

Desenvolvimento de um Sistema de Batalha RPG em Linguagem C: RPG Text

Henrique Ferreira, Fillippe Barcelos

Dr. Aldo Henrique Mendes (Orientador)

Centro Universitário Euro-Americano (Unieuro), Brasília, DF, Brasil

Resumo—Este trabalho apresenta o desenvolvimento de um sistema de RPG textual em linguagem C, denominado RPG Text. A solução implementa cadastro de personagens (Guerreiro, Mago, Troll, Orc), batalhas em turnos e exibição de status, aplicando structs, ponteiros e alocação dinâmica com malloc e free. A lógica de combate utiliza a fórmula $\text{dano} = \text{ataque} - \text{defesa}$ com valor mínimo de 1, em laço while alternando turnos. Os testes realizados mediram tempo médio de execução próximo a 2 segundos, consumo de memória inferior a 5 MB e ausência de erros de execução em cinco rodadas consecutivas. O projeto reforça a relevância da linguagem C no ensino de programação estruturada e gerenciamento manual de memória.

Palavras-chave—Linguagem C; RPG; Structs; Ponteiros; Alocação dinâmica.

I. INTRODUÇÃO

Jogos do tipo RPG (Role-Playing Game) são reconhecidos pela estrutura baseada em personagens, atributos e sistemas de combate. Sistemas de combate são elementos centrais para a experiência do usuário, responsáveis por grande parte do engajamento e da imersão [1].

A escolha da linguagem C justifica-se por sua eficiência, pelo controle de memória e pela proximidade com o hardware, características que a tornam adequada ao aprendizado de fundamentos da computação [2]. A implementação de sistemas de combate permite explorar lógica condicional, estruturas de repetição e organização modular, sendo ferramenta didática útil em cursos de tecnologia [3].

II. METODOLOGIA

O sistema foi estruturado em torno da struct Personagem, que armazena nome, pontos de vida, ataque e defesa. Estruturas permitem organizar dados heterogêneos de forma eficiente, facilitando a manipulação dos personagens durante a execução [2].

Para flexibilidade na gestão de memória, utilizou-se a função malloc para alocação dinâmica dos personagens em tempo de execução. A liberação é feita por free ao final do programa, prevenindo vazamentos de memória.

A lógica de combate baseia-se em turnos, implementada em laço while com alternância de ações. O cálculo de dano segue a fórmula $\text{dano} = \text{ataque} - \text{defesa}$, com valor mínimo de 1. A passagem por referência via ponteiros reduz o consumo de pilha e melhora o desempenho [2].

III. RESULTADOS

O sistema demonstrou estabilidade em cinco execuções consecutivas, com comportamento consistente em múltiplas simulações de combate. A Tabela I sintetiza os resultados de tempo, consumo de memória e erros observados.

TABELA I
RESULTADOS DAS EXECUÇÕES

Execução	Tempo (s)	Memória (MB)	Erros
1	2,1	4,8	0
2	1,9	4,7	0
3	2,0	4,9	0
4	2,2	5,0	0
5	1,8	4,6	0

A liberação de memória com free garantiu a integridade do processo, sem ocorrência de vazamentos. As Figuras 1 a 3 ilustram, respectivamente, o menu de cadastro de classes, uma batalha em andamento (Surg vs Trundle) e o encerramento da batalha com declaração do vencedor.

IV. DISCUSSÃO

Os resultados indicam que a linguagem C é eficiente para o desenvolvimento de sistemas baseados em lógica estruturada, como jogos de RPG textuais. A alocação dinâmica e o uso de ponteiros proporcionam controle sobre os recursos computacionais, ainda que exijam maior atenção do programador quando comparados a linguagens de alto nível [2].

O modelo de combate em turnos mostrou-se simples e eficaz, com possibilidade de expansão futura para inclusão de inteligência artificial nos oponentes, interface gráfica e sistema de progressão de personagem [1], [3].

V. CONCLUSÃO

O desenvolvimento do RPG Text demonstrou ser uma abordagem eficaz tanto técnica quanto educacional. O projeto permitiu a aplicação prática de structs, ponteiros, alocação dinâmica e lógica de turnos em laço while. O sistema mostrou-se estável e expansível, evidenciando que a linguagem C continua relevante para o desenvolvimento de aplicações com requisitos de eficiência.

REFERÊNCIAS

- [1] K. Salen e E. Zimmerman, Rules of Play: Game Design Fundamentals. Cambridge, MA: MIT Press, 2004.
- [2] B. W. Kernighan e D. M. Ritchie, The C Programming Language, 2ª ed. Englewood Cliffs, NJ: Prentice Hall, 1988.
- [3] T. Fullerton, Game Design Workshop, 4ª ed. Boca Raton: CRC Press, 2018.
- [4] H. M. Deitel e P. J. Deitel, C: How to Program, 8ª ed. Boston: Pearson, 2016.
- [5] E. Adams, Fundamentals of Game Design, 3ª ed. Berkeley, CA: New Riders, 2014.