

# Space Invaders em Assembly MIPS

Gabriel Vieira

Giovanni Guidini

Vitor Dullens

University of Brasília, Dept. of Computer Science, Brazil

## Abstract

Este trabalho apresenta o desenvolvimento de um famoso jogo de arcade, mais conhecido como Space Invaders [Wikipedia]. A proposta do jogo, em nossa versão remasterizada e única, se trata de matar seus inimigos antes que os mesmos cheguem até sua nave. Um projeto que aparenta ser simples, porém é supercomplexo e com desafios lógicos e computacionais bastante interessantes.

## 1 Introdução

A criação do Space Invaders foi um desafio e tanto, no aspecto da dificuldade de implementar um algoritmo eficiente. Apesar do jogo ter sido remasterizado, a plataforma(MARS) que utilizamos pode gerar algum tipo de bug ou falhas na execução. A linguagem de baixo nível que usamos para o desenvolvimento do programa, Assembly MIPS 32, apresentou certas limitações para o desenvolvimento do jogo, devido à seu conjunto de instruções de baixo nível. O que de tal forma foi muito bom, pois estimulou-nos a programar um jogo relativamente simples, fazendo nos esforçar muito para conseguir fazer o jogo de uma forma bem feita. Cada pequeno detalhe do programa feito por nós, foram colocados ali, para gerar uma melhor interação do usuário com o jogo em si, melhorando assim, a experiência do usuário com o jogo.

## 2 Metodologia

Nós desenvolvemos o jogo utilizando a linguagem Assembly MIPS32, e também usamos o simulador MARS, plataforma na qual desenvolvemos o nosso jogo, testamos e rodamos. O simulador MARS MIPS Assembler and Runtime Simulator [State] é um simulador desenvolvido pela Missouri State University. O mesmo foi desenvolvido com a função de ser um simulador e também um monitor para a execução de programas de linguagem Assembly MIPS.

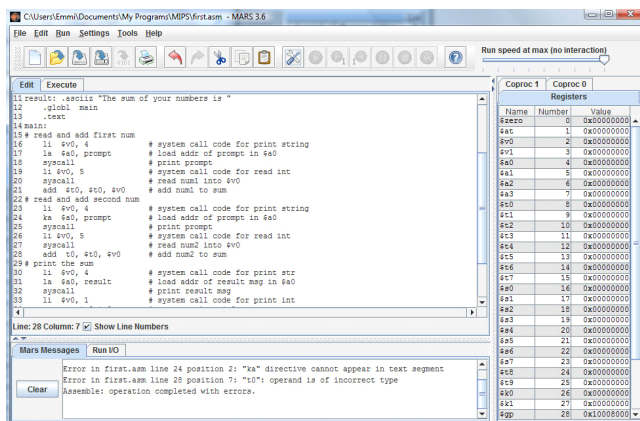


Figure 1: Editor de texto do MARS

A partir da Figura 1 é possível ver como é o MARS, com sua interface intuitiva que facilita o programador com alguns comandos e "tools"(canto superior esquerdo).

Esse simulador possui inúmeras ferramentas que auxiliam o programador, no que tange a simular um processador MIPS. Para tal auxílio, usamos duas ferramentas fundamentais do MARS, que são: Bitmap Display e o Keyboard and Display MMIO simulator.

O Bitmap Display, nos auxiliou no quesito de mostrar na tela uma imagem por meio de uma memória mapeada, onde cada pixel, equivale a uma quantia de bytes, a qual foi definida pelos programadores. Essa ferramenta é indispensável, devido ao fato dela sim-

plesmente mostrar o screen do jogo, as posições das respectivas naves, sua nave, e a HUD do jogo.

O Keyboard and Display MMIO simulator, é como um teclado do computador, só que na tela. Ele é usado para navegar pelo jogo usando-o como um teclado, e utilizando as teclas predefinidas pelos programadores para interagir com o jogo.

Usamos uma ferramenta muito útil também, cedido por um amigo nosso, que foi o Bmp2bin, que nos ajudou a converter as imagens de bitmap(.bmp) para binário(.bin) e coloca-las em nosso trabalho, de tal forma que, o MARS conseguisse lê-las.

Alem do mais, como já foi citado, foi utilizada a linguagem Assembly MIPS para a resolução do projeto, tal linguagem é considerada de baixo nível, devido a sua carencia de funções pré definidas, o modo como utilizar as funções e a sua proximidade com a linguagem de máquina(binário).

Na linguagem Assembly, existe uma correspondência entre o código programado e o resultado, ou seja, uma instrução do programa equivale a uma operação do computador. Com isso, ela se torna uma linguagem mais complexa e menos intuitiva de se trabalhar, aumentando ainda mais o desafio de fazer o projeto.

## 3 Resultados Obtidos

A partir da tentativa de montar uma versão de space invaders, atingimos resultados relativamente satisfatórios. É notavel certa lentidão durante a execução do programa e certos bugs que não foram possíveis de serem consertados. Contudo, utilizamos todos os conhecimentos adquiridos em sala, além de pesquisas em sites diversos para trazer a mais satisfatoria experiencia de um jogo, dentro do possível, levando em consideração o tempo limitado.

Quando inicializar o jogo no MARS, basta pressionar qualquer tecla, após fazer isso, o jogo iniciara, com as teclas "A" ou "D" é possível se movimentar para esquerda(A) e para direita(D), para atirar basta pressionar "Space". Caso digite qualquer outra coisa, não acontecerá nada com a nave.



Figure 2: Tela inicial

A Figura 2 representa a tela do início do Space Invaders, que ficará piscando "Enter" até o usuário pressionar qualquer tecla para dar início ao jogo. Nela é possível ver a mundaça que nós trouxemos a nossa versão, "Logic Gates Attack". Fizemos uma referência ao que foi aprendido em sala de aula sobre Circuitos Digitais.

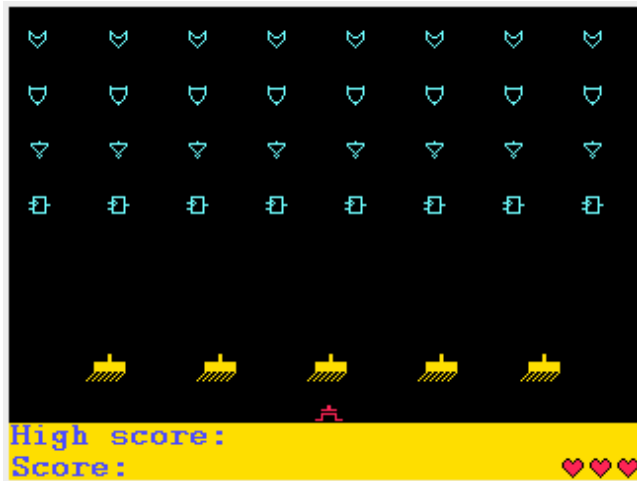


Figure 3: Tela do jogo

A partir da Figura 3 é possível ter uma noção melhor de como é o jogo. O personagem controlado pelo usuário, representado por um transístor, possui o objetivo de destruir o maior número possível de inimigos, representados pelas portas lógicas. Enquanto isso, o usuário deve evitar ser atingido pelas mesmas para não perder vidas (canto inferior direito). É possível, também, utilizar as barreiras, representadas pelos aterramentos, para se defender dos inimigos, porém elas podem atrapalhar, tendo em vista que os tiros não perfuram as mesmas.



Figure 4: Tela de Game Over

O jogador possui apenas 3 vidas e caso não haja nenhuma vida restante, o jogo se encerrará com a tela clássica de Game Over (Figura 4).

*No repositório do GitHub - [github.com/Gguidini/SpaceInvaders](https://github.com/Gguidini/SpaceInvaders) - estão todas as instruções para abrir, montar no MARS e rodar o jogo.*

## 4 Conclusão

Neste projeto realizado em nosso primeiro semestre foi uma experiência muito boa, para que possamos nos acostumar a desenvolver uma lógica de programação relativamente complexa, pois utilizamos uma linguagem de baixo nível como foi dito anteriormente. A utilização desta, foi fundamental para que entendêssemos os aspectos básicos da computação já que ela está muito próxima da linguagem de máquina e é muito importante entender o funcionamento de ambas para a realização do projeto.

Além da programação de baixo nível, houve a necessidade de aprender de como fazer a estrutura geral de um jogo relativamente

simples, mas que foi um desafio e tanto. Tal conhecimento pode ser aplicado em futuros projetos, relacionados a qualquer assunto, já que a lógica adquirida no processo não será perdida. Porém se relacionarmos a jogos a lógica implementada é mais relevante, pois utilizamos dela e dos conhecimentos que temos, para desenvolver o Space Invaders, como foi proposto.

## References

STATE, M. MARS - Mips Assembly and Runtime Simulator.

WIKIPEDIA. Space invaders.