Classic Tetris Multiplayer

Felipe C. Gruendemann, Juan Burtet

[1]Laboratory of Ubiquitous and Parallel Systems – UFPEL
Pelotas, RS - Brasil

{fcgruendemann,jburtet}@inf.ufpel.edu.br

Resumo. *O jogo Tetris é um dos jogos mais conhecidos, difundidos e reinventados da plataforma NES ao longo dos anos e que ainda promove torneios pelo mundo. Neste trabalho é apresentada a proposta de uma aplicação cliente-servidor multiplayer do jogo Tetris baseada no jogo clássico de NES. Foram definidos protocolo da camada de aplicação e as mensagens do cliente e do servidor para a aplicação proposta. Foi escolhida a linguagem Python e o protocolo da camada de transporte TCP para implementação.*

1. Introdução

Tetris é um jogo eletrônico desenvolvido por Alexey Pajitnov, Dmitry Pavlovsky e Vadim Gerasimov lançado em Junho de 1984. O jogo consiste em empilhar tetraminós (formas geométricas compostas por quatro quadrados idênticos), que descem na tela com objetivo de completar linhas horizontais. Existem sete tipos de tetraminós no Tetris. Da esquerda para direita, eles são nomeados de I, O, T, S, Z, J e L (Figura 1). No jogo, quando uma linha é preenchida a mesma é apagada e as camadas superiores com blocos descem na tela, dessa maneira o jogador incrementa seus pontos. Quando a pilha de peças atinge topo da tela, a partida é encerrada.

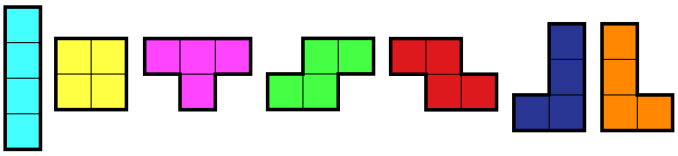

Figura 1. Peças do Jogo Tetris

A aplicação proposta tem como objetivo proporcionar uma experiência multijogador que seja a mais fiel possível à fornecida pela versão original do jogo Tetris da plataforma NES (*Nintendo Entertainment System*) de 1989. Na versão do NES - considerada como a versão definitiva do jogo - existe um sistema de pontuação simples, onde a maior quantidade de linhas completadas em um único movimento garante a maior quantidade de pontos para o jogador. Além disso, o sistema de nível indica a velocidade de queda das peças e também aumenta a quantidade de pontos recebidos. Dessa forma principal diferença entre a versão original e a proposta neste trabalho é a possibilidade de jogar na modalidade *multiplayer*.

A Figura 2 ilustra a interface na qual o projeto foi baseado, sendo recriadas a indicação da quantidade de linhas eliminadas pelo jogador, a sua pontuação atual, a próxima peça a ser recebida e o nível atual do jogo.

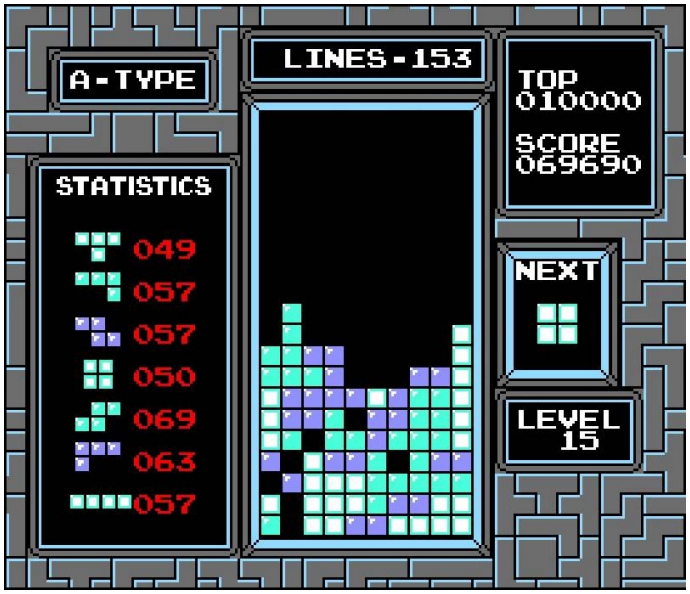

Figura 2. Imagem do jogo Tetris para NES.

Como o jogo original não possui uma versão *Multiplayer* disponível, foi usado como inspiração o modelo utilizado pelo Campeonato Mundial de Tetris Clássico. Neste evento, é iniciado um jogo simultaneamente entre dois jogadores com a mesma sequência de peças para ambos, onde os jogadores devem fazer a maior pontuação possível. Caso um jogador perca a partida enquanto tiver uma pontuação superior ao seu adversário, este deve continuar jogando até fazer uma pontuação superior ou será considerado derrotado. Na Figura 3 é apresentada a interface utilizada no campeonato mundial.

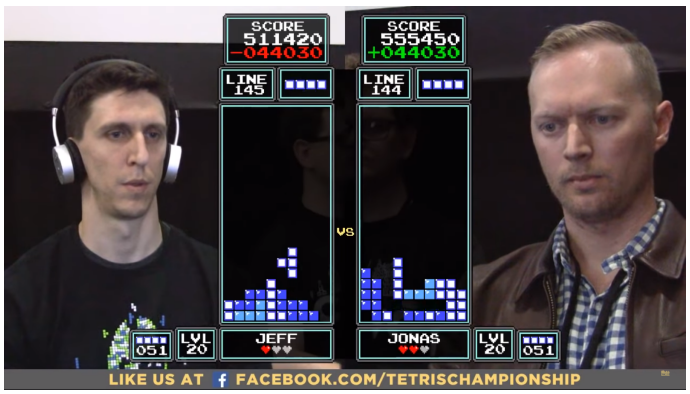

Figura 3. Campeonato Mundial de Tetris Clássico (Edição de 2016)

2. Manual de uso da aplicação

A linguagem de desenvolvimento escolhida para a aplicação foi Python3. As principais bibliotecas utilizadas foram *pygame*, *socket*, *_thread*. Para jogar uma partida, o jogador deve possuir um *nickname* e solicitar conexão ao *host* informando o endereço e a porta do servidor. Após estabelecida a conexão, o jogador só precisa esperar que outro jogador se conecte também e assim iniciar uma partida.

Ao iniciar uma partida, os dois jogadores jogam um contra o outro, até algum dos jogadores vencer 3 rounds. Para o jogador vencer um round, deve ser necessário fazer mais pontos que o seu adversário. Existe 2 possibilidades quando um dos jogadores perder a partida, ou seja, deixar a pilha de peças chegar até o topo da tela:

1. Se o jogador que perdeu a partida tinha menos pontos que o adversário, o adversário é considerado o vencer do *Round*.
2. Se o jogador que perdeu a partida tinha mais pontos que o adversário, o adversário continua jogando até ultrapassar os pontos do jogador que perdeu a partida.

Durante a partida, toda a informação do jogo é compartilhada com o seu adversário, onde é possível ver a pontuação de ambos os jogadores,

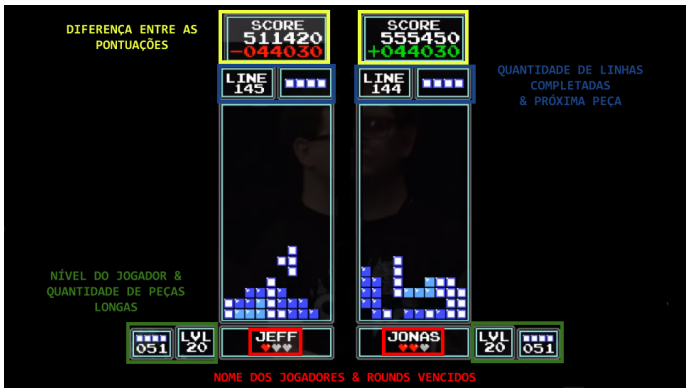

Figura 4. Imagem da partida multiplayer com informações auxiliares.

Durante uma partida todos os comandos durante a rodada aplicam movimentações as peças, sendo possível: movimentá-las para os lados, acelerar a queda e rotacionar.

3. Protocolo da Camada de Aplicação

O protocolo da aplicação é divido em duas partes: cliente e servidor. O servidor dá suporte para que os clientes se conectem e formem salas de dois jogadores. Os clientes podem se conectar ao servidor e acompanhar o jogo do adversário até que a partida seja finalizada, e também ver o ganhador - aquele que marcar a maior pontuação na partida.

3.1. Cliente

A aplicação que roda no lado do cliente é responsável pela execução do jogo, processamento das entradas do cliente, pelo envio dos dados do cliente e recebimento e processamento dos dados do adversário durante uma partida. Além disso o cliente sempre realiza um *ping* no servidor, para manter a conexão ativa.

Primeiramente o cliente seta seu *nickname* e depois solicita a conexão com o servidor enviando seus dados. Após isso o jogador fica esperando até que o servidor envie uma conexão com o jogador 2.

Com os 2 jogadores conectados ao servidor, o cliente dá início ao jogo. Durante a partida o cliente envia constantemente seus dados para o servidor e recebe os dados do adversário para poder exibi-los na tela.

Caso o cliente finalize o seu *round*, o seu adversário ainda pode continuar jogando com a intenção de ultrapassar sua pontuação, mas caso o adversário finalize o *round* o cliente em questão também pode continuar aumentando sua pontuação. Quando ambos jogadores finalizarem um *round*, o jogador vencedor tem sua quantidade *rounds* vencidos incrementado. Se a quantidade de *rounds* vencidos de um jogador chegar a 3, a partida é finalizada e ele é considerado o vencedor, finalizando a conexão. O protocolo cliente é ilustrado com algumas simplificações na Figura 5.

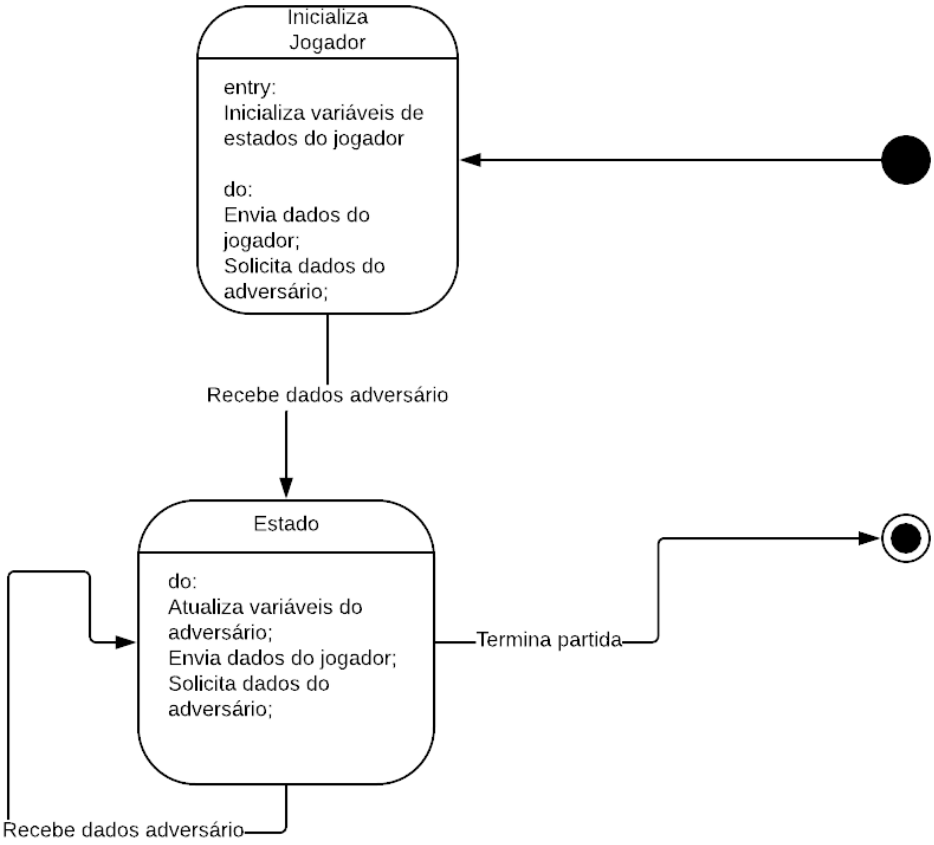

Figura 5. Diagrama - Máquina de Estados do protocolo Cliente.

3.2. Servidor

A aplicação do servidor é responsável por iniciar as partidas, receber/enviar dados de/para jogadores adversários. O servidor espera que pares de jogadores efetuem a conexão para poder iniciar uma partida. Durante a partida o servidor apenas redireciona os dados que recebe dos clientes para seus adversários. A Figura 6 ilustra o diagrama que representa o protocolo servidor.

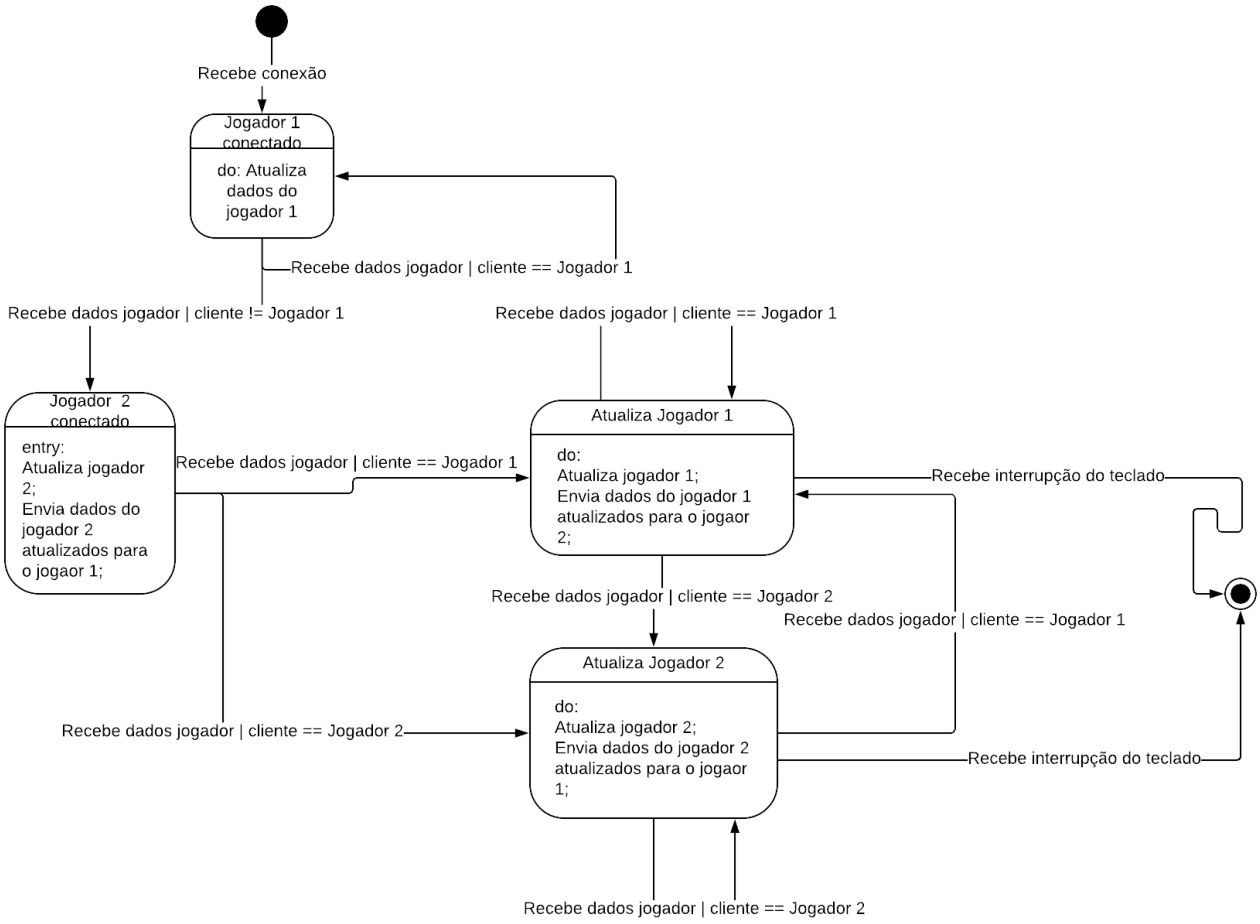

Figura 6. Diagrama - Máquina de Estados do protocolo Servidor.

3.3. Mensagens

A fim de se simplificar a arquitetura do projeto adotou-se a estratégia de usar apenas um formato de mensagem contendo os dados relevantes de cada cliente e apartir desses dados, são extraídas as variáveis de estado da aplicação:

- `[nickname][score][level][lines][fallingPiece][nextPiece][board][wins][state]`

  Valores possíveis para os parâmetros das mensagens:

1. `[nickname]` - Nome do jogador;
2. `[score]` - Pontuação do jogador no *round* atual;
3. `[level]` - Nível do jogador no *round* atual;
4. `[lines]` - Quantidade de linhas eliminadas pelo jogador no *round* atual;
5. `[fallingPiece]` - Peça atual caindo no campo;
6. `[nextPiece]` - Peça seguinte do jogador;
7. `[board]` - Estado Atual do Tabuleiro;
8. `[wins]` - Quantidade de *rounds* vencidos;
9. `[state]` - *Boolean* indicando se o jogador está "vivo"no round;

4. Conclusão

Mensagem	Parâmetros	Aplicação
Dados	*nickname; score; level; lines; fallingPiece; nextPiece; board; wins; state;*	Cliente informa a situação do seu jogo a cada loop ao servidor. Servidor envia os dados de um jogador para o outro a cada loop.