Ramon Gonçalves Gonze

1. **Introdução**

O objetivo do Trabalho Prático 1 de AEDS1 é desenvolver um jogo semelhante ao Bomberman. Esse desenvolvimento faz com que sejam aplicados os conceitos básicos de programação aprendidos em sala, tais como manipulação de vetores, structs, arquivos, entre outros. Para a parte gráfica, foi preciso adquirir conhecimento da biblioteca Allegro. Essa foi utilizada na sua versão 5.0.10.

1. **Implementação**
   1. **Estrutura de Dados**

Para a implementação do trabalho foram criados dois Tipo Abstrato de Dado (TAD): **bomba** e **player**. A estrutura **bomba** possui 4 elementos:

* ALLEGRO\_BITMAP \*imagem\_bomba: Variável para receber a imagem da bomba;
* int coord\_x: Variável para receber a posição x da bomba;
* int coord\_y: Variável para receber a posição y da bomba;
* int relogio: Variável que é utilizada para contar o tempo que a bomba permanece no mapa.

A estrutura **player** possui 5 elementos:

* ALLEGRO\_BITMAP \*imagem\_player: Variável para receber a imagem do player;
* int coord\_x: Variável para receber a posição x do player;
* int coord\_y: Variável para receber a posição y do player;
* int vida: Variável que é utilizada indicar se o player está vivo ou morto. 1 significa vivo, 0 significa morto.
* bomba bomb[3]: Vetor do tipo bomba, e indica que cada player carrega consigo 3 bombas.
  1. **Funções e Procedimentos**

O TADs criados foram utilizados nas seguintes funções: moveMonstrosX, moveMonstrosY, movePerseguidor, testeMorte e iniciaInimigo. Em todos, o(s) parâmetro(s) passado(s) é uma variável do tipo ponteiro, e cada função faz a alteração dos elementos desses TADs, de acordo com a finalidade com que são chamados na função main.

* + 1. **Função moveMonstrosX**

Esta função é utilizada para alterar a posição X dos inimigos na tela, através da alteração da variável coord\_x (que está contida na estrutura **player**). A mudança da variável é feita randomicamente, através da função rand(). Antes da alteração da coordenada, são feitos os testes dos limites do mapa e do labirinto e as posições das 3 bombas do jogador. Após a alteração da variável coord\_x, também é feito a troca da variável **imagem\_player**, que troca a imagem de acordo com a incrementação ou decrementação da posição X.

* + 1. **Função moveMonstrosY**

Esta função tem a mesma funcionalidade da função moveMonstrosX, a diferença é que a alteração é feita na coordenada Y do monstro, contida na variável coord\_y (também contida na estrutura **player**).

* + 1. **Função movePerseguidor**

Esta função possui a finalidade de fazer com que o monstro que está sendo passado como parâmetro (player inimigo[i]) persiga o jogador. Ela verifica a posição em que o jogador está no momento em que ela é chamada, e faz a alteração das posições X e Y do monstro se aproximarem das posições X e Y do jogador. Também é feita a troca da variável **imagem\_player** de acordo com a incrementação ou decrementação das posições X e Y do monstro.

* + 1. **Função testeMorte**

Esta função possui a finalidade de verificar se algum monstro morreu. A única forma de um monstro morrer, é por uma bomba do jogador. A função é chamada toda vez que o laço de repetição while principal do jogo é executado. Ela verifica se as posições X e Y do monstro coincidem com as posições X e Y de alguma das três bombas, no momento em que a variável **relogio** delas é igual a 0. Caso essas posições se coincidam, é atribuído 0 à variável **vida** do monstro, e suas posições X e Y recebem o número 3000 (eles são movidos para fora do mapa).

* + 1. **Função iniciaInimigo**

Esta função possui a finalidade de atribuir posições X e Y dos monstros randomicamente no mapa do jogo, e atribuir 1 à variável **vida** do monstro, indicando que ele está vivo e está no mapa.

* 1. **Programa principal**

O programa principal funciona basicamente por meio do laço de repetição while principal. Antes de este laço começar a ser executado, foram declaradas todas as variáveis e algumas funções do alegro foram executadas, cada qual com a sua funcionalidade. Por exemplo, para serem utilizadas fontes de textos, as funções permitem essa utilização são as al\_init\_font\_addon() e al\_init\_ttf\_addon(). O fim do jogo é sinalizado com a parada de execução do laço while. Esse evento acontece quando o X da janela é pressionado ou quando o jogador morre.

* 1. **Organização do Código, Decisões de Implementação e Detalhes Técnicos**

O código possui um arquivo principal, de nome *main.c*. O compilador utilizado foi o GCC no sistema operacional Windows 10. A pasta em que está o programa já possui um arquivo executável *main.exe*, mas caso queria compila-lo, basta executar a linha de comando *make* no diretório em que está a pasta com os arquivos do jogo.

1. **Conclusão**

O desenvolvimento do trabalho pode ser concluído dentro das expectativas, e o conteúdo aprendido em sala de aula pode ser totalmente aplicado.

A principal dificuldade foi em separar as funções contidas no código em arquivos diferentes do principal. Houveram problemas ao utilizar funções do Allegro que reproduzem áudios. Por este motivo, o código possui somente um arquivo.