



|   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 0 | # | # | # | # | # | # | # | # |
| 1 | # | # | # | # | # | # | # | # |
| 2 | # | # | # | # | # | # | # | # |
| 3 | # | # | # | # | # | # | # | # |
| 4 | # | # | # | # | # | # | # | # |
| 5 | # | # | # | # | # | # | # | # |
| 6 | # | # | # | # | # | # | # | # |
| 7 | # | # | # | # | # | # | # | # |

# MINESWEEPER GAME

11/0/2024



prof. Ramon Santos Nepomuceno  
UNIVERSIDADE FEDERAL DO CARIRI (UFCA) - CAMPUS JUAZEIRO DO NORTE

## Visão geral

O "Minesweeper" (campo minado em português) é um jogo de tabuleiro que desafia os jogadores a descobrir células sem bombas em um campo minado. Os números revelados indicam quantas bombas estão adjacentes a uma determinada célula. O objetivo é descobrir todas as células seguras sem acionar uma bomba.

### [Explicação do jogo](#)

Este projeto consiste em implementar uma versão em assembly MIPS do jogo minesweeper.

.

|   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 0 | # | # | # | # | # | # | # | # |
| 1 | # | # | # | # | # | # | # | # |
| 2 | # | # | # | # | # | # | # | # |
| 3 | # | # | # | # | # | # | # | # |
| 4 | # | # | # | # | # | # | # | # |
| 5 | # | # | # | # | # | # | # | # |
| 6 | # | # | # | # | # | # | # | # |
| 7 | # | # | # | # | # | # | # | # |

Tabuleiro do jogo.

.

## Especificações do Projeto

O projeto consiste em implementar o jogo Minesweeper em linguagem Assembly MIPS. Para a implementação do projeto, foi disponibilizado um repositório com o código base. O repositório está dividido em vários arquivos para facilitar a organização. Cada função do jogo está contida em um arquivo separado.

### Estrutura do Repositório

1. `main.asm`: Contém a função principal (`main`) que controla o fluxo do jogo em Assembly MIPS.
2. `printboard.asm`: Implementa a função para imprimir o tabuleiro.
3. `initializeboard.asm`: Implementa a função para inicializar o tabuleiro.
4. `plantbombs.asm`: Implementa a função para posicionar as bombas no tabuleiro.
5. `macros.asm`: Contém macros úteis para facilitar o desenvolvimento em MIPS.
6. `Mars.jar`: Executável do Mars MIPS, necessário para rodar os arquivos `.asm`.
7. `minesweeper.c`: Contém a implementação em C do jogo Minesweeper. Este arquivo serve como referência para a lógica do jogo e pode ser utilizado para comparação com as implementações em Assembly MIPS.
8. `play.asm`, `checkvictory.asm`, `revealcells.asm`: Arquivos em branco. Os alunos devem implementar essas funções em Assembly MIPS.

## Instruções de Execução

1. Abra o terminal na pasta do repositório.
2. Compile o código C usando um compilador padrão C, como o GCC, com o seguinte comando: `gcc minesweeper.c -o minesweeper`
3. Execute o executável gerado: `./minesweeper`
4. Siga as instruções no console para jogar o Minesweeper em C.

## Execução dos Arquivos em Assembly MIPS

1. Abra o terminal na pasta do repositório.
2. No linux, execute o Mars MIPS digitando o seguinte comando: `java -jar Mars.jar`
3. No windows, basta usar o duplo clique.
4. No Mars MIPS, abra cada arquivo .asm individualmente e monte/executa o código.
5. A saída do jogo será exibida na console do Mars MIPS.

## Observações

1. As macros no arquivo macros.asm foram criadas para simplificar o desenvolvimento e podem ser utilizadas conforme necessário.
2. Divirta-se jogando Minesweeper em Assembly MIPS!

## Dicas de Implementação

- UTILIZEM COMO BASE OS CÓDIGOS QUE FORAM ENTREGUES, PARA CADA FUNÇÃO DO CÓDIGO EM C, FAÇA SUA EQUIVALENTE EM MIPS.
- LEMBRE-SE DAS CONVENÇÕES DE USO DOS REGISTRADORES
- VOCÊ PODE UTILIZAR TODOS OS RECURSOS DO MARS, INCLUSIVE AS PSEUDO INSTRUÇÕES.
- NÃO DEIXE PARA ÚLTIMA HORA, O TRABALHO É DESAFIADOR.

## Entrega e Avaliação

- O trabalho poderá ser feito em dupla, mas a avaliação é **INDIVIDUAL**;
- Os que optarem por fazer o trabalho, a prova valerá 6 e o trabalho 6.
- A nota da prova **interfere** na nota do trabalho. Não faz sentido errar coisas básicas na prova, se tiver feito o trabalho.
- O grupo deverá apresentar o trabalho ao professor da disciplina semanalmente que, no ato, fará perguntas a todos os integrantes.
- Além da apresentação, o grupo deverá entregar um relatório explicando o funcionamento do circuito e os arquivos do projeto. Além disso, os integrantes deverão, de forma sigilosa, mandar uma mensagem privada ao professor informando o percentual de participação de cada componente do grupo.
- A data da entrega do projeto será divulgada no google sala de aula.
- **EM CASO DE PLÁGIO OU QUALQUER FRAUDE, TODOS OS GRUPOS ENVOLVIDOS TERÃO A NOTA ZERADA!!**