Implementação do Jogo Bomberman em C: Relatório

Universidade Federal do Rio de Janeiro – UFRJ
Instituto de Matemática – Departamento de Ciência da Computação
Curso de Ciência da Computação

Disciplina: Programação de Computadores II
Professor: Marcos Tomazzoli Leipnitz
Período: 2025.1

Implementação do jogo Bomberman em C com Raylib
Trabalho Prático

Grupo:

- Nikolas Miguez – DRE 121129789
- Giulia Tafuri – DRE 122030496
- Pedro Miller – DRE 122067883
- Thales Gabriel – DRE 123752017

Rio de Janeiro, julho de 2025

1. **Descrição do Trabalho**

Este projeto consistiu na implementação de uma versão simplificada do jogo Bomberman, desenvolvida em linguagem C utilizando a biblioteca gráfica Raylib. O principal objetivo foi aplicar os conceitos abordados na disciplina Programação de Computadores II, tais como manipulação de matrizes, estruturas, ponteiros, listas encadeadas, leitura de arquivos e modularização de código.

O jogo é composto por fases, cuja quantidade é definida pela quantidade de arquivos de mapa externos, nos quais o jogador deve coletar cinco chaves escondidas em caixas e evitar ser atingido por explosões ou inimigos. O programa inclui movimentação do jogador, inimigos com deslocamento automático, plantio e detonação de bombas, e um sistema de pontuação. O jogo avança automaticamente para o próximo mapa após a coleta de todas as chaves da fase atual. O jogo termina quando o jogador passar por todos os mapas disponíveis, ou perder todas as suas vidas em determinado mapa.

2. **Instruções de Uso**

2.1 Requisitos

Compilador C (gcc);

Bibliotecas Raylib e Dirent instaladas no sistema;

Sistema operacional: Linux ou Windows com suporte à Raylib;

Diretório maps/ contendo os arquivos de mapa no formato .txt;

Arquivos de Bomberman, disponíveis em https://github.com/gtafuri/Bomberman, ou no arquivo Zip disponibilizado ao professor via Classroom.

2.2 Compilação

- Em sistemas Linux com Raylib instalada:

gcc main.c jogo.c mapa.c ui.c -o bomberman -lraylib -lm -ldl -lpthread -lGL

- Em sistemas Windows (com MSYS2 e Raylib):

Nossa recomendação: com a biblioteca Make instalada, utilize o comando abaixo para compilar o makefile disponível no zip do projeto.

Make

Manualmente:

gcc -I. -Wall -o bomberman.exe main.c jogo.c mapa.c ui.c -lraylib -lopengl32 -lgdi32 -lwinmm -lm

ou

gcc main.c jogo.c mapa.c ui.c -o bomberman.exe -lraylib -lopengl32 -lgdi32 -lwinmm

2.3 Execução

No terminal, execute o programa com:

- Linux:

./bomberman

- Windows:

./bomberman.exe

Certifique-se de que o diretório maps/ esteja no mesmo local do executável e contenha pelo menos um mapa válido com:

Exatamente 25 linhas e 60 colunas por linha;

1 jogador (J);

5 caixas com chave (K);

5 inimigos (E).

2.4 Controles

W, A, S, D ou setas: movimentar o jogador;

B ou Espaço: plantar bomba;

TAB: abrir ou fechar o menu;

ENTER: selecionar opção; (ou N para novo jogo, C para carregar jogo, S para salvar jogo, Q para sair do jogo e V para voltar).

ESC: encerrar o jogo.

2.5 Objetivo

O jogador deve encontrar e coletar cinco chaves (representadas por K) escondidas em caixas. Ao destruir as caixas certas com bombas, as chaves são reveladas. Quando todas as chaves forem coletadas, o jogador avança para a próxima fase. O jogo termina quando todas as fases forem concluídas ou quando o jogador perder todas as vidas.

3. **Divisão de tarefas**

Giulia: Criação da lógica geral do jogo, codificando uma versão simplificada que atendia os requisitos obrigatórios. Depois, realizou ajustes para modularizar o código e melhorar sua jogabilidade (ex: avisar se não existe mapa disponível no diretório, ajustar a velocidade de inimigos, permitir a movimentação constante do jogador ao segurar teclas de direção, adicionar animações via Raylib para tornar o jogo mais dinâmico, etc). Adicionou a funcionalidade extra de gravar um placar em arquivo binário. Ao completar o jogo, ou perder com uma boa pontuação, o jogador receberá um aviso sobre sua colocação se estiver entre os 10 primeiros.

Pedro: Ajustou o código para que o jogador (e os inimigos) não pudessem passar por cima das bombas, nem andar na diagonal.

Nikolas: mapas, relatório

Thales: funções extra