

Documentação de Jogo com Allegro 5: SCdoku

Artur Camera Segat¹, Enzo Augusto Mucha Domingues¹, Izabely Brum Rodrigues¹

¹Universidade Federal de Santa Maria (UFSM)
Av. Roraima nº 1000, bairro Camobi, Santa Maria - RS, 97105-900

Abstract. *This work presents the development of a digital game written in the C programming language using the Allegro 5 graphics library. The project was designed to apply concepts of modular programming, event handling, graphical rendering, and data persistence in an interactive environment. The game implements a structure based on multiple rooms (menu, game, settings, history, victory, and defeat), each with its own input and drawing logic. In addition, the system records gameplay history in local files, storing information such as game time, difficulty, errors, and final result. The project aims not only to strengthen low-level programming skills but also to serve as a foundation for studying 2D game development. The results show a functional, stable, and extensible game, with potential for future improvements in graphics, sound, and gameplay.*

Resumo. *Este trabalho apresenta o desenvolvimento de um jogo digital em linguagem C utilizando a biblioteca gráfica Allegro 5. O projeto foi idealizado com o objetivo de aplicar conceitos de programação modular, tratamento de eventos, manipulação gráfica e persistência de dados em um ambiente interativo. O jogo implementa uma estrutura de múltiplas salas (menu, jogo, configurações, histórico, vitória e derrota), cada uma com lógica de entrada e renderização própria. Além disso, o sistema registra o histórico de partidas em arquivos locais, armazenando informações como tempo de jogo, dificuldade, erros e resultado final. O projeto visa não apenas reforçar habilidades em programação de baixo nível, mas também servir como base para o estudo de desenvolvimento de jogos 2D. Os resultados demonstram um jogo funcional, estável e extensível, com possibilidades futuras de aprimoramento gráfico, sonoro e de jogabilidade.*

1. Introdução

O desenvolvimento de jogos digitais é uma área que envolve múltiplas habilidades da computação, desde programação de baixo nível até design gráfico e estruturação de interfaces. O projeto **SCdoku** tem como objetivo principal o desenvolvimento de um jogo baseado em Sudoku utilizando a linguagem C e a biblioteca Allegro 5, com foco no aprendizado de estruturas modulares, manipulação gráfica e tratamento de eventos.

Ao propor a construção de um jogo completo, o projeto busca reforçar o conhecimento dos desenvolvedores sobre a organização de código em múltiplos arquivos, o uso de enums e estados, a estruturação de interfaces com interação via mouse e a leitura e escrita de arquivos para salvar o progresso e o histórico das partidas.

2. Desenvolvimento

O jogo foi desenvolvido utilizando a linguagem C e a biblioteca Allegro 5, uma ferramenta amplamente utilizada para o desenvolvimento de jogos 2D. O projeto segue uma estrutura modular que facilita a organização do código, a leitura e a manutenção, sendo dividido em diversos diretórios que separam responsabilidades de forma clara:

- **src/game/**: contém a lógica central do Sudoku, como verificação de erros, controle do tempo da partida, geração e validação do tabuleiro, além da lógica de vitória e derrota.
- **src/draw/**: responsável por toda a renderização do jogo. Contém funções como `draw_menu_room()`, `draw_game_room()`, `draw_victory_room()`, entre outras, que são chamadas diretamente na `main.c`, dependendo do estado atual do jogo.
- **src/control/**: trata as entradas do usuário, em especial eventos de mouse. Funções como `handle_menu_events()`, `handle_game_events()`, entre outras, são invocadas também diretamente na `main.c`, e são responsáveis por detectar cliques, interações com botões e seleção de células e números.
- **src/save/**: implementa o sistema de persistência de dados, lidando com a leitura e escrita do histórico de partidas no arquivo `historico.txt`.
- **src/online/**: camada preparada para implementar funcionalidades de jogo em rede, como conexão entre dois jogadores via IP. Utiliza sockets para troca de dados, permitindo partidas multiplayer.
- **src/states/**: define as principais estruturas de dados utilizadas ao longo do jogo, incluindo o `Game`, `GameState`, `Cell` e enumeradores de estado da aplicação.

O fluxo principal do jogo é controlado no arquivo `main.c`, que inicializa a biblioteca Allegro, configura o display, carrega os recursos e entra no loop principal de eventos. Esse loop identifica o estado atual do jogo (armazenado em `GameState`) e chama as funções de manipulação de eventos (`handle_...`) e de renderização (`draw_...`) de acordo com a sala correspondente.

O jogo é baseado em uma máquina de estados bem definida, onde cada “sala” representa uma fase do fluxo do jogo. Os estados são representados por um enumerador:

- `ROOM_MENU` – Tela inicial com opções de iniciar o jogo, histórico e configurações.
- `ROOM_CONFIG` – Permite ajustes como sons e cores (em expansão).
- `ROOM_DIFFICULTY` – Seleção de dificuldade: fácil, médio ou difícil.
- `ROOM_GAME` – Tela do jogo com o tabuleiro do Sudoku.
- `ROOM_HISTORY` – Exibe uma tabela com partidas anteriores.
- `ROOM_VICTORY` – Tela exibida ao vencer o jogo.
- `ROOM_LOSE` – Tela exibida ao exceder o número de erros permitido.

Cada sala possui lógica específica encapsulada em suas funções de controle e desenho. Por exemplo, `handle_game_events()` é responsável por capturar o clique do usuário, verificar se uma célula foi selecionada, se um número foi escolhido e se a jogada foi válida. Já `draw_game_room()` renderiza o tabuleiro, a seleção de números, a quantidade de erros e o tempo de jogo.

O tabuleiro é construído dinamicamente com base em arquivos ou geradores de nível, dependendo da dificuldade selecionada. A grade é composta por células (`Cell`)

que armazenam valor, se é fixa ou jogada pelo usuário, e se há erro. A interface apresenta uma barra de seleção de números de 1 a 9, com destaque visual para o número atualmente selecionado (em azul), os números completos (cinza escuro) e os ainda faltantes (cinza claro), permitindo ao jogador identificar rapidamente suas opções.

O sistema de histórico salva os dados de cada partida concluída em `historico.txt`, incluindo:

- Data e hora da partida;
- Dificuldade selecionada;
- Tempo total de jogo;
- Quantidade de erros cometidos;
- Resultado final (vitória ou derrota).

Essas informações são utilizadas posteriormente para compor a sala de histórico, apresentada em uma interface gráfica em formato de tabela. A leitura e escrita são feitas utilizando operações padrão de arquivos em C.

Por fim, a estrutura modular do projeto permite que novas salas e funcionalidades sejam integradas facilmente. As chamadas diretas às funções de `draw` e `handle` a partir da `main` promovem controle explícito e facilitam o rastreamento de bugs durante o desenvolvimento e testes.

3. Como Jogar

O **SCdoku** oferece dois modos principais de jogo: *Solo* e *Online*.

3.1. Modo Solo

No modo solo, o jogador enfrenta o desafio de resolver um tabuleiro de Sudoku dentro do menor tempo possível e com o menor número de erros. Para jogar:

1. Inicie o jogo e selecione “**SOLO**” no menu principal.
2. Escolha o nível de dificuldade: fácil, médio ou difícil.
3. Use o mouse para selecionar uma célula e depois escolher um número entre 1 e 9 na barra inferior.
4. O número será inserido na célula selecionada, a menos que seja uma célula fixa (pré-definida).
5. O jogo termina em vitória ao preencher corretamente todas as células ou em derrota ao exceder o número máximo de erros.

3.2. Modo Online

O modo online permite que dois jogadores joguem juntos em tempo real. Um jogador atua como servidor (host) e o outro como cliente (conectando ao host). Para jogar:

1. Um dos jogadores deve iniciar o jogo e selecionar a opção “**MULTIPLAYER**” e deve colocar o IP do oponente.
2. O outro jogador deve escolher “**MULTIPLAYER**” e inserir o IP do host.
3. Para testar em uma mesma máquina, utilize o IP `127.0.0.1`.
4. Ambos os jogadores devem aguardar a conexão ser estabelecida.
5. O jogo se inicia automaticamente após a conexão e sincroniza as jogadas entre os dois participantes.

O modo online exige que ambos estejam na mesma rede local ou que o host esteja com a porta de conexão liberada (caso jogando remotamente).

4. Resultados

O projeto resultou em um jogo funcional, estável e visualmente organizado. As principais conquistas incluem:

- Interface gráfica intuitiva, com botões de menu e sistema de navegação fluido.
- Estrutura de código modular, favorecendo a manutenção e expansão.
- Armazenamento automático de partidas no histórico.
- Verificação de vitória e derrota, com telas dedicadas.
- Sistema visual de seleção de números e células.

Os testes realizados demonstraram que o jogo responde corretamente aos eventos do usuário e que os dados do histórico são registrados de forma adequada. A renderização se manteve estável mesmo em configurações com resoluções variadas.

5. Conclusão

O projeto **SCdoku** proporcionou aos desenvolvedores uma experiência completa no ciclo de desenvolvimento de jogos 2D utilizando C e Allegro 5. Além de consolidar conhecimentos de lógica, modularidade e manipulação gráfica, o jogo serviu como base para aprimorar habilidades em organização de código, persistência de dados e controle de eventos.

O projeto está disponível para expansão e serve como exemplo prático de aplicação dos conhecimentos de programação adquiridos ao longo da formação acadêmica.

6. Referências

Knuth, D. E.

The TeXbook. Addison-Wesley, 1984.

Allegro 5 – Game Programming Library.

Disponível em: <https://liballeg.org/>. Acesso em: jul. 2025.

Kernighan, B. W.; Ritchie, D. M.

The C Programming Language. 2. ed. Prentice Hall, 1988.

Nystrom, R.

Game Programming Patterns. Genever Benning, 2014.

Allegro 5 Tutorials.

Disponível em: https://wiki.allegro.cc/index.php?title=Allegro_5_Tutorials. Acesso em: jul. 2025.