Projeto de Laboratórios de Informática 1
Grupo 107

Eduardo Rocha (A77048) André Vieira (A78322)

24 de Março de 2018

Resumo

Este documento apresenta o relatório do projeto de Laboratórios de Informática 1, da Licenciatura em Engenharia Informática da Universidade do Minho.

O trabalho feito corresponde às tarefas 1, 2, 3 e 4, que são responsáveis por construir um tabuleiro de jogo, realizar jogadas, codificar/descodificar o tabuleiro de jogo e realizar a passagem de tempo, respetivamente.

Conteúdo

1 Introdução

Neste projecto foi-nos proposta a realização de um jogo de Bomberman feito todo em Haskell. No nosso trabalho foram realizadas as Tarefas 1, 2, 3 e 4. Esta secção apresenta uma introdução ao trabalho realizado em cada tarefa.

Na *Tarefa 1* é apresentado o problema de construir um mapa válido de Bomberman. O tabuleiro de jogo deve obedecer a algumas regras: a altura deve ser igual ao comprimento, o comprimento do mapa deve ser maior que cinco e o comprimento deve ser sempre ímpar. Após o mapa for produzido, é anexado ao mapa uma lista de strings correspondentes à localização de jogadores, power ups e bombas.

Na *Tarefa 2* é pedida a implementação de comandos de jogo, ou seja: U, D, L, R e B, representando estes comandos a movimentação de um jogador para cima, baixo, esquerda e direita, e a colocação de uma bomba.

Na *Tarefa 3* é realizada a compressão e descompressão de informação correspondente a um tabuleiro de jogo. Nesta tarefa, durante a comprressão, tenta-se

reduzir o tamanho do tabuleiro ao mínimo tamanho possível e durante a descompressão tenta-se restaurar o tabuleiro comprimido para a sua forma original.

Na *Tarefa 4* realizamos a passagem de tempo dentro do jogo de Bomberman. A passagem de tempo inclui a redução do temporizador das bombas assim como a sua explosão caso o temporizador seja reduzido para zero. No caso da explosão de uma bomba, os jogadores, power ups e primeiros tijolos que se encontrem dentro do raio da explosão devem ser retirados. Caso o tempo restante chegue a um certo ponto predefinido, relacionado com a dimensão do tabuleiro de jogo, o mapa começa a fechar-se, ou seja, são colocadas pedras numa forma de espiral em direção dos ponteiros do relógio em todos os instantes de jogo, até que o jogo acabe. Caso uma pedra seja colocada em cima de um jogador, power up ou bomba, a informação correspondente a essa identidade será removida.

Nas secções seguintes serão descritos mais detalhes acerca do trabalho. Na secção **??** serão descritos os problemas encontrados durante a realização de cada uma das tarefas, na secção 3 desenvolvemos os métodos usados para resolver os problemas encontrados e na secção 4 é apresentada a conclusão do relatório.

2 Descrição do Problema

A *Tarefa 1* serve para gerar um mapa válido de Bomberman. Surgem então dois problemas: o primeiro que inclui a criação de um tabuleiro de jogo vazio e o segundo que inclui a geração da lista de strings correspondente às descrições das posições e tipos de power ups. Pretendemos então criar o tabuleiro e a lista de informações separadamente, e depois comparar a lista ao mapa para poder colocar os tijolos e criar as informações sobre power ups, juntando as duas partes no final.

A *Tarefa 2* tem a função de realizar os comandos feitos pelos jogadores, que podem incluir a movimentação de um jogador ou a colocação de uma bomba. Se receber os comandos U, D, L ou R a função deverá verificar se existem objetos nas novas coordenadas pretendidas que impedem o movimento e caso existam, não deve executar quaisquer mudanças. Caso seja dado o comando B, deve verificar se é possível colocar uma bomba naquelas coordenadas e caso seja, deverá incluir a informação necessária sobre a posição da bomba, raio, temporizador e qual jogador a colocou. Quando um jogador se deslocar para as mesmas coordenadas onde se encontra um power up, a tarefa também deverá remover a informação correspondente ao power up do mapa de jogo e deverá acrescentar à linha do jogador que este apanhou o power up.

A *Tarefa 3* apresenta o problema de codificar e descodificar o mapa de jogo. Durante o processo de codificação o programa deverá tentar reduzir para o mínimo tamanho possível o mapa de jogo, sem sacrificar qualquer tipo de informação necessária para o jogo. No processo de descodificação, o mapa de jogo codificado deve ser restaurado para o seu estado original.

A *Tarefa 4* trata da passagem de tempo durante o jogo de Bomberman, criando três problemas: o primeiro que é a redução do temporizador de todas as bombas, o segundo que é verificar se alguma das bombas reduzidas tem agora um temporizador de valor zero e nesse caso deverá executar as mudanças necessárias ao tabuleiro por causa da explosão da bomba e o terceiro problema que envolve fechar o mapa quando o tempo de jogo excede o limite predefinido, colocando uma pedra onde adequado.

3 Concepção da Solução

As subsecções seguintes apresentarão as estruturas de dados usadas, a implementação usada e os testes realizados, respetivamente, para todas as tarefas realizadas.

3.1 Estruturas de Dados

Em todas as tarefas foram utilizadas principalmente listas. Foram usadas listas de Strings para as várias representações do mapa em diferentes estados do trabalho e listas de Integers na *Tarefa 1* para a lista de números gerados aleatoriamente pela tarefa para depois gerar todos os power ups.

3.2 Implementação

A *Tarefa 1* começa por verificar que as medidas fornecidas para o comprimento são válidas e depois chama as funções necessárias para construir o mapa se os inputs fornecidos forem válidos. O principal problema apresentado nesta tarefa foi descobrir como criar o mapa e criar a lista de números a usar para gerar a lista de informação sobre os power ups e como comparar as duas listas para saber onde seriam colocados os power ups e os tijolos. A solução encontrada foi criar uma versão do mapa com os tijolos já colocados e depois usar duas funções: a addBombas e a addFlames para criar as listas de power ups Bombas e Flames que seriam adicionados ao mapa, e depois no final simplesmente juntar tudo - primeiro o mapa, depois as Bombas e depois os Flames.

```
mapaAux :: Int -> [String] -> [Int] -> [String]
mapaAux l m p = m ++ addBombas m p 0 l ++ addFlames m p 0 l
```

A *Tarefa 2* primeiro verifica se os inputs fornecidos são válidos e depois qual o tipo de input fornecido para saber quais funções deverá chamar. Caso o input seja *U*, *D*, *L* ou *R*, o objetivo será tentar mover o jogador que está a fazer a jogada para uma nova posição. Caso o input seja B, o objetivo será tentar colocar uma bomba na posição atual do jogador. Durante esta Tarefa, foi preciso atenção à interação entre vários elementos do jogo, nomeadamente quais podem coexistir na mesma posição e quais não podem. Um dos problemas principais que encontramos foi a implementação do que acontece quando um jogador apanha um power up. Nesse caso, a informação correspondente ao power up apanhado deverá ser eliminada e o um símbolo correspondente ao tipo de power up apanhado será adicionado à descrição do jogador. O principal problema aqui foi saber distinguir quando o power up apanhado foi o primeiro apanhado. A função responsável por atualizar a descrição do jogador é a função atualiza.

```
atualiza :: Int -> (Int,Int,String) -> Char -> String
atualiza a (x,y,s) c | c == '+' || c == '!' =  show a ++ [' '] ++ show x ++ [' ']
                                               ++ show y ++ [' '] ++ s ++ [c]
                     | otherwise = show a ++ [' '] ++ show x ++ [' '] ++ show y
                                               ++ " " ++ s
```

A *Tarefa 3* está dividida em duas partes. A primeira parte inclui a codificação do código e a segunda a descodificação do código. Durante a codificação

do código são retiradas as paredes laterais ao mapa e durante a descodificação sao restauradas as paredes removidas durante a codificação. Para isto são utilizadas as funções compress e decompress, respetivamente.

```
compress :: [String] -> [String]
compress (h:t) = retiraP t

decompress :: [String] -> [String]
decompress l = linhaPU ((length h)+2) : adicionaH (adicionaV (h:t)) ((length h)+2)
```

A *Tarefa 4* trata da passagem de tempo no jogo. Primeiro, a tarefa tratará de reduzir o temporizador de todas as bombas. A seguir, irá verificar se existem bombas cujo temporizador atingiu zero. Por cada bomba prestes a explodir (temporizador a zero) chamará funções que estarão encarregues de alterar a informação do mapa afetada pela explosão da bomba, ou seja, tijolos, jogadores, power ups ou bombas que serão removidas/alteradas. Caso necessário, após serem realizadas as alterações anteriormente descritas, o mapa continuará o seu processo de ser fechado, ou seja, será colocada uma pedra (carácter cardinal) no local adequado. Descobrir o local adequado foi o principal problema apresentado durante a realização desta tarefa. A solução encontrada foi passar pelo mapa na direção dos ponteiros do relógio em forma de espiral, removendo camadas exteriores preenchidas completamente por pedras. A função responsável pelo processo anteriormente descrito, a spiral, verifica a primeira linha para ver se é feita toda de pedras, depois verifica a última coluna, a seguir verifica a última linha e finalmente a primeira coluna. Se verificar que existem espaços numa das linhas/colunas analizadas, irá tentar colocar uma pedra no sítio adequado; se verificar que a camada exterior do mapa está toda preenchida, irá remover essa camada, reiniciando a função para o mapa interior. Após a função acabar, o mapa será restaurado ao seu estado original.

3.3 Testes

Para testar as várias tarefas recorremos à plataforma disponibilizada pelos docentes assim como testes realizados manualmente por nós no terminal a cada tarefa. Como exemplo de mapa usamos também o mapa seguinte:

```
#########
#       #
# #?# # #
#     ? #
#?# # #?#
# ?  ?  #
# #?#?# #
#  ??   #
#########
+ 3 3
! 5 5
* 7 7 1 1 10
0 4 3 +
1 7 7
```

4 Conclusões

Em conclusão, estamos contentes com os trabalhos realizados nas Tarefas 1, 2 e 4, mas achamos que a Tarefa 3 podería estar muito mais desenvolvida, estando bastante incompleta por falta de tempo. Gostariamos também de ter realizado os trabalhos necessários para as Tarefas 5 e 6, que não foram desenvolvidas pelo nosso grupo.