



**Centro de
Informática**
UFPE

ZOMBIE RAMPAGE

Jogo 2D – Grupo 2 da Disciplina de Algoritmos

1. Introdução:

O projeto tem como objetivo o desenvolvimento de um jogo **2D RPG com elementos de estratégia**, dividido em **duas etapas de complexidade crescente**, utilizando a linguagem **C** como base.

O jogo iniciou em uma versão **textual (terminal)**, com foco nos **conceitos fundamentais de programação em C**, e promete evoluir para uma versão **gráfica 2D**, implementando **estruturas de dados, algoritmos e elementos de IA**.

O tema escolhido é **pós-apocalíptico**, em um mundo devastado por um vírus que transforma humanos em zumbis. O jogador assume o papel de um sobrevivente que precisa explorar, lutar e administrar recursos para permanecer vivo.

Membros:

- Lucas Silva Cabral <lsc9>
- Bruno Ramos <bgprs>
- Gryghor Camonni <gcfc>
- Diogo Rodrigues <dsr>
- Flávia Vitória <fves>

Link do Repositório:

<https://github.com/LucasCabra7/Algorithm-group-2.git>

2.Enredo e Ambientação:

Após uma epidemia global causada por um vírus experimental, a humanidade entrou em colapso. Cidades foram destruídas, governos desapareceram e o mundo agora é dominado por hordas de zumbis famintos. O jogador interpreta (**nome do jogador**), um ex-militar que desperta em um abrigo subterrâneo semanas após o surto. Sem contato com outros sobreviventes, ele precisa:

- Explorar o ambiente em busca de suprimentos,
- Lutar contra inimigos (zumbis),
- Escolher classes e habilidades,
- E tentar encontrar uma saída para um local seguro.

O jogo mistura **exploração, combate em turnos e gerenciamento de inventário**, com foco em **decisões estratégicas e sobrevivência tática**.

3.Mecânicas Principais:

3.1. Sistema de Classes:

- *Soldado*: alto ataque, baixo intelecto.
- *Engenheiro*: cria armadilhas e usa itens especiais.
- *Médico*: cura e melhora de status.

3.2. Sistema de Batalhas por Turnos:

- Jogador e inimigo alternam ataques e defesas.
- O sistema de dano é baseado em atributos da classe, itens e níveis.

3.3. Exploração de Mapa (por matriz):

- O mapa é representado por uma matriz.
- Cada célula pode conter terreno, zumbi, recurso ou obstáculo.

3.4. Inventário e Itens:

- Coleta de armas, munições, kits médicos e suprimentos.
- Cada item ocupa espaço limitado (inventário com vetor dinâmico).

3.5. Sistema de Progressão:

- Pontos de experiência ganhos em batalhas (XP).
- Evolução de atributos e desbloqueio de habilidades.

4. Funcionalidades por Etapa:

Etapa 1 – Versão Textual (Terminal):

Objetivo: Implementar os fundamentos da linguagem C e criar uma base funcional.

Funcionalidades:

- Interface por texto (menus e opções via terminal).
- Criação e seleção de personagem.
- Sistema simples de batalhas em turnos (jogador vs zumbi).
- Inventário básico (adição, remoção e listagem de itens).
- Mapa representado em matriz textual (# = obstáculo, Z = zumbi, P = player).
- Salvamento e carregamento de progresso via **arquivos texto/binários**.

Conceitos aplicados:

- **Struct** para personagem, itens e inimigos.
- **Funções** para modularização do código.
- **Ponteiros e malloc()** para manipular memória dinamicamente.
- **Arquivos (fopen, fwrite, fread)** para salvar progresso.
- **Variáveis globais** para estados de jogo (nível, HP, XP).

5. Parte Técnica - Estruturas, Algoritmos e Ferramentas:

Recursos do projeto		
 Componentes	 Estrutura / Algoritmo	 Aplicação
Mapa	Matriz (2D array)	Representação dos locais e movimentação.
Inventário	Vetor Dinâmico	Armazenamento e manipulação de itens coletados.
Sistema de Combate	Estrutura de Fila	Gerenciamento de turnos (jogador e inimigos alternam ações).
Inimigo	Algoritmos de busca (BFS, A*)	Inimigos perseguem o jogador no mapa.
Ordenação de itens	Quick Sort / Bubble Sort	Organização de inventário por tipo, peso ou valor.
Salvamento	Estruturas de registro (struct) + arquivos binários	Persistência de dados do jogador, itens e mapas.

6. Possibilidades de Expansão:

- **Novo sistema de crafting** (criação de armas e itens).
- **Mapa procedural** (gerado aleatoriamente com grafos).
- **Mais classes e habilidades especiais.**
- **Diálogo com NPCs e decisões de narrativa.**
- **Modo sobrevivência infinita.**
- **Sistema de missões e conquistas.**

Essas adições permitem crescimento contínuo do projeto e aplicação de novos conceitos de estruturas e algoritmos.

7. Seção Bônus - Texto explicativo sobre a ideia do jogo:

O jogo se passa em um futuro devastado, onde a humanidade enfrenta o colapso após um vírus transformar grande parte da população em criaturas sedentas por carne. O jogador, Alex, desperta em meio ao caos e precisa lutar pela sobrevivência.

A primeira versão do jogo traz uma **experiência de sobrevivência estratégica por texto**, onde o jogador escolhe ações, enfrenta zumbis e gerencia recursos de forma tática. Essa etapa foca no aprendizado dos fundamentos da linguagem C — manipulação de estruturas, ponteiros, funções e arquivos — tudo dentro de uma narrativa envolvente.

Na segunda fase, o projeto ganha vida em **2D**, com movimentação livre pelo mapa, sprites e batalhas animadas em turnos. A inteligência artificial dos inimigos utiliza **busca em grafos para perseguir o jogador**, enquanto o inventário é gerido com **listas dinâmicas e sistemas de ordenação**.

A progressão entre as etapas reflete o avanço do desenvolvedor: de um sistema lógico simples em terminal, até uma arquitetura modular e escalável, com conceitos reais de engenharia de software e jogos digitais.

8. Resumo:

- **Tema:** RPG Pós-apocalíptico com zumbis
- **Tipo:** Single-player com batalhas em turnos
- **Etapa 1:** Jogo textual (fundamentos da linguagem C)
- **Etapa 2:** Jogo gráfico 2D (estruturas de dados e IA)
- **Ferramentas:** C padrão + SDL2 / Raylib
- **Foco:** Estratégia, aprendizado de algoritmos e modularização

9.Dividas Técnicas / Futuras Features:

9.1 Etapa 2 – Versão Gráfica 2D:

Objetivo: Evoluir o projeto para um jogo gráfico 2D com sprites e escalonável, aplicando estruturas e algoritmos.

Funcionalidades:

- Interface 2D (tiles, sprites e HUD).
- Controle por teclado (movimentação e ações).
- Sistema de combate em turnos visual com animações.
- IA básica para inimigos (seguir ou atacar jogador).
- Estrutura de dados para mapa, inventário e inimigos.
- Menu principal e sistema de save/load visual.

Ferramentas e bibliotecas:

- SDL2 ou Raylib → renderização 2D e controle de entrada.
- CMake → organização e compilação do projeto modular.
- Valgrind → análise de memória (debugging).

9.2 Dívida técnica:

Objetivo: Nas próximas etapas (Sprints) do projeto, serão realizadas correções e otimizações com o objetivo de tornar a estrutura do sistema mais organizada, eficiente e robusta. As melhorias serão conduzidas em conjunto com a equipe, garantindo a aplicação de boas práticas de desenvolvimento, padronização do código e maior estabilidade da funcionalidade.

Problemas:

- Durante o desenvolvimento, foram identificadas dificuldades na implementação do comando de **fuga dos zumbis**. Atualmente, ao selecionar essa opção, quando bem sucedida (probabilidade do evento ocorrer), o jogador consegue escapar de **todos os zumbis presentes no jogo simultaneamente**, o que não corresponde ao comportamento desejado. Essa limitação impacta o equilíbrio da mecânica de jogo e será ajustada nas próximas etapas para garantir uma funcionalidade mais realista e desafiadora.