Projeto de Laboratórios de Informática 1
Grupo 166

Jéssica Lemos (A82061) Ana Ribeiro (A82474)

30 de dezembro de 2016

Resumo

Este relatório aborda a implementação do clássico Bomberman que se insere na disciplina de Laboratórios de Informática 1 (LI1), do Mestrado Integrado em Engenharia Informática da Universidade do Minho.

Neste expomos os problemas e as soluções encontrados na realização das seis tarefas que constituem o projeto. A primeira consiste na construção do mapa, a próxima permite o movimento dos jogadores, a terceira na compressão/descompressão do código, a tarefa seguinte permite reagir à passagem do tempo, na quinta tarefa é implementada a parte gráfica do jogo e na última é elaborada uma estratégia de combate.

Conteúdo

Resumo **i**

1 Introdução **1**

2 Descrição do Problema **2**

3 Concepção da Solução **2**
3.1 Implementação 2
3.2 Testes 3

4 Conclusões **5**

1 Introdução

Com este projeto, pretendemos implementar o clássico Bomberman em *battle mode*. Este jogo tem como objetivo eliminar os inimigos e os obstáculos. Para tal, o jogador deverá colocar bombas, cujo raio de ação permita alcançar um dos objetivos anteriormente referidos. É de realçar que este deverá ter em atenção as bombas colocadas no mapa, de modo, a não ser atingido por estas. Com o objetivo de tornar a sua ação mais eficaz, este terá de apanhar os *Power ups*.

O projeto encontra-se divido em duas fases, cada uma com três tarefas. A primeira fase inicia-se com uma tarefa, cujo objetivo é construir o mapa de jogo, para qualquer dimensão, desde que seja ímpar e maior ou igual a cinco. Assim, implementamos um código que a cada ponto do mapa atribuía o conteúdo correspondente, recorrendo às suas coordenadas. A segunda tarefa pretende implementar movimentos no jogador através de cinco comandos ("U","D","L","R","B"). Deste modo, verificamos se o jogador se podia realizar o movimento facultado e em caso afirmativo efetuava-o, caso contrário devolvia o mesmo estado de jogo. A última tarefa desta fase, tem como propósito a elaboração de um método de compressão/descompressão.

A quarta tarefa, já pertencente à segunda fase, tem como finalidade a alteração do estado do jogo em função do tempo. Desta forma, como passar dos ticks fomos diminuindo ao tempo de explosão das bombas e caso alguma alcançasse um, alterávamos o estado do jogo consoante as suas consequências. A tarefa cinco, pretende a realização da parte gráfica do jogo, utilizando a biblioteca *Gloss*. A última tarefa tem como objetivo a elaboração de um *bot*, que terá como base uma estratégia de jogo.

2 Descrição do Problema

Como referido anteriormente, a primeira tarefa deste projeto consiste na elaboração do mapa do jogo com uma dimensão ímpar, maior ou igual a cinco. O input é a dimensão do mapa e um número inteiro positivo para usar como semente. Através da utilização da semente é devolvida uma uma lista com números aleatórios, que posteriormente irão definir o conteúdo de cada célula do mapa não pré-definida. Se o valor atribuído a uma determinada célula for menor que um então estará escondido um *Power up* Bombs atrás de um tijolo, caso esteja entre dois e três então será um *Power up* Flames atrás de um tijolo. Se estiver entre quatro e trinta e nove será tijolo, caso contrário será vazio. Após o mapa impresso deverá surgir a imformação dos *Power ups*, sendo que os Bombs deverão aparecer primeiro que os Flames, ordenadamente. Note-se que um pedra corresponde, um tijolo e um vazio. É, também, de realçar que todo o contorno do mapa será pedra, assim como cada linha ou coluna alternadamente e os quatro cantos do mapa tem três células vazias.

Na segunda tarefa, cujo objetivo já foi mencionado, o input é o mapa do jogo, um comando e o identificador do jogador. Os comandos possíveis são, em que o jogador se movimenta para cima, que permite o movimento para baixo, e que permitem o movimento para a esquerda e para a direita, respetivamente. Ainda temos, o comando em que o jogador coloca uma bomba. Esta é identificada por seguida do identificador do jogador, a sua posição, o seu raio de ação e o tempo restante para a sua explosão. O jogador é caracterizado pelo seu identificador, seguido das suas coordenadas e dos *Power ups* que possui. Caso o jogador se movimente para um local com *Power up*, é necessário retirar este do mapa e acrescentar à informação do jogador o símbolo do *Power up* apanhado. É de destacar que no mapa do jogo as bombas surgem ordenadamente depois dos *Power ups* e os jogadores, ordenados pelos seus identificadores, após estas. E, também, é de salientar que caso a ação na seja possível é devolvido o mesmo estado de jogo.

Na tarefa seguinte, o objetivo é implementar um método de compressão e descompressão, que possibilita a redução de caracteres.

Na primeira tarefa, da fase seguinte, é dado o estado de jogo e os ticks que faltam para terminar o jogo. A cada tick é necessário diminuir o tempo de explosão de cada bomba. Quando alguma destas chegar a um, esta irá explodir e será necessário implementar no estado do jogo as alterações consequentes. Caso um jogador se encontre no raio de ação da bomba, este irá morrer pelo que a sua informação será eliminada do estado de jogo. Quando a chama atinge uma pedra, esta impede-a de prosseguir. Se encotrar um tijolo este será destruido e a chama deixa de se propagar. Caso a chama passe num *Power up*, a informação deste será eliminado e a sua passagem será bloqueada. Na hipótese de a chama alcançar outra bomba, o tempo desta será reduzido para um. Assim que faltar a dimensão menos dois ao quadrado instantes para o jogo terminar irá começar a cair blocos de pedras no efeito de espiral. A quinta tarefa é caracterizada pela implementação do jogo através da biblioteca *Gloss*.

Por fim, na última tarefa será elaborado um *bot*, que joga automáticamente.

3 Concepção da Solução

3.1 Implementação

Na primeira tarefa, começamos por definir os pontos do mapa pré-definidos. De seguida, selecionamos o número de células por definir da lista de números aleatório gerada através da semente. Tendo em conta que cada número se encontra relacionado a um determinado conteúdo, fomos estabelecer a cada célula esse mesmo conteúdo. Note-se que nas células onde se encontram *Power ups* foi realizada uma conversão para o símbolo de tijolo, dado que se encontram aí escondidos. Já com cada ponto do mapa determinado, associamos todos os pontos de cada linha. Posteriormente, agregamos todas as linhas geradas anteriormente. Com o mapa formado adicionamos a informação relativa aos *Power ups*. De modo a obtermos a informação destes, definimos os *Power ups* das células que os continham. Após, juntamos os *Power ups* Bombs por linha e posteriormente, asso-

ciamos as respetivas linhas. Assim, aplicamos o mesmo processo aos *Power ups* Flames. De modo a que os *Power ups* Bombs surjam primeiro que os *Power ups*, associamos as funções referidas anteriormente por esta ordem. É de realçar, que como o mapa se inicia na coordenada (0,0), subtraímos um à dimensão utilizada nas outras funções.

Iniciamos a segunda tarefa elaborando uma função que ia ao estado do jogo retirar à lista da informação do jogador, ao qual foi implementado o comando, as suas coordenadas. Convertendo esta informação num número do mapa, verificamos se a ação do jogador era possível e se o jogador existia. Caso não fosse, devolviamos o estado de jogo atual. Na eventualidade, de o jogador pretender colocar uma bomba e não tiver *Power ups*então é devolvido o mapa inicial. Se verificarmos que o movimento pode ser efetuado, então criamos uma lista com as novas coordenadas deste. Posteriormente, criamos uma função que averiguava todos os casos possíveis e que realizava as alterções no mapa consoante estes. Inicialmente, abordamos o caso do jogador se movimentar para uma célula com *Power ups* e de já possuir outros. É de salientar, que nesta situação separamos os *Power ups* Bombs dos Flames, dado que de acordo com o tipo de *Power up* acrescentávamos um símbolo diferente à informação do jogador. Assim, era necessário ir ao estado de jogo alterar a informção deste. Na eventualidade de o jogador não possuir *Power ups* e ir para um local com um, ao adicionarmos o símbolo correspondente, necessitamos de inserir também um espaço para o separar da coordenada deste. Na hipótese, de o jogador receber o comando e no mapa já existir um bomba, recorremos a uma nova função, que no caso deste pretender colocá-la numa célula onde já existe uma, é devolvido o estado de jogo inicial. Na possíbilidade, de a coordenada da bomba que será posicionada ser superior à da bomba já existente, então é adicionada ao estado de jogo, a informação da nova bomba anteriormente à informação da bomba já existente. Caso contrário, a ordem pela qual é acrescentada é a inversa.

Na primeira tarefa da segunda fase, começamos por elaborar uma função que a cada vez que a função principal (avanca) fosse evocada ia subtrair um ao tempo de explosão de todas as bombas existentes no mapa. Caso não existisse bomba no mapa então era devolvido o estado de jogo inicial. Na eventualidade,de no mapa existirem bombas prestes a explodir e bombas cujo tempo de explosão é superior a um, então reduzimos o tempo de explosão das bombas com tempo superior a uma e associamos à restante informação do mapa que já não continha a informação relativa à bomba que explodiu e que contém as transformações causadas por esta. Se o mapa apenas contiver bombas prestes a explodir, então ao mapa já com as alterações causadas pela explosão da bomba, fomos retirar a informação referente a esta. Na possiblidade de no estado de jogo não existirem bombas prestes a explodir, vamos apenas modificar os tempos de explosão das bombas. As situações que conduzem a alterações são cinco: a chama encontrar uma pedra, um tijolo, *Power ups*, um jogador ou uma bomba. Quando for uma pedra, a chama não prossegue mais e como tal o mapa devolvido é o mesmo. Se for tijolo, este é destruido e como tal é eliminado do estado de jogo, e a chama não progride. Os *Power ups* quando atingidos pela chama são suprimidos do mapa e a chama não avança. Os jogadores são mortos pela chama, pelo que a sua informação é excluída do estado de jogo. Por fim, no caso de ser uma bomba, o seu tempo de explosão passará automaticamente para um. No nosso projeto, fomos realizando as alterações, separadamente, para cada uma das direções em que a chama se podia propagar. Assim, necessitamos de recorrer a quatro funções que iam associando as informações referentes a cada direção. Para verificar as células afetadas pela chama, verificamos o seu alcance através do raio.

Na última tarefa, a nossa estratégia de jogo, foi conduzir o jogador até à linha seguinte do meio do mapa. Para tal, e tendo em conta que este inicia sempre num dos cantos, indicamo-lhe um movimento para cima ou para baixo, consoante a sua posição. Na eventualidade, de este não se movimentar e célula para a qual pretendia ir for um tijolo, este coloca uma bomba.

3.2 Testes

Neste projeto, foi necessário realizar teste para duas tarefas: tarefa dois e tarefa quatro, que pretendiam testar o máximo de casos possíveis. Para a segunda tarefa realizamos oito exemplos, em que averiguamos as situações em que o jogador não pode efetuar o movimento; em que adquire um *Power up*, sendo que este já possui outros; em que o jogador tenta colocar uma bomba numa

célula já com uma; em que o jogador tem que posicionar uma bomba e o mapa já contém uma; em que o jogador não tem *Power ups* e dirige-se para um local com um.

Para a quarta tarefa, utilizando os exemplos anteriores verificamos o caso em que não existem bombas prestes a explodir e elaboramos mais 4 exemplos para testar as restantes situações: colocamos duas bombas seguidas prestes a explodir; em que chama atinge uma pedra; em que a chama atinge um tijolo; um jogador encontra-se na mesma célula que a bomba; posicionamos duas bombas consecutivas, sendo que uma está prestes a explodir e a outra não; quando atinge um *Power up*; quando um jogador se encontra no raio de ação da bomba.

4 Conclusões

Em última instância, este projeto consiste na elaboração do clássico Bomberman em *battle mode*, sendo que se encontra divido em duas fases subdivididas em três tarefas cada. A primeira fase tinha como objetivo a construção do mapa de jogo, a reação a comandos e à compressão/descompressão do jogo. Já a segunda fase visava definir o efeito da passagem do tempo, a criação da parte gráfica e a conceção de um *bot*. —- Tendo em conta os objetivos referidos anteriormente e dado que não realizamos a tarefa três, a tarefa cinco e a segunda tarefa continha erros, consideramos que, apesar de tudo, foi notória a nossa evolução ao longo do projeto. Esta evidenciou-se em diversos aspetos, nomeadamente no tempo de execução e na qualidade do código.