



**UNIVERSIDADE FEDERAL DE SERGIPE
DEPARTAMENTO DE SISTEMA DE INFORMAÇÃO**

DOCUMENTAÇÃO DO JOGO JOKENPO

**Itabaiana – Sergipe
2025**

DOCUMENTAÇÃO DO JOGO JOKENPO

Documentação técnica referente ao desenvolvimento do jogo Jokenpo, elaborada para a disciplina de Programação II do Departamento de Sistema de Informação da Universidade Federal de Sergipe.

Autores: David de Jesus, Gustavo Santiago de Oliveira, Izaque dos Santos, João Pereira, João Victor Meneses Souza

Conteúdo

1	Introdução	3
2	Arquitetura do Software	3
2.1	Módulo Game ('game.h' e 'game.c')	3
2.1.1	Estruturas de Dados	3
2.1.2	Funções Principais	4
2.2	Módulo Ranking ('ranking.h' e 'ranking.c')	4
2.2.1	Estruturas de Dados	4
2.3	Módulo Principal ('main.c')	5
2.3.1	Funções Principais	5
3	Funcionalidades Implementadas	5
3.1	Animação de Abertura	5
3.2	Menu Principal	6
3.3	Lógica do Jogo	6
3.4	Gerenciamento de Ranking	6
4	Considerações Técnicas	6
4.1	Uso da Biblioteca 'gconio.h'	6
4.2	Alocação Dinâmica de Memória	7
4.3	Persistência de Dados	7
5	Conclusão	7

1 INTRODUÇÃO

Este documento apresenta a documentação técnica detalhada do jogo Jokenpo (Pedra, Papel e Tesoura), desenvolvido em linguagem C. O objetivo é descrever a arquitetura do software, as funcionalidades implementadas, a estrutura do código-fonte e as decisões de design tomadas durante o processo de desenvolvimento.

O jogo Jokenpo é uma implementação clássica do popular jogo de mãos, onde o jogador compete contra um adversário controlado pelo computador (bot). As funcionalidades incluem seleção e cadastro de jogadores, registro de pontuações em um ranking e exibição de regras. A interface do usuário é baseada em console, utilizando a biblioteca ‘gconio.h’ para manipulação de tela e cores, proporcionando uma experiência interativa.

2 ARQUITETURA DO SOFTWARE

A arquitetura do jogo Jokenpo é modular, dividida em quatro componentes principais: ‘main’, ‘game’, ‘ranking’ e ‘gconio’, com a interface gráfica baseada em console. Cada módulo tem responsabilidades distintas: ‘main’ inicializa e executa o jogo; ‘game’ gerencia a lógica principal e a interface visual com o usuário; ‘ranking’ armazena, manipula e persiste os dados dos jogadores; ‘gconio’ provê funções gráficas de console. A comunicação entre os módulos é realizada através de inclusão de arquivos de cabeçalho (‘.h’) e variáveis globais, quando necessário.

2.1 Módulo Game (‘game.h’ e ‘game.c’)

O módulo ‘game’ é o coração do jogo, responsável pela lógica principal, interação com o usuário e controle do fluxo do jogo. Ele define as estruturas de dados para as escolhas (Pedra, Papel, Tesoura) e os resultados das partidas (Vitória, Derrota, Empate). As funções neste módulo gerenciam a exibição de menus, a animação de abertura, a interface de jogo e a determinação do vencedor de cada rodada.

2.1.1 Estruturas de Dados

O arquivo ‘game.h’ define dois tipos enumerados (‘enum’) para representar as escolhas dos jogadores e os resultados das partidas:

- **Escolha:** Representa as opções que um jogador pode fazer: Pedra, Papel ou Tesoura.

```
typedef enum { PEDRA = 1, PAPEL, TESOURA } Escolha;
```

- **Resultado:** Representa o desfecho de uma rodada: Vitória, Derrota ou Empate.

```
typedef enum { VITORIA, DERROTA, EMPATE } Resultado;
```

2.1.2 Funções Principais

As principais funções do módulo ‘game’ incluem:

- **void jogar()**: Função principal que orquestra o fluxo do jogo, incluindo a exibição do menu, a interação com o usuário e a chamada das demais funções de jogo.
- **void animacao_abertura()**: Exibe uma animação inicial e uma tela de carregamento ao iniciar o jogo.
- **const char* obter_nome_escolha(Escolha escolha)**: Retorna o nome da escolha (Pedra, Papel, Tesoura) em formato de string.
- **Resultado determinar_vencedor(Escolha jogador, Escolha bot)**: Determina o resultado de uma rodada com base nas escolhas do jogador e do bot.
- **void carregar_ranking()**: Carrega os dados do ranking de jogadores de um arquivo.
- **void salvar_ranking()**: Salva os dados do ranking de jogadores em um arquivo.
- **void cadastrar_jogador()**: Permite o cadastro de um novo jogador.
- **void liberar_memoria()**: Libera a memória alocada dinamicamente para o ranking.

2.2 Módulo Ranking (‘ranking.h’ e ‘ranking.c’)

O módulo ‘ranking’ é responsável pelo gerenciamento dos dados dos jogadores e suas pontuações. Ele define a estrutura ‘Jogador’ e gerencia o armazenamento e a recuperação desses dados, utilizando alocação dinâmica de memória para o ranking. Embora o arquivo ‘ranking.c’ não tenha sido fornecido, as funções de manipulação do ranking estão presentes no ‘game.c’, indicando uma integração direta.

2.2.1 Estruturas de Dados

O arquivo ‘ranking.h’ define a estrutura ‘Jogador’:

- **Jogador**: Armazena o nome e a pontuação de um jogador.

```
typedef struct {
    char nome[50];
    int pontuacao;
} Jogador;
```

Além disso, declara variáveis globais para o gerenciamento do ranking:

- **‘extern Jogador *ranking’**: Ponteiro para o array de jogadores.
- **‘extern int totalJogadores’**: Número total de jogadores cadastrados.
- **‘extern int capacidadeJogadores’**: Capacidade atual do array de jogadores.
- **‘extern int jogadorAtual’**: Índice do jogador atualmente selecionado.

2.3 Módulo Principal (‘main.c’)

O módulo ‘main’ é o ponto de entrada do programa. Ele é responsável por inicializar a interface do console, chamar a função principal do jogo (‘jogar()’) e liberar a memória alocada ao final da execução. Também configura o título da janela do console e o conjunto de caracteres para exibição correta de caracteres especiais.

2.3.1 Funções Principais

As principais funções do módulo ‘main’ incluem:

- **int main()**: Função principal que inicia a execução do jogo.
- **void inicializar_interface()**: Configura o ambiente do console, como o título da janela, o tamanho e as cores de fundo e texto.

3 FUNCIONALIDADES IMPLEMENTADAS

O jogo Jokenpo oferece as seguintes funcionalidades principais:

3.1 Animação de Abertura

Ao iniciar o jogo, uma animação de título e uma mensagem de boas-vindas são exibidas, seguidas por uma barra de carregamento. Esta funcionalidade é implementada pela função `animacao_abertura()` no arquivo `game.c`.

3.2 Menu Principal

O jogo apresenta um menu principal com as seguintes opções:

- **Jogar:** Inicia uma partida de Jokenpo contra o bot.
- **Selecionar Jogador:** Permite escolher um jogador existente no ranking.
- **Cadastrar Novo Jogador:** Permite registrar um novo jogador.
- **Ver Ranking:** Exibe a lista de jogadores e suas pontuações, ordenada de forma decrescente.
- **Regras:** Mostra as regras básicas do jogo.
- **Sair:** Encerra o jogo, salvando o ranking atual.

3.3 Lógica do Jogo

As partidas são jogadas em rodadas, onde o jogador escolhe entre Pedra, Papel ou Tesoura, e o bot faz uma escolha aleatória. O resultado é determinado pelas regras clássicas do Jokenpo. A pontuação do jogador é atualizada após cada rodada (vitória +1, derrota -1, empate 0). O jogador pode optar por jogar novamente ou retornar ao menu principal.

3.4 Gerenciamento de Ranking

O sistema de ranking permite cadastrar novos jogadores, carregar jogadores existentes e salvar as pontuações. O ranking é persistido em um arquivo de texto ('ranking.txt'). A memória para o ranking é alocada dinamicamente e realocada conforme a necessidade, garantindo eficiência no uso de recursos.

4 CONSIDERAÇÕES TÉCNICAS

4.1 Uso da Biblioteca 'gconio.h'

A biblioteca 'gconio.h' é utilizada para manipulação da interface do console, permitindo o uso de funções como 'clrscr()' para limpar a tela, 'gotoxy()' para posicionar o cursor, 'textcolor()' e 'textbackground()' para definir cores, e 'getch()' para leitura de caracteres sem a necessidade de pressionar Enter. Esta biblioteca é fundamental para a experiência visual e interativa do jogo.

4.2 Alocação Dinâmica de Memória

O ranking de jogadores é gerenciado com alocação dinâmica de memória ('malloc' e 'realloc'), o que permite que o número de jogadores não seja fixo. Isso garante que o sistema possa lidar com um número variável de jogadores de forma eficiente, expandindo a capacidade de armazenamento conforme novos jogadores são cadastrados.

4.3 Persistência de Dados

Os dados do ranking são salvos em um arquivo de texto ('ranking.txt') no formato CSV (Comma Separated Values), onde cada linha representa um jogador com seu nome e pontuação. Isso garante que as pontuações sejam mantidas entre as sessões do jogo.

5 CONCLUSÃO

O jogo Jokenpo, desenvolvido em C, demonstra uma arquitetura modular e a implementação de funcionalidades essenciais para um jogo de console interativo. A utilização de estruturas de dados apropriadas, alocação dinâmica de memória e persistência de dados contribui para um sistema robusto e eficiente. A documentação detalhada, visa fornecer uma compreensão clara do projeto, servindo como referência para futuras manutenções ou expansões.