UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
INSTITUTO METRÓPOLE DIGITAL
INTRODUÇÃO ÀS TÉCNICAS DE PROGRAMAÇÃO
DOCENTE: FERNANDO MARQUES FIGUEIRA FILHO
RELATÓRIO TÉCNICO – CAMPO MINADO (VERSÃO TEXTO)

FLÁVIO HENRIQUE DE OLIVEIRA NEVES

1. Introdução:

O campo minado é um jogo que usa lógica e dedução para descobrir, em um "mapa" de linhas e colunas, onde estão as minas e depois desarmá-las. Ao selecionar uma célula desse mapa, algumas delas são reveladas. As células reveladas mostram números que indicam a quantidade de minas que existem nas oito células adjacentes a eles. Se o jogador seleciona para revelar uma célula onde há mina, ela explode e o jogo acaba com derrota. Através da dedução, o objetivo é descobrir e marcar onde estão todas as bombas sem detoná-las, ao mesmo tempo em que revela todas as outras células. Para a marcação, o jogador usa uma "bandeira" para marcar a célula.

Como o Campo Minado é um jogo criado para o sistema operacional Windows (que usa o cursor do mouse para percorrer as células, o clique do botão esquerdo para abrir a escolhida e o clique do botão direito para colocar a bandeira), o desafio nesse projeto é portá-lo para um sistema em linguagem C que use o CLI (Interface de Linha de Comando, em português) para selecionar as ações. Sendo assim o jogador precisará digitar comandos (números para a escolha de linha e coluna e caracteres para abrir ou colocar bandeira), que serão apresentados a ele como instrução antes de cada jogada.

Para a segunda parte do projeto, correspondente à Unidade 2 da disciplina Introdução às Técnicas de Programação, o objetivo principal é implementar o jogo propriamente dito, com as funcionalidades básicas necessárias ao funcionamento do mesmo. Para isso, foram adicionadas ao código base novas funcionalidades através de funções, como por exemplo: 'gerarMinas', colocando as minas no mapa de maneira randomizada; as 'lerJogada' e 'colocarBandeira', que permitem ao jogador escolher a ação que deseja executar (jogando o jogo, literalmente); e a 'revelarCelula', que revela a célula escolhida. As demais funcionalidades, além das que já foram aqui citadas, serão explicadas com mais detalhes no decorrer deste relatório.

2. Metodologia:

Para esse projeto, foi usado o Visual Studio Code como editor de código em linguagem C, o GCC para compilar o código e o GitHub como plataforma de hospedagem na internet de todos os arquivos e pastas que fazem parte do projeto.

O código do projeto foi feito inteiramente em um único arquivo, o ‘main.c’, e foi organizado da seguinte forma: todas as funções auxiliares foram dispostas antes da função principal, em ordem de evolução do código de acordo com as necessidades que foram surgindo. Primeiramente, vieram as funções de criar o mapa, depois a de mostrá-lo em tela. Depois a de gerar as minas, e assim sucessivamente.

Na função principal, estão inicializadas algumas variáveis como ‘lin’ e ‘col’, que são elas que dirão o tamanho do mapa, além da inicialização de todas as funções e os condicionais necessários para o funcionamento do jogo.

3. Análise do Código:

Como o jogo Campo Minado é feito basicamente usando arrays aninhados (ou matrizes), tentei implementar funcionalidades ‘auxiliares’ de maneira a abordar o máximo possível dos conhecimentos aprendidos no decorrer do curso. Strings, matrizes, repetições aninhadas e ponteiros são utilizados com função de deixar a experiência de jogo ainda mais interativa e semelhante à original.

Para a manipulação de strings, procurei utilizá-la nos menus. Desde a escolha da dificuldade até as opções de jogada, toda a interação do usuário com o jogo é feita com essa técnica. O nível de dificuldade é escolhido com base na referência de uma letra, as ações de abrir a célula (‘a’), colocar bandeira (‘b’), reiniciar o jogo (‘r’) e sair (‘s’). Vale citar que o programa não distingue maiúsculas de minúsculas, facilitando a interação.

As repetições aninhadas foram usadas na maioria das funções. As funções ‘inicializarJogo’, ‘mostrarJogo’, ‘colocarMinas’, ‘contarMinasAdjacentes’ são só alguns exemplos mas, como já citado neste relatório, esse tipo de estrutura é a base fundamental do jogo. A primeira interação do usuário é a escolha da dificuldade do jogo, que está implementada em três níveis: o nível básico contará com um mapa de tamanho 9 x 9 (9 linhas e 9 colunas) e 10 células serão minas; no nível intermediário, o mapa será de 16 x 16 (16 linhas e 16 colunas), onde haverá 40 minas; e no nível avançado, 24 x 24 (24 linhas e 24 colunas), com 99 minas. O jogador digitará 1, 2 ou 3 para escolher, respectivamente, uma das três opções. Foi criado uma repetição no formato “do while” dentro da função principal (main()) para tratar esse dado, ou seja, se o jogador selecionar uma opção diferente das apresentadas, uma mensagem o informará da escolha inexistente e o mesmo menu surgirá novamente, até que ele escolha uma das três opções

válidas. Nela também serão iniciadas as variáveis necessárias ao funcionamento geral, bem como todas as funções implementadas. Para essa parte final do projeto, se fez necessária a inclusão das bibliotecas <stdio.h>, para possibilitar a entrada e saída de dados entre programa e usuário; <stdlib.h> e <time.h>, para a geração randômica das minas no mapa.

Na sequência, será apresentado, no CLI, o mapa, de acordo com a escolha do jogador. Nessa etapa do projeto, cada célula será preenchida com o caractere "-", indicando que o mapa está "fechado". O próximo passo será o surgimento de um menu interativo para o usuário escolher como quer agir. Ele pode, a qualquer momento durante o jogo, escolher entre abrir a célula escolhida ('a'), colocar bandeira em uma célula escolhida ('b'), reiniciar o jogo ('r') ou sair ('s'). Se ele opta por reiniciar, a aplicação volta para o menu de escolha de dificuldade. Quando ele opta por sair, a aplicação simplesmente encerra. Se ele optar por abrir a célula ou marcar a bandeira, irá escolher um valor para a linha e um valor para a coluna. A opção pela bandeira fará o caractere 'B' aparecer, e indicará que ali o jogador indica como uma potencial mina, e essa célula não poderá ser revelada a não ser que ele escolha novamente a opção 'b' e a desmarque. Já a escolha por revelar a célula trará duas consequências: ou nela existirá uma mina e o jogo acaba com derrota do jogador, ou serão revelados os espaços adjacentes que não conterão minas nem os números que indicam a presença de outras minas ao redor deles. Essa maneira de revelação das minas 'automaticamente' é mais uma implementação de função auxiliar, a 'contarMinasAdjacentes'.

Na função 'lerJogada', foi necessária a utilização de ponteiros, pois é necessário que se modifique diretamente as variáveis da função main. O jogo precisa que a jogada que o usuário faz seja registrada na variável criada e utilizada na função principal. Então, com o uso do '&', o programa envia o endereço de memória das variáveis originais, possibilitando assim o correto funcionamento da aplicação como um todo. Sem isso, as jogadas do usuário não seriam refletidas corretamente no decorrer do jogo.

4. Dificuldades e Soluções:

As principais dificuldades técnicas encontradas na elaboração deste projeto passam pelo correto uso da lógica e estrutura de dados. O tabuleiro precisa ser corretamente representado, de maneira que seja fácil a identificação tanto das jogadas disponíveis quanto das que já foram feitas. Para a criação de minas aleatórias, foi tido o cuidado de não colocar duas delas no mesmo espaço, além de que seriam realmente aleatórias, com o número exato de acordo com a escolha do usuário. Também, o cálculo

das células adjacentes foi feito de maneira a não deixar as bordas da matriz serem afetadas, procurando não ultrapassar os limites de cada matriz. Da mesma forma, a revelação das células com nenhuma mina ao redor foi pensado para implementar a recursão correta e evitar loops infinitos. E, também, foram implementadas ferramentas para impedir ações inválidas, desde a escolha da dificuldade até as jogadas. A condição de derrota foi bem especificada, atualiza o tabuleiro corretamente e permite reinicialização e saída do jogo a qualquer momento.

5. Conclusão:

O desenvolvimento deste jogo me permitiu aplicar, de forma prática, conceitos importantes da linguagem, tais como: matrizes, funções, ponteiros e geração de números aleatórios. Durante a implementação, foi necessário estruturar o tabuleiro do jogo, gerar minas de forma aleatória e criar a lógica para revelar células, incluindo a abertura automática das áreas sem minas ao redor.

Alguns desafios foram especialmente marcantes, como evitar erros de acesso à matriz, calcular corretamente as minas adjacentes e implementar a recursão para abrir células vazias. O uso de ponteiros também foi essencial para que a função de leitura da jogada pudesse alterar diretamente as variáveis do programa principal.

No geral, o projeto atingiu o objetivo do qual se propõe, criando uma versão funcional do Campo Minado com diferentes níveis de dificuldade e interação objetiva pelo terminal. Além disso, o trabalho ajudou a reforçar o entendimento sobre lógica de programação, organização do código e o funcionamento de ponteiros em linguagem C.