

Projeto de Laboratórios de Informática 1

Grupo 163

Bruno Manuel Chaves Martins (A80410)
Pedro Miguel de Leal Meireles Pereira (A80627)

2 de Janeiro de 2017

Resumo

Este documento apresenta o relatório do projeto de Laboratórios de Informática 1 (LI1), da Licenciatura em Engenharia Informática da Universidade do Minho.

O trabalho proposto foi uma reprodução do jogo Bomberman na linguagem de programação *Haskell*, existem duas Fases e cada uma delas aborda diferentes aspetos da conceção do jogo. A primeira Fase incide sobre a produção do mapa, a movimentação de jogadores e a compressão do estado do jogo. A segunda Fase incide sobre a passagem do tempo e suas consequências, a impressão de gráficos do jogo e a produção de um Bot capaz de jogar sozinho.

Conteúdo

1 Introdução

Este documento apresenta o relatório do trabalho da UC de Laboratórios de Informatica do primeiro ano do Mestrado Integrado em Engenharia Informática da Universidade do Minho, onde foi proposto a reprodução do jogo Bomberman.

Este desafio foi-nos posto para a aplicação de conceitos teóricos lecionados na UC de Programação Funcional com o intuito de utilizar a linguagem de programação *Haskell* e desta forma melhorar a utilização da mesma.

Este problema foi dividido em duas Fases cada uma subdividida em três Tarefas de forma a que a resolução das mesmas tivesse uma dificuldade crescente e quem realizava o trabalho tivesse de aprofundar cada vez mais os seus conhecimentos para a resolução dos ditos *problemas*, desenvolvendo uma capacidade de interpretação de problemas e resolução dos mesmos cada vez maior.

A primeira fase do projeto centrava-se mais na impressão do mapa com os powerups e na movimentação de um jogador, já a segunda fase era focada no aspeto mais técnico do jogo onde era necessário fazer os gráficos do Jogo, a passagem do tempo e um bot que conseguisse jogar sozinho.

Neste relatório, a Secção Problema descreve o problema a resolver enquanto a Secção Solução apresenta e discute a solução proposta pelos alunos. O relatório termina com conclusões na Secção Conclusão, onde é também apresentada uma análise crítica dos resultados obtidos.

2 Descrição do Problema

O *problema* que se pretende resolver no projeto de LI1 é a criação do jogo Bomberman funcional. Para isso como referido anteriormente o projeto foi dividido em duas Fases e cada fase em três Tarefas.

Na primeira Fase o primeiro desafio era a impressão do mapa do jogo com os PowerUps a partir de uma lista de numeros aleatorios com base numa *Dimensão* e uma *Seed* de forma a que cada mapa desde que tivesse uma dimensão ou seed diferente, o mapa produzido seria distinto. ainda na primeira fase outra tarefa era a movimentação dos jogadores, ou seja, quando premida uma tecla de movimento, fazer a posição do jogador mudar com base na tecla. Na terceira Tarefa foi pedido o *Encode* e o *Decode* do mapa, ou seja, a compressão do mapa para um formato mais simplificado mas que permitisse depois a sua descompressão a partir do codificado.

Na segunda Fase e na primeira Tarefa desta, era necessário fazer a passagem do tempo, sendo a principal consequência a explosão das *Bombas* e por consequência os elementos do jogo removidos, ainda nesta Tarefa foi preciso fazer uma espiral que a partir de um determinado momento do jogo calculado a partir da dimensão do mapa começa e remove todos os elementos do jogo e substitui por uma pedra fixa com o intuito de acelerar a conclusão do jogo. Na segunda Tarefa era necessário a impressão dos gráficos do jogo a partir da biblioteca Gloss. Na terceira Tarefa é necessário a criação de um bot capaz de reagir e "tomar" decisões por si próprio, por exemplo fugir de uma bomba antes de ela explodir, de modo a sobreviver e ganhar a outros boots construídos.

3 Conceção da Solução

Esta secção deve descrever o trabalho efetivamente desenvolvido pelos alunos para resolver o problema apresentado na secção anterior. Segue-se uma sugestão de organização para esta secção.

3.1 Estruturas de Dados

A *estrutura de dados* ao longo do projeto foi sempre começada a partir do mapa com os PowerUps, Bombas e jogadores. Sempre que era necessário algum dado para a realização das Tarefas era percorrido a *Lista de Strings* original até encontrar a informação relevante.

3.2 Implementação

Na Tarefa 1 começou-se por definir uma função que à lista de numeros aleatorios dada como argumento da função original retirasse o número exato de elementos necessários para contruir o mapa a partir da dimensão dada. A impressão do mapa em si começa com a construção das paredes e pedras fixas no mapa, coisas que não variam qualquer que seja o mapa. Para os *Tijolos* e os *PowerUps* aparecerem no mapa foi criada uma função que a corria a lista de números aleatórios e verificava se o nº correspondia ao da *Bomba* ou do *Flame*, bem como se fosse *Tijolo*. Mais Tarde foi necessária uma função que a colocasse no fim do mapa as coordenadas dos *PowerUps* (*Bombas e Flames*), onde percorria a lista de nº aleatórios e com base numa coordenada de acumulação e na dimensão do mapa achava as coordenadas. No final foi encontrado um problema que era que, em vez de ser imprimido uma *Lista de Strings* era apenas imprimida uma *String*. Para resolver esse problema foi criada uma função que a partir da dimensão dividia a *String* no correto nº de elementos, adicionando depois as coordenadas dos *PowerUps*.

Na Tarefa 3 o *Encode* foi iniciado pela produção de uma função que a partir do primeiro elemento da *Lista de Strings* calculava a dimensão do mapa e substituiu esse elemento por um nº inteiro, o qual mais tarde é a base do *Decode*, depois outra função retirava o primeiro e último elemento de cada *String* (os cardinais). São criadas duas funções que selecionam e retiram as pedras fixas do mapa. Outra função calcula o nº de espaços vazios consecutivos e substitui pelo nº final na *String*, e o mesmo é feito às *Strings dos PowerUps*. No *Decode* a partir da dimensão calculada, recria-se as Pedras fixas no mapa, e a partir do nº de espaços consecutivos calculado volta-se a colocar os espaços nos devidos locais.

Na Tarefa 4, começou-se por criar uma função que a cada instante do jogo, retirava um "Tick" se houvesse bombas plantadas no mapa. Outra função verificava se não existem bombas no instante zero para explodir. Caso exista alguma Bomba no instante zero, recorre-se a uma função que imprime o resultado da explosão com base em quatro vetores de direção, para cima, baixo, esquerda

e direita. Seguem-se uma série de condições para averiguar o que é destruído no mapa ou onde pára a chama da explosão. Sempre que alguma alteração é necessária existiam funções que destruíam tijolos, PowerUps ou jogadores, mas também caso a chama de uma bomba atingisse outra, garantir que passava os seu cornómetro para 1 de modo a que explodisse no instante seguinte. Para o começo da espiral criou-se uma condição que a partir dos instantes que faltavam para terminar o jogo decidia se a espiral começava. Caso a espiral fosse iniciada, baseava-se nos quatro vetores de movimento tal como explosão de uma bomba. Sendo assim a função foi programada para sempre que atingisse uma borda do mapa se alterasse o vetor de movimento. À medida que as pedras eram postas no decorrer da espiral tudo o que estivesse naquela posição iria ser removido do mapa, bem como se pertencessem também à lista no final, de Jogadores, PowerUps ou Bombas.

4 Conclusões

A produção deste Trabalho demonstrou ao longo da sua execução que o que parece simples ao utilizador final, por trás tem muito conhecimento e dificuldades que são necessárias ultrapassar para que o produto final seja o que queremos. Este projeto ajudou-nos no desenvolvimento de ferramentas de pesquisa e de resolução de problemas ao longo que eles apareciam, também ajudou no desenvolvimento da capacidade de trabalho de grupo pois nem sempre as opiniões para a resolução do problema são as mesmas mas temos de chegar a um entendimento que agrada a todas as partes.

O objetivo proposto no início do trabalho não foi atingido porque o jogo Bomberman não ficou totalmente operacional, apenas partes do trabalho proposto foram executadas como se pedia, nomeadamente a Tarefa 1 e 3, ficando a Tarefa 4 parcialmente executada porque não concretizava totalmente o pedido.

O não atingimento deveu-se parcialmente à falta de tempo para a execução das tarefas mais exigentes e elaboradas e também à falta de alguma prática para a resolução de certos problemas ao longo do trabalho o que prologava a sua resolução. Mesmo assim todas as dificuldades ultrapassadas contribuíram para um aumento do conhecimento técnico da linguagem de programação Haskell e a correção, num futuro projeto, das razões para a não concretização dos objetivos, irão facilitar o atingimento do objetivo proposto.