



Licenciatura em Engenharia Informática

U. C. Programação 1

Professora Teresa Gonçalves e Professor Vítor Nogueira

“Minesweeper”

	A	B	C	D	E	F	G	H
1
2
3
4
5
6
7

Discentes:

André Baião, 48092

Gonçalo Barradas, 48402

Turma A

Janeiro, 2021



Índice

Índice

Introdução	3
Equipamento.....	4
Desenvolvimento	4
Variáveis do tabuleiro de jogo	4
Função “generate_board”	5
Função “print_board”	5
Função “win”	6
Função “game_over”	6
Função “uncover”	6
Função “flag”	6
Função “main”	7
Discussão de Resultados	9
Conclusão	10
Referencias Bibliográficas / Webgrafia	11



Introdução

Este trabalho, tem como objeto de estudo o minesweeper, mais concretamente a construção e desenvolvimento do seu código utilizando a linguagem c.

Este videojogo ficou conhecido na década de 80 e caracteriza-se como, um jogo de quebra-cabeças/puzzle. O objetivo deste jogo, é conseguir revelar todas as células sem revelar nenhuma mina. Quando é revelada uma mina o jogador perde automaticamente. Cada célula tem um valor que varia conforme o número de minas em redor da mesma, caso as células em redor da célula selecionada pelo jogador tiverem valor 0 (nenhuma mina em seu redor), as células em seu redor também serão reveladas.(Anon 2021)

Neste jogo o jogador tem duas opções: revelar e sinalizar células. A sinalização é identificada por uma bandeira.

O objetivo deste trabalho é avaliar e testar os nossos conhecimentos e domínio na linguagem c e, por sua vez, conseguir elaborar um código que preencha os requisitos solicitados pelos professores e descrever o percurso da concretização do mesmo.



Equipamento

Para a realização deste trabalho utilizámos a ferramenta *Virtual Studio Code* (VSC) em computadores com o sistema operativo *Linux*.

Desenvolvimento

A elaboração do código, teve por base todo o conhecimento adquirido durante as aulas dos docentes da unidade curricular de programação 1, e a consulta de um website (Belwariar 2018; Nogueira and Gonçalves 2021).

Variáveis do tabuleiro de jogo

De forma, a criar o nosso tabuleiro de jogo, começamos por definir as variáveis do mesmo:

- **board:**
Aqui é onde são guardados os valores de cada célula, ou seja, se é uma mina um número ou se está em branco.
- **Show_board:**
Variável utilizada para o imprimir o tabuleiro, que o utilizador vê durante o jogo, no início é composto por pontos que vão ser revelados durante o jogo.
- **nlinha e ncoluna:**
Aqui é onde se vai definir o tamanho do tabuleiro, isto é, onde são guardados o número de linhas e o número de colunas.
- **mines:**
Nesta variável, fica guardado o número de minas que o utilizador define. O número de mina é igual ao número de flags e fica guardado na variável “nflags”.



De seguida foram determinadas quais as funções que devem ser utilizadas para o desenvolvimento do jogo.

Função “generate_board”

A função “generate_board” determina o valor de cada célula:

- Esta função começa por determinar as coordenadas para as minas de maneira aleatória. Utilizando a função “rand” dentro da função “generate_board” é possível atribuir uma mina a cada célula diferente alternadas em cada jogo.
- De seguida analisa-se cada célula e conforme o número de minas á volta de uma célula, assim é determinado o seu valor. Por meio de um ciclo “for” com vários ciclos “if” determina-se o valor de cada célula com valor diferente de “-1” (valor definido para mina) dependendo da quantidade de minas em seu redor.

Função “print_board”

A função “printi_board” imprime o tabuleiro de jogo:

- Inicialmente, é apresentada uma linha com coordenadas para orientação do utilizador, utilizando um ciclo “for” para as letras e outro ciclo “for” para os números. Ainda dentro do segundo “for” é criado um ciclo “if/else if” para imprimir o valor de cada célula.
 - Quando uma célula recebe o valor de 9 (valor de quando a célula ainda não foi revelada) é mostrado um ponto “.”.
 - Quando uma célula toma valor de “-2” (valor da célula com uma mina e quando selecionada pelo utilizador) é impresso um “B”, o jogo termina e o utilizador perde.
 - Se uma célula tomar o valor de “10” (valor de quando a célula está sinalizada como “flag” pelo utilizador) é impresso um “f”.
 - Quando uma célula tem valor de “-1” (valor de uma célula quando é uma mina, mas não foi selecionada pelo utilizador) é impresso um “*”.



- Para todas as células com valor “0” (valor da célula quando está “vazia” e sem minas em seu redor) é imprimido um espaço em branco “ ”.
- Por fim, para todas as outras células é impresso o valor guardado da variável “board” se a célula já estiver sido revelada.

Função “win”

A função “win” é executada quando o utilizador ganha:

- Quando o utilizador revela todas as células com valor diferente de “-1” ganha o jogo, esta função é ativada e mostra ao utilizador uma mensagem de felicitação.

Função “game_over”

A função “game_over” é executada quando o utilizador perde:

- Quando o utilizador seleciona uma célula com valor “-1” o utilizador perde a partida. A célula passa a tomar o valor “-2” e é impresso o “B” e todas as outras minas por revelar acabam por ser reveladas, por meio de um “for”. Por fim é mostrada uma mensagem de reprovação.

Função “uncover”

A função “uncover” é utilizada para revelar as células:

- Esta função começa por avaliar as coordenadas fornecidas por o utilizador para verificar se a célula pertence ao tabuleiro.
 - Quando a célula pertence ao tabuleiro a função revela a célula selecionada, caso esta célula tenha o valor “0” todas as células em seu redor serão avaliadas e se as mesmas tomarem o valor diferente de “-1” e “10” utiliza-se a recursividade para as revelar.
 - Quando a célula não pertence ao tabuleiro é retornado o valor de 0 e nada acontece.

Função “flag”

Para poder sinalizar células com flags utiliza-se a função “flag”:



- Esta função permite ao utilizador que selecione uma célula que supõe conter uma mina e sinalizá-la com uma flag (representa-se com um “f”). O utilizador começa por escolher a opção de jogo (“u” ou “f”), para sinalizar uma célula com uma flag o utilizador tem de optar por “f” e selecionar uma célula, por sua vez esta célula vai ser avaliada para perceber se a mesma já está sinalizada ou não.
 - Caso a célula esteja sinalizada (a célula tem valor de “10”), a mesma deixa de estar sinalizada (passa a ter valor “9”), é removido o “f” e passa a um “.” e é adicionado 1 ao valor de bandeiras (“nflags”) que por sua vez é igual ao número de minas (“mines”).
 - Se a célula selecionada pelo utilizador não estiver ainda sinalizada com uma flag (tem valor de “9”), esta passa a ficar sinalizada (toma valor “10”), deixa de ser representada por “.” para “f” e é subtraído 1 ao número de flags (“nflags”).

Função “main”

Tendo todas estas funções iremos agora trabalhar na função “main”.

- Esta função começa por imprimir uma mensagem de boas vindas ao utilizador e de seguida apresenta um menu de seleção. “0 – Exit, 1 – Generate board, 2 – Read board:
 - No caso de o utilizador escolher 0 sai do jogo.
 - Caso o utilizador escolha 1 vão ser pedidos os dados necessários para criar o tabuleiro de jogo (número de linhas, colunas e minas). Após o utilizador fornecer os dados vai ser gerado um tabuleiro, onde as minas vão ser distribuídas aleatoriamente.
 - Quando o utilizador escolhe a opção 2 vai ser pedido o nome do ficheiro com o quadro predefinido, pelo qual se obtém o valor das linhas, colunas e minas também estas já predefinidas. Ainda nesta opção vão ser avaliadas as células (célula a célula) para determinar o seu valor que depende do número de minas em seu redor.
- Após a escolha de um dos modos (1 ou 2) e do tabuleiro gerado vai ser utilizada a função print_board para imprimir o tabuleiro. Feito isto inicia-se um ciclo while que só termina após o utilizador ganhar ou perder a partida.



Neste ciclo é pedida qual a opção de jogo (“u” ou “f”) e quais as coordenadas da célula pretendida. É utilizado um “for” para determinar qual o valor da coluna a que corresponde a letra.

- De seguida a célula selecionada é avaliada por um ciclo “if/else if” para o caso da célula não pertencer ao tabuleiro, ou de já ter sido revelada e nestes dois casos são impressas mensagens de aviso. Caso nenhum destes dois casos se verifique é chamada a função “uncover” ou “flag” dependendo da seleção do utilizador:
 - Quando o utilizador selecionar a opção “u” e a célula tiver valor de “-1” o utilizador perde a partida e é chamada a função “game_over”, Caso contrário é chamada a função “uncover” para revelar a célula ou células em questão.
 - Se o utilizador selecionar a opção “f” é chamada a função “flag” para sinalizar ou não uma célula.
- De seguida é utilizado um “for” para avaliar se todas as células com minas (valor de “-1”) estão sinalizadas ou se todas as células sem minas (valor diferente de “-1”) estão reveladas, e caso uma destas condições se confirme o utilizador vence a partida.



Discussão de Resultados

Após a realização do código, testamos o mesmo á procura de erros. Um dos erros que encontrámos foi que ao revelar uma célula ao lado de uma flag, eram reveladas ambas as células, o que levou á implementação de mais uma condição na nossa função “uncover”. Posteriormente descobrimos erros relacionados com pontuações (ponto vírgulas e vírgulas) detetamos um erro relacionado com as flags que quando uma célula era sinalizada com uma flag, o número de flags não diminuí. Assim tivemos de alterar a função “flag”.

De seguida, realizamos os testes fornecidos pelo docente.

O primeiro teste (youlost) servia para verificar o que acontece quando se selecciona uma célula com uma mina. Este teste correu bem e obtivemos o resultado pretendido.

O segundo teste (uncover1) o resultado obtido é o pretendido até ser revelada a célula com as coordenadas predefinidas, mas após essa célula ser revelada, como não existe mais coordenadas no ficheiro de teste, o programa entra num loop infinito a pedir mais coordenadas.

No terceiro teste (uncover-rec) vai ser avaliada a ação de quando são inseridas as coordenadas de uma célula que já tenha sido revelada. O resultado é o pretendido, mas como a jogada não é finalizada(win ou game over), o programa entra num loop infinito a pedir mais coordenadas.

Por último o quarto teste (youwin) correu bem e obtivemos os resultados pretendidos. Este teste verifica o que acontece quando todas as células que não sejam minas estão reveladas, ou seja, o utilizador vence a partida.



Conclusão

Com este trabalho percebemos que programar não é algo fácil e que requiere muita paciência, atenção, trabalho e dedicação. No início foi difícil para conseguir começar, mas com persistência e trabalho acreditamos ter conseguido alcançar o objetivo pretendido, e desenvolver um código de acordo com o solicitado.

A perceção é fundamental para uma boa compreensão do código, isto é, conseguir compreender os erros do código para os corrigirmos.

Queremos mencionar também que a elaboração deste trabalho teve uma extrema importância para uma melhor compreensão de todo o conteúdo lecionado durante o semestre.

Mediante os testes realizados e a deteção dos erros foi necessário proceder a uma avaliação detalhada do código, através da utilização dos conhecimentos adquiridos ao longo do semestre, foi possível alterar os defeitos do nosso código que, no entanto, nem sempre foram fáceis de resolver.



Referencias Bibliográficas / Webgrafia

Anon. 2021. "Wikipedia." Retrieved January 24, 2021
([https://en.wikipedia.org/wiki/Minesweeper_\(video_game\)](https://en.wikipedia.org/wiki/Minesweeper_(video_game))).

Belwariar, Rachit. 2018. "Geeksforgeeks." Retrieved January 24, 2021
(<https://www.geeksforgeeks.org/cpp-implementation-minesweeper-game/>).

Nogueira, Vitor and Teresa Gonçalves. 2021. "Material Disponibilizado Nas
Aulas de Programação 1."