**UNIVERSIDADE FEDERAL DO CARIRI**

**BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

**CÍCERO IGOR ALVES TORQUATO DOS SANTOS**

**JOÃO MARCELO LOBO MATOS**

**RELATÓRIO MINESWEEPER**

**Juazeiro do Norte**

**2024**

**INTRODUÇÃO:**

O presente relatório descreve o processo de implementação do jogo Minesweeper em linguagem Assembly MIPS. O projeto foi desenvolvido com base em um código base fornecido pelo professor, organizado em diversos arquivos para facilitar a compreensão e manutenção do código. Cada funcionalidade do jogo foi encapsulada em um arquivo separado, visando uma organização eficiente e modular do projeto. Neste relatório, serão apresentados os desafios enfrentados, as estratégias adotadas para superá-los, e os resultados obtidos ao final do desenvolvimento.

**DESCRIÇÃO DO JOGO:**

O Minesweeper é um jogo de tabuleiro de lógica onde o objetivo principal é descobrir todas as células vazias sem revelar as minas ocultas. O tabuleiro típico do Minesweeper é uma matriz bidimensional, onde cada célula pode estar em um de dois estados: revelada ou oculta.

Alguns detalhes importantes:

* Cada célula pode conter uma mina ou estar vazia.
* O jogador pode revelar uma célula usando como entrada o número da linha e coluna.
* Se uma célula vazia for revelada, ela mostrará o número de minas adjacentes a ela.
* Se uma célula com uma mina for revelada, o jogo acaba.

O desafio do jogo está em usar a lógica para deduzir a localização das minas, com base nas dicas fornecidas pelas células reveladas. O objetivo final é revelar todas as células vazias sem clicar em uma única mina.

**DESENVOLVIMENTO:**

Como foi citado anteriormente, o projeto está dividido em vários arquivos para facilitar a organização. Os arquivos main, macros, play, plantBombs, initializeBoard e printBoard já tiveram seus códigos disponibilizados pelo profesor. Porém, os códigos dos arquivos play, checkVictory, countAdjacentBombs e revealNeighboringCells precisavam ser desenvolvidos pelos alunos. Vamos detalhar cada arquivo a seguir:

1. **Arquivo "main"**: Este arquivo é o ponto de entrada do programa. Ele inicializa o tabuleiro do jogo, plantando as minas em posições aleatórias e definindo o estado inicial de cada célula. Além disso, o arquivo "main" também contém a função principal do programa, que controla o fluxo de execução do jogo e chama as funções necessárias para interagir com o jogador e atualizar o estado do tabuleiro.
2. **Arquivo "macros"**: Neste arquivo, são definidas macros úteis que podem ser utilizadas em todo o projeto. Isso inclui macros para salvar e restaurar o contexto, que são dados que estavam nos registradores , e são utilizadas em quase todos os arquivos que implementam funções.
3. **Arquivo "initializeBoard"**: Este arquivo contém a função responsável por inicializar o tabuleiro do jogo. Isso inclui definir o tamanho do tabuleiro, a quantidade de minas a serem plantadas e o estado inicial de cada célula.
4. **Arquivo "plantBombs"**: Aqui, é implementada a lógica para plantar as minas em posições aleatórias no tabuleiro. Isso é feito garantindo que cada célula tenha uma chance igual de conter uma mina e que a quantidade de minas plantadas corresponda ao número especificado no início do jogo.
5. **Arquivo "printBoard"**: Este arquivo contém a função responsável por imprimir o estado atual do tabuleiro na tela. Isso inclui exibir as células reveladas, as minas são reveladas quando o jogador perde o jogo.
6. **Arquivo "play"**: Esta função representa a jogada do jogador. Se o jogador clicar em uma célula com uma bomba, o jogo termina e retorna 0 (game over).

Se o jogador clicar em uma célula vazia, a célula é marcada como revelada e, se não houver bombas adjacentes, as células adjacentes também são reveladas utilizando a função revealAdjacentCells. Retorna 1 se o jogo continuar.

1. **Arquivo “countAdjacentBombs”:** Esta função conta o número de bombas adjacentes a uma célula especificada por sua posição na matriz do tabuleiro. Ela percorre todas as células adjacentes à célula especificada, verificando se estão dentro dos limites do tabuleiro e se contêm uma bomba (representada pelo valor -1). O contador é incrementado sempre que uma célula adjacente contém uma bomba. O resultado final é o número total de bombas adjacentes à célula especificada.
2. **Arquivo “revealAdjacentCells”:** Esta função revela as células adjacentes a uma célula vazia (sem bombas adjacentes) especificada por sua posição na matriz do tabuleiro. Ela percorre todas as células adjacentes à célula especificada, verificando se estão dentro dos limites do tabuleiro e se são células vazias (representadas pelo valor -2). Se uma célula adjacente for vazia, ela é marcada como revelada e a função é chamada recursivamente para continuar a revelação das células adjacentes a essa célula.
3. **Arquivo “checkVictory”:** Esta função verifica se o jogador venceu o jogo.

Percorre todas as células do tabuleiro e conta o número de células que foram reveladas (ou seja, não contêm bombas). Retorna 1 se todas as células válidas do tabuleiro (ou seja, todas as células exceto as que contêm bombas) foram reveladas. Caso contrário, retorna 0.

**CONCLUSÃO:**

A implementação do jogo Minesweeper em linguagem Assembly MIPS representou um desafio significativo, mas também uma oportunidade valiosa para aplicar conceitos de programação de baixo nível e lógica de jogo. Ao longo do desenvolvimento, enfrentamos diversos obstáculos, desde compreender completamente a lógica do jogo até garantir a integração adequada das diversas funcionalidades em um código coeso e funcional.

A divisão do projeto em arquivos separados facilitou a organização e a manutenção do código, permitindo que cada funcionalidade fosse desenvolvida de forma modular e independente.O acompanhamento do projeto, por parte do profesor, em sala, foi de extrema importância pois dúvidas em pontos fundamentais do projeto foram esclarecidas.

No final, conseguimos implementar com sucesso todas as funcionalidades essenciais do jogo, incluindo a contagem de bombas adjacentes, a função de jogada do usuário, a função de revelar as células adjacentes e a verificação da vitória. O resultado final é um jogo Minesweeper funcional e divertido, pronto para ser apreciado por jogadores de todas as idades.

**Links para o GitHub:**

Código-fonte disponível em: https://github.com/IgorTorquatto/Arquitetura\_de\_Computadores/tree/main/Projeto/minesweeper