

# Implementação do Tetris em Assembly MIPS Utilizando Simulador MARS

Jonathas Conceição  
Lucas Bretana

6 de julho de 2016

Data da Entrega: 08, Julho, 2016

## 1 Objetivo

Implementar o clássico jogo russo da década de 80, *Tetris*, em Mips Assembly utilizando o simulador MARS(*MIPS Assembler and Runtime Simulator*).

### 1.1 Definições

**Interface:** É proposto a implementação uma interface gráfica utilizando o Bitmap Display do MARS, possuindo em sua interface a area de jogo, bem como pontuação, linhas completadas e proxima peça.

**Menu:** Possuir um menu básico de interação com usuário permitindo começar uma nova partida ou sair do jogo.

**Movimentação:** Permitir a movimentação e rotação das peças por meio de uma leitura do teclado utilizando a ferramenta de *Memory-Mapped I/O* presente no MARS.

## 2 Recursos Utilizados

Para este projeto foram utilizados os seguintes recursos presentes no MARS:

- **Macros**, utilizado para simplificar a codificação do projeto.
- **Bitmap Display**, para interface gráfica.
- **Keyboard and Display MMIO Simulator**, para leitura das ações do usuário.

- **Pseudo Instruções**, para simplificar a codificação do projeto.
- **Tratamento de Exceções**, para gerar uma peça aleatória e para finalizar a simulação.

### 3 Metodologia

Primeiramente foi desenvolvido uma série de macros relacionados a interface gráfica para simplificar o desenho da interface bem como das peças na tela e a aritmética para movimentação dos ponteiros.

Fez-se necessário a implementação de uma outra estrutura de dados além da pilha, uma fila para armazenamento de dados para a movimentação das peças. Junto aos macros foram utilizados também sobrotinas(funções) mas deixar o código mais claro. As subrotinas possuem uma nomenclatura mais direta a cerca do que é feito no código, e esta por sua vez chama os macros necessários para sua execução.

A interface possui um tamanho de 512x512 pixels. Para simplificar o desenho e a movimentação do ponteiro foi convencionado uma divisão da tela de 32x32 blocos com 16x16 pixels cada.

### 4 Resultados

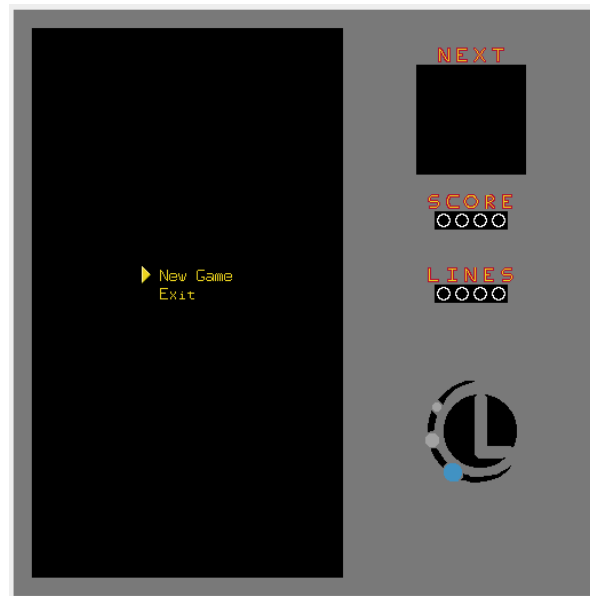


Figura 1: Interface inicial do jogo.

O jogo possui na sua interface(Figura 1) o espaço para o jogo contendo 272 pixels de largura. Uma caixa para exibir a próxima peça que será lançada na tela. Duas caixas pequenas para exibir a quantidade de linhas completadas e a pontuação do jogador.

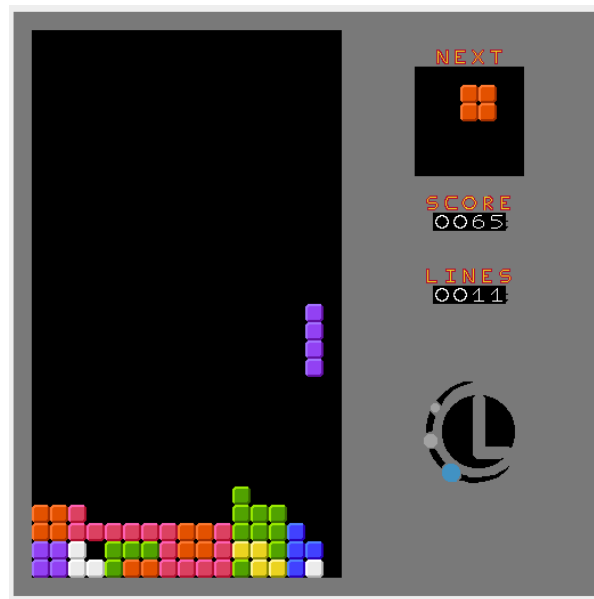


Figura 2: Jogo em execução.

As peças podem se mover livremente para esquerda, direita, baixo, e girar dentro do espaço da tela, desde que haja espaço livre. De tempo em tempo a peça descerá automaticamente, caso não seja possível descer a peça será fixada na tela, linhas completas serão removidas, a pontuação e contador de linhas atualizadas e uma nova peça lançada na tela(Figura 2).

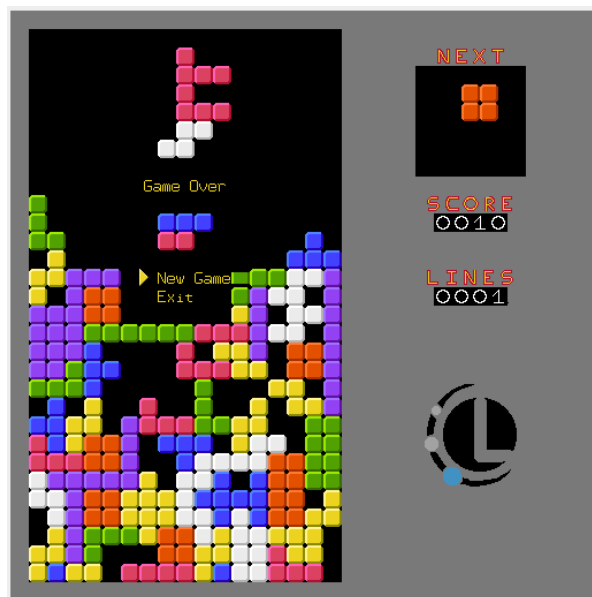


Figura 3: Tela de *Game Over*.

Caso uma nova peça não possa ser lançada em jogo, a mensagem de *Game Over* aparecerá juntamente ao Menu(Figura 3).

Para jogar o jogo é necessário iniciar MARS carregar o arquivo nomeado **Main.asm**.

Abrir a ferramenta *Keyboard and Display MMIO Simulator*. Abrir a ferramenta *Bitmap Display*, configurar para 512x512 com pixels 1x1, ou, 1024x1024 com pixels 2x2, selecionar endereço base para \$gp. Montar o código, conectar as ferramentas ao MARS e lançar a execução em velocidade máxima.

Curiosidades

- O jogo possui um *Ester Egg* que pode ser ativado no Menu inicial ao se digitar *J* ou *B*.
- As paletas de cores utilizadas no jogo pertencem ao conjunto de cores do SNES.
- O sistema de pontuação do jogo dá pontuações maiores quando a linha base é completada