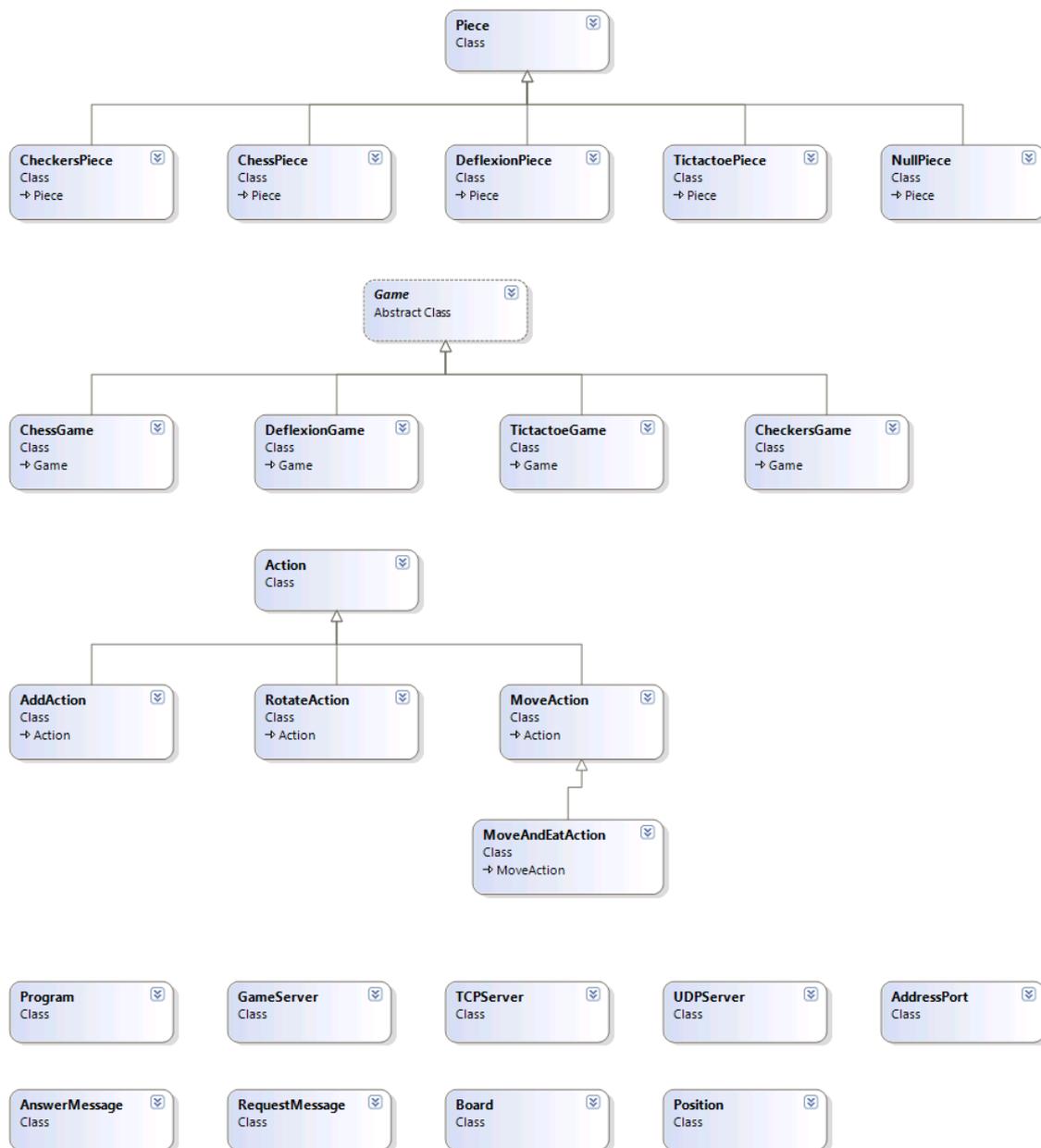


Figura 1 - Diagrama de classes do sistema: ServidorSimple

As seguintes classes pertencem ao “pacote” manager, responsável por conter as regras do jogo. Dentro deste pacote temos três super classes: *Piece*, *Game* e *Action*. A classe *Piece* e suas subclasses são responsáveis por gerenciar as peças dos diversos jogos. A classe *Game* e suas subclasses são responsáveis por gerenciar cada tipo de jogo (jogadas válidas, estado atual, etc). Classe *Action* e suas sub-classes são responsáveis por armazenar uma ação tomada pelo usuário.

Piece: super classe para representar uma peça de um dado jogo.

CheckersPiece: classe responsável por gerenciar peças do jogo de damas.

ChessPiece: classe responsável por gerenciar peças do jogo de xadrez.

DeflexionPiece: classe responsável por gerenciar peças do jogo de deflexion. Uma curiosidade desta classe é que as peças de deflexion têm orientação.

TictactoePiece: classe responsável por gerenciar peças do jogo da velha.

NullPiece: classe responsável por gerenciar peças nulas (casas em branco do tabuleiro).

Game: super classe para representar um jogo de tabuleiro. Um jogo contém um tabuleiro (que contém as peças) e um conjunto de métodos padrões (regidos pelas regras do jogo), tais como: adicionar peça, mover peça, etc.

CheckersGame: classe responsável por gerenciar um jogo de damas. As regras implementadas são as seguintes:

- A pedra movimenta-se em diagonal, sobre as casas escuras, para a frente, e uma casa de cada vez.
- A pedra que atingir a oitava casa adversária, será promovida a "dama", única peça que poderá andar para frente e para trás e comer para frente e para trás; porém apenas UMA casa de cada vez (assim como a peça normal).
- A tomada NÃO é obrigatória.
- A peça que toma poderá tomar mais de uma peça num mesmo movimento ou parar quando após a primeira tomada.
- Um jogador ganha o jogo quando tomar (comer) todas as peças do adversário ou quando o adversário não tiver mais nenhum movimento.
- Não foram estabelecidos limites para repetições de uma mesma jogada ou de um conjunto de jogadas.

ChessGame: classe responsável por gerenciar um jogo de damas. As regras implementadas são as seguintes:

- Não foram estabelecidos limites para repetições de uma mesma jogada ou de um conjunto de jogadas.
- A promoção de peão está ocorrendo de forma automática para rainha.

DeflexionGame: classe responsável por gerenciar uma instância do jogo deflexion (também conhecido como The World of Khet) <http://www.khet.com/>

TictactoeGame: classe responsável por gerenciar um jogo da velha.

Action: super classe para representar uma ação tomada pelo usuário. Alguns jogos precisam armazenar a última ação tomada pelo usuário para avaliar se a próxima ação será ou não válida.

AddAction: classe responsável por representar a adição de uma peça feita pelo usuário.

RotationAction: classe responsável por representar a rotação de uma peça feita pelo usuário.

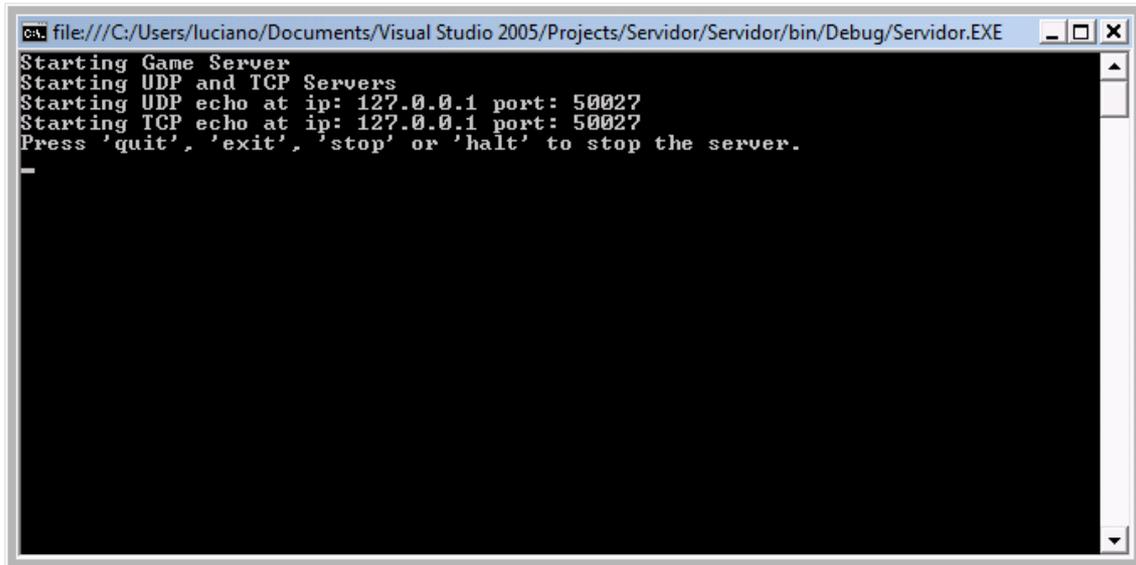
MoveAction: classe responsável por representar o movimento de uma peça feita pelo usuário.

MoveAndEatAction: classe responsável por representar o movimento de uma peça feita pelo usuário que causou a eliminação de uma peça (uma peça "comeu" a outra).

2. Modo de Uso:

Execute a classe GameServer passando como parâmetro a porta (TCP/UDP) a qual o servidor ficará escutando as mensagens de aplicações clientes para a criação e gerenciamento de jogos.

A porta padrão é a 50027. A Figura 2 apresenta uma cópia de tela da inicialização do GameServer. A Figura 3 apresenta uma cópia de tela das mensagens de log exibidas após o servidor receber e enviar algumas mensagens (no caso específico, mensagens da interação com um cliente que está jogando damas).



```
file:///C:/Users/Luciano/Documents/Visual Studio 2005/Projects/Servidor/Servidor/bin/Debug/Servidor.EXE
Starting Game Server
Starting UDP and TCP Servers
Starting UDP echo at ip: 127.0.0.1 port: 50027
Starting TCP echo at ip: 127.0.0.1 port: 50027
Press 'quit', 'exit', 'stop' or 'halt' to stop the server.
_
```

Figura 2 – Cópia de Tela da Inicialização da classe GameServer

