

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
SÃO PAULO**

MATHEUS AUDIBERT

COMBO KEY

**CAMPOS DO JORDÃO
2025**

RESUMO

Este trabalho apresenta o desenvolvimento do jogo eletrônico COMBO KEY, cujo objetivo é proporcionar uma experiência interativa baseada em reflexo e memorização de sequências. Desenvolvido em linguagem C++ com o uso da biblioteca gráfica Raylib e o editor Visual Studio Code, o jogo desafia o jogador a pressionar, em ordem e no tempo correto, uma sequência aleatória de teclas exibidas na tela. A cada acerto, o jogador acumula pontos e aumenta seu combo, sendo penalizado ao errar qualquer tecla. O jogo oferece três níveis de dificuldade, que se diferenciam pelo tempo de resposta disponível para cada tecla, aumentando gradualmente o desafio. A metodologia adotada baseou-se na aplicação prática de conceitos fundamentais da programação de jogos, como entrada via teclado, controle de tempo, animação por spritesheets e gerenciamento de estados. O projeto foi desenvolvido de forma iterativa, com foco na modularização do código e na clareza estrutural, facilitando testes e futuras expansões. Como resultado, o jogo mostrou-se funcional, estável e adequado à proposta, demonstrando a viabilidade da linguagem C++ e da biblioteca Raylib como ferramentas eficazes para projetos educacionais. Conclui-se que o desenvolvimento do COMBO KEY proporcionou uma experiência prática enriquecedora e contribuiu para o aprimoramento das habilidades de programação e design de jogos digitais.

Palavras-Chave: Desenvolvimento de jogos; Raylib; C++; Reflexos; Sequência de teclas.

ABSTRACT

This work presents the development of the electronic game COMBO KEY, which aims to provide an interactive experience based on reflexes and memorization of sequences. Developed in C++ using the Raylib graphics library and the VS Code editor, the game challenges the player to press, in the correct order and within the time limit, a random sequence of keys displayed on the screen. Each correct input increases the player's score and combo, while any mistake resets the combo. The game features three difficulty levels, which vary the time available to respond, progressively increasing the challenge. The methodology applied focused on the practical implementation of fundamental game development concepts, such as keyboard input handling, time control, sprite-based animation, and state management. The project was developed iteratively, with emphasis on code modularization and structural clarity, facilitating testing and future enhancements. As a result, the game proved to be functional, stable, and consistent with its objectives, demonstrating the viability of C++ and Raylib as effective tools for educational projects. It is concluded that the development of COMBO KEY provided a rich practical experience and contributed to the improvement of programming and game design skills.

Keywords: Game development; Raylib; C++; Reflexes; Key sequences.

LISTA DE ILUSTRAÇÕES

FIGURA 1 – Plano de Fundo (Autor, 2025)	14
FIGURA 2 – Sprite de Tecla (Autor, 2025)	14
FIGURA 3 – Menu Principal (Autor, 2025)	18
FIGURA 4 – Jogo Iniciado (Autor, 2025)	19
FIGURA 5 – Game Over (Autor, 2025)	20

SUMÁRIO

1	INTRODUÇÃO _____	6
1.1	Objetivos _____	6
1.2	Justificativa _____	7
1.3	Aspectos Metodológicos _____	8
1.4	Aporte Teórico _____	9
2	FUNDAMENTAÇÃO TEÓRICA _____	10
2.1	Ferramentas Utilizadas _____	10
2.2	Programação Orientada a Objetos _____	10
3	PROJETO PROPOSTO _____	12
3.1	Concepção e Planejamento do Jogo _____	12
3.2	Recursos Visuais e Sonoros _____	13
3.3	Descrição do Processo de Desenvolvimento _____	14
4	AVALIAÇÃO _____	16
4.1	Resultados _____	17
5	CONCLUSÃO _____	21
	REFERÊNCIAS _____	22

1 INTRODUÇÃO

O desenvolvimento de jogos eletrônicos tem se consolidado como uma das áreas mais dinâmicas e multidisciplinares da computação, integrando conhecimentos de lógica de programação, design gráfico, interação com o usuário e inteligência artificial. Nesse contexto, o presente trabalho tem como proposta a criação de um jogo digital chamado **COMBO KEY**, desenvolvido em linguagem C++ com apoio da biblioteca gráfica **Raylib**. A escolha desse projeto se deu tanto pelo interesse pessoal em jogos quanto pela oportunidade de aplicar, de forma prática e integrada, diversos conceitos fundamentais da programação e do desenvolvimento de software.

O objetivo principal do projeto é desenvolver um jogo interativo baseado em reflexos e memorização, onde o jogador deve pressionar, em sequência e no tempo adequado, as teclas exibidas na tela. A mecânica simples, porém desafiadora, busca oferecer uma experiência envolvente e educativa, além de explorar aspectos como controle de tempo, entrada de dados via teclado, animação de elementos gráficos e gerenciamento de estados do jogo. O jogo ainda conta com três níveis de dificuldade, que modificam o tempo de resposta permitido, tornando a experiência progressivamente mais desafiadora.

1.1 Objetivos

O principal objetivo deste projeto foi desenvolver o jogo eletrônico **COMBO KEY**, que desafia o jogador a reproduzir sequências de teclas em tempos determinados, estimulando reflexos e memória. Durante o desenvolvimento, buscou-se aplicar conceitos fundamentais de programação, incluindo controle de tempo, manipulação de eventos de teclado e gerenciamento de estados do jogo, além de aprimorar habilidades na utilização da biblioteca gráfica **Raylib** em C++. Além disso, o projeto proporcionou uma experiência prática na criação de uma arquitetura modular e organizada, visando facilitar futuras manutenções e expansões. Para alcançar esse objetivo geral, foram definidos os seguintes objetivos específicos:

- Investigar os princípios essenciais para o desenvolvimento de jogos que envolvem entrada e processamento de comandos do teclado, controle de temporização e animações visuais;
- Projetar e implementar uma estrutura de código clara e modular, promovendo boas práticas de desenvolvimento e possibilitando a evolução do projeto;
- Criar diferentes níveis de dificuldade, ajustando o tempo para a entrada correta das teclas, com o intuito de aumentar o desafio conforme o progresso do jogador;
- Converter o código C++ para JavaScript por meio da ferramenta Emscripten, possibilitando que o jogo seja executado diretamente em navegadores, ampliando seu alcance e acessibilidade sem necessidade de instalação;
- Realizar testes para garantir a estabilidade, usabilidade e aderência aos objetivos, proporcionando uma experiência satisfatória ao usuário final.

1.2 Justificativa

A relevância deste trabalho está fundamentada na crescente importância do desenvolvimento de jogos eletrônicos como ferramenta pedagógica e prática para o aprendizado de conceitos essenciais da programação e da engenharia de software. O projeto **COMBO KEY** se justifica pela sua capacidade de proporcionar uma experiência interativa que estimula habilidades cognitivas importantes, como a agilidade mental, a memória de curto prazo e a coordenação motora, ao desafiar o jogador a reproduzir sequências de teclas dentro de limites temporais. Além disso, a implementação do jogo utilizando C++ e a biblioteca gráfica **Raylib** permite aprofundar conhecimentos técnicos em programação orientada a objetos, manipulação de eventos, controle de tempo e renderização gráfica, habilidades amplamente requisitadas no mercado de trabalho tecnológico.

Outro aspecto relevante para a justificativa deste documento é a conversão do jogo para JavaScript por meio da ferramenta Emscripten, que amplia significativamente o alcance do projeto ao permitir sua execução em navegadores web, sem ne-

cessidade de instalação. Isso facilita o acesso de estudantes, educadores e entusiastas, democratizando o uso da aplicação e incentivando a experimentação e o aprendizado prático.

Por fim, este documento visa também servir como modelo e referência para trabalhos acadêmicos e projetos de desenvolvimento de jogos em contextos educacionais, especialmente para os alunos do IFSP Campos do Jordão, incentivando a produção técnica e o desenvolvimento de competências essenciais para a formação profissional na área de tecnologia da informação.

1.3 Aspectos Metodológicos

O desenvolvimento do jogo **COMBO KEY** adotou uma metodologia iterativa, iniciando com o estudo das mecânicas fundamentais de jogos que envolvem sequências de comandos e resposta rápida do jogador. A escolha do ambiente **Visual Studio Code**, aliado à biblioteca **Raylib**, proporcionou um ambiente leve e eficiente para a implementação gráfica e o tratamento dos eventos de teclado. O processo começou com a definição dos requisitos básicos, como o reconhecimento das teclas, o controle do tempo para pressionar as sequências e a criação da interface visual simples e clara.

Em seguida, foi desenvolvido um protótipo inicial focado na lógica de verificação das teclas pressionadas e na contagem dos combos. Testes constantes foram realizados para ajustar o tempo de resposta, garantir a fluidez das animações e a estabilidade do jogo. A modularização do código foi prioridade para facilitar futuras manutenções e a expansão do projeto, especialmente para implementar os diferentes níveis de dificuldade. A fase final contemplou também a conversão do código para JavaScript, utilizando Emscripten, possibilitando a execução do jogo diretamente em navegadores, sem necessidade de instalação. Todo o ciclo contou com avaliações contínuas para assegurar a qualidade e a aderência aos objetivos do projeto.

1.4 Aporte Teórico

Neste trabalho, apresentam-se de forma breve as principais bases teóricas que sustentam o desenvolvimento do jogo **COMBO KEY**. As fundamentações envolvem conceitos de programação orientada a objetos, controle de eventos de entrada via teclado, manipulação de temporizadores e renderização gráfica em 2D, com foco na biblioteca **Raylib**. Além disso, abordam-se as técnicas para conversão de código nativo C++ para JavaScript utilizando a ferramenta Emscripten, ampliando o acesso ao jogo em plataformas web. Na seção 2, essas bases teóricas serão exploradas com maior detalhamento, contextualizando os conceitos e trabalhos relacionados. A seção 3 apresenta a metodologia de desenvolvimento e a arquitetura do sistema implementado. Na seção 4, são descritos os testes e a avaliação do jogo, enquanto a seção 5 encerra com as conclusões do trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Embora o desenvolvimento do jogo COMBO KEY tenha se baseado principalmente na criatividade e na experimentação prática, algumas bases conceituais foram fundamentais para sua criação. Entre elas, destacam-se os princípios da programação orientada a objetos, essenciais para estruturar o código de forma organizada e modular. Além disso, conceitos básicos de manipulação de eventos de teclado e controle temporal foram aplicados para garantir a dinâmica do jogo e a interação com o usuário. A biblioteca gráfica **Raylib** ofereceu os recursos necessários para a renderização em 2D e o controle dos elementos visuais, facilitando o desenvolvimento. Por fim, a utilização da ferramenta Emscripten possibilitou a transposição do jogo para a web, ampliando seu alcance e acessibilidade. Esta fundamentação teórica, embora não baseada em pesquisas acadêmicas específicas, fundamentou as decisões técnicas adotadas durante o desenvolvimento do projeto.

2.1 Ferramentas e Bibliotecas

A biblioteca gráfica **Raylib** foi escolhida pela sua facilidade de uso e bom desempenho para desenvolvimento de jogos 2D, permitindo a criação das mecânicas e a renderização dos elementos visuais de forma simples e eficiente. O código foi escrito em C++ utilizando o **Visual Studio Code**, que oferece um ambiente leve e flexível, com suporte para depuração e extensões úteis para programação. Para possibilitar a execução do jogo na web, foi utilizada a ferramenta Emscripten, que compila o código C++ para JavaScript, eliminando a necessidade de instalação pelo usuário.

2.2 Programação Orientada a Objetos

A programação orientada a objetos foi adotada para estruturar o código de forma organizada e modular, facilitando a manutenção e a expansão do jogo. Essa abordagem permitiu separar as funcionalidades em classes específicas, como gerenciamento das sequências de teclas e controle do tempo para pressioná-las. A OOP é amplamente recomendada em projetos de software por promover clareza, reutilização e escalabilidade do código.

3 PROJETO PROPOSTO

A metodologia adotada para o desenvolvimento do jogo **COMBO KEY** em C++ seguiu uma abordagem sistemática e organizada, visando garantir a eficiência do processo e o atendimento aos objetivos definidos. O desenvolvimento foi estruturado em fases, com planejamento detalhado para assegurar que cada etapa estivesse alinhada aos requisitos técnicos e às demandas do projeto.

O processo iniciou-se com a concepção do jogo, na qual foram estabelecidos os objetivos centrais, as funcionalidades essenciais e os elementos de design a serem implementados. Nesta etapa, foram analisadas as mecânicas de jogo, o desempenho esperado e as prioridades em termos de jogabilidade, interação e aspectos visuais. Essa definição clara dos elementos essenciais proporcionou uma visão coesa do projeto, permitindo que as fases seguintes fossem adaptadas às necessidades reais do desenvolvimento.

Também foram selecionadas as ferramentas e tecnologias mais adequadas para o projeto: a linguagem C++ para a programação, o **Visual Studio Code** como ambiente de desenvolvimento e a biblioteca gráfica **Raylib** para suporte visual. Além disso, a utilização do **Emscripten** possibilitou a conversão do jogo para uma versão web, ampliando seu alcance e facilitando o acesso dos usuários.

Essa fase inicial foi crucial para estabelecer a base do projeto, assegurando que o desenvolvimento fosse conduzido com organização, foco e clareza. Dessa forma, garantiu-se a criação de um produto funcional, estável e coerente com as expectativas, facilitando também a manutenção e futuras expansões.

3.1 Concepção e Planejamento do Jogo

A concepção do **COMBO KEY** partiu da proposta de desenvolver um jogo leve, rápido e desafiador, voltado para a memorização e agilidade na digitação de sequências de teclas. A ideia central foi criar uma experiência direta, com foco na habilidade do jogador em responder corretamente a comandos aleatórios em sequência.

Durante o planejamento, definiu-se uma estrutura simples composta por apenas três telas principais: o menu inicial, onde é possível iniciar ou sair do jogo; a tela de jogo, responsável por exibir as sequências de teclas e capturar as respostas do jogador; e a tela de game over, acionada apenas quando o jogador pressiona uma tecla incorreta, encerrando a tentativa atual. Essa estrutura objetiva garantir um fluxo ágil e intuitivo, sem distrações desnecessárias.

Os elementos visuais também foram planejados para reforçar a clareza e a resposta imediata do sistema às ações do jogador. As teclas são representadas por sprites animados, e cada acerto ou erro possui uma resposta visual direta. A escolha por uma mecânica de jogo enxuta permitiu focar nos aspectos fundamentais da experiência: reflexo, precisão e repetição.

Durante essa fase, também foram escolhidas as ferramentas de desenvolvimento mais adequadas ao projeto: C++, pela sua eficiência e controle de performance; **Raylib**, por ser uma biblioteca gráfica 2D de fácil integração e alto desempenho; e **Visual Studio Code**, pela sua leveza e compatibilidade com o ambiente de desenvolvimento proposto. Esse planejamento serviu como base para o restante do projeto, garantindo clareza, objetividade e consistência no desenvolvimento.

3.2 Recursos Visuais e Sonoros

Os elementos visuais e sonoros de **COMBO KEY** foram cuidadosamente selecionados para criar uma experiência coesa e envolvente. O plano de fundo é único em todas as telas do jogo, garantindo consistência visual e permitindo que o foco do jogador permaneça nas teclas e nas interações centrais da jogabilidade.

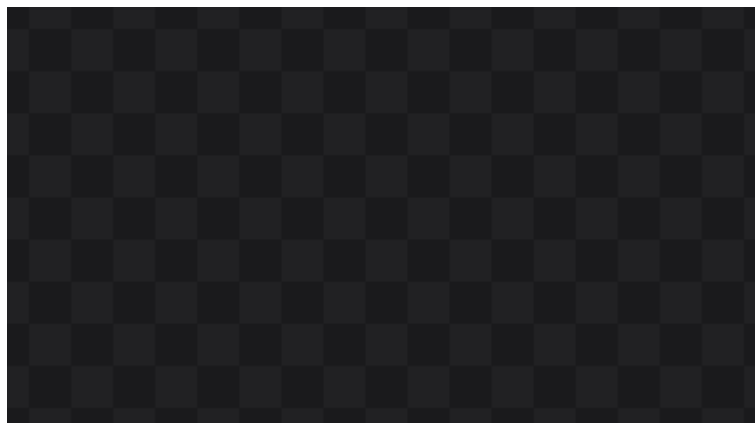


Figura 1 – Plano de Fundo (Autor, 2025)

As teclas que aparecem na tela são renderizadas a partir de um sprite padronizado, com variação apenas no caractere exibido (como letras, números ou setas). Essa padronização tornou mais simples a implementação e manutenção gráfica do jogo, mantendo uma estética clara e organizada.



Figura 2 – Sprite de Tecla (Autor, 2025)

No aspecto sonoro, **COMBO KEY** utiliza efeitos para reforçar o feedback das ações do jogador. Ao pressionar uma tecla correta, é reproduzido aleatoriamente um entre três sons distintos de tecla, oferecendo variedade sem perder a clareza do retorno auditivo. Um som com a pronúncia do nome do jogo ("Combo Key") é executado ao iniciar o programa, servindo como introdução e ambientação. Além disso, um efeito sonoro específico marca o início da partida, enquanto outro som é tocado quando o jogador comete um erro e a tela de game over é exibida. Esses sons contribuem para a imersão e ajudam a reforçar os momentos importantes da experiência de jogo.

3.3 Descrição do Processo de Desenvolvimento

1. Configuração do Ambiente de Desenvolvimento:

- O desenvolvimento teve início com a instalação do **Visual Studio Code**, juntamente com a extensão para C++ e a configuração do compilador g++.

- A biblioteca **raylib** foi baixada diretamente do site oficial e integrada manualmente ao projeto, incluindo os diretórios de cabeçalho e os arquivos de link necessários para compilação.

2. Estruturação do Projeto e Código Base:

- O projeto foi iniciado com a criação de arquivos separados para organizar as principais partes do jogo, como a renderização, os controles e a lógica de teclas.
- As estruturas fundamentais, como a representação de teclas, sprites, e a enumeração dos estados do jogo (Menu, Jogo, Game Over), foram definidas nessa etapa.

3. Implementação das Funcionalidades do Jogo:

- A lógica de exibição das telas (menu, jogo e game over) foi organizada por meio de um sistema de estados, facilitando a troca entre elas com clareza e controle.
- O jogo principal foi implementado com uma sequência de teclas aleatórias, que o jogador deve pressionar corretamente. Caso erre, o estado muda para "Game Over".

4. Implementação dos Assets Visuais e Sonoros:

- O plano de fundo foi aplicado de forma estática e usado em todas as telas do jogo.
- Sprites de teclas foram padronizados, alterando-se apenas o caractere exibido.
- Foram integrados efeitos sonoros distintos para os momentos de início do jogo, acerto de tecla (escolhido aleatoriamente entre três sons), e fim de jogo.
- Também foi adicionado um áudio de introdução dizendo o nome do jogo ("Combo Key") na tela inicial.

5. Testes e Refinamento:

- O jogo foi testado continuamente em ciclos curtos de desenvolvimento, com foco na estabilidade, desempenho e fidelidade da mecânica de combo.
- Pequenas correções e melhorias visuais foram aplicadas ao longo do processo, garantindo uma experiência interativa consistente e funcional.

6. Transformação para JavaScript e Hospedagem:

- Após o desenvolvimento da versão em C++, o jogo foi convertido para JavaScript utilizando a ferramenta Emscripten, permitindo a execução diretamente em navegadores web.
- Essa transformação possibilitou que o jogo fosse acessado facilmente por qualquer pessoa, sem a necessidade de instalação ou configuração adicional.
- Para garantir ampla disponibilidade e facilidade de acesso, o jogo foi hospedado na plataforma **Vercel** (<https://combokey.audibert.dev>), que oferece deploy rápido e confiável de aplicações web.

4 AVALIAÇÃO

O jogo Combo Key demonstrou que suas principais funcionalidades foram implementadas com sucesso. A troca dinâmica das teclas, a detecção correta da sequência de comandos e as transições entre as telas funcionaram conforme o esperado. Os efeitos sonoros contribuíram para o feedback do jogador, tornando a experiência mais envolvente.

Durante os testes, o jogo apresentou estabilidade, boa resposta aos comandos e clareza na proposta. A mecânica de combinação de teclas foi considerada intuitiva, embora alguns participantes tenham sugerido um aumento gradual na dificuldade para prolongar o desafio.

De maneira geral, o projeto atingiu os objetivos propostos, entregando uma experiência simples, funcional, divertida e está disponível em: <https://combokey.audibert.dev> . Com base nas observações coletadas, há espaço para futuras melhorias, como a adição de fases, placares ou variação nos visuais, o que pode tornar o jogo ainda mais interessante.

4.1 Resultados

O jogo foi desenvolvido em cinco arquivos principais: `main.cpp`, `game_logic.cpp`, `game_logic.h`, `key_button.cpp` e `key_button.h`, permitindo uma separação clara entre lógica, estrutura e exibição. Durante os testes, todas as funcionalidades essenciais responderam corretamente: o menu inicial, a execução do jogo e a transição para a tela de game over ocorreram sem falhas.

A geração de combinações aleatórias de teclas e a verificação de entradas foram realizadas com precisão. Quando uma tecla correta era pressionada, um dos três efeitos sonoros era reproduzido aleatoriamente, reforçando a resposta ao jogador. Sons distintos também foram tocados no início da partida e ao final do jogo, além da vinheta com o nome “Combo Key” na introdução.

Visualmente, o plano de fundo permaneceu constante durante todo o jogo, e os sprites das teclas, embora representando letras e símbolos distintos, seguiram o mesmo modelo gráfico. Isso garantiu uma aparência uniforme e coerente.

De modo geral, o jogo se comportou como esperado, com bom desempenho, estabilidade e fidelidade às mecânicas propostas.



Figura 3 – Menu Principal (Autor, 2025)

Menu principal apresentando um layout simples e direto, com destaque para o título centralizado e dois botões interativos: ENTER para iniciar o jogo e BACKSPACE para sair, além de botões de + para alterar a dificuldade do jogo. Essa tela funciona como ponto de partida para a experiência do jogador, introduzindo a identidade visual do projeto.

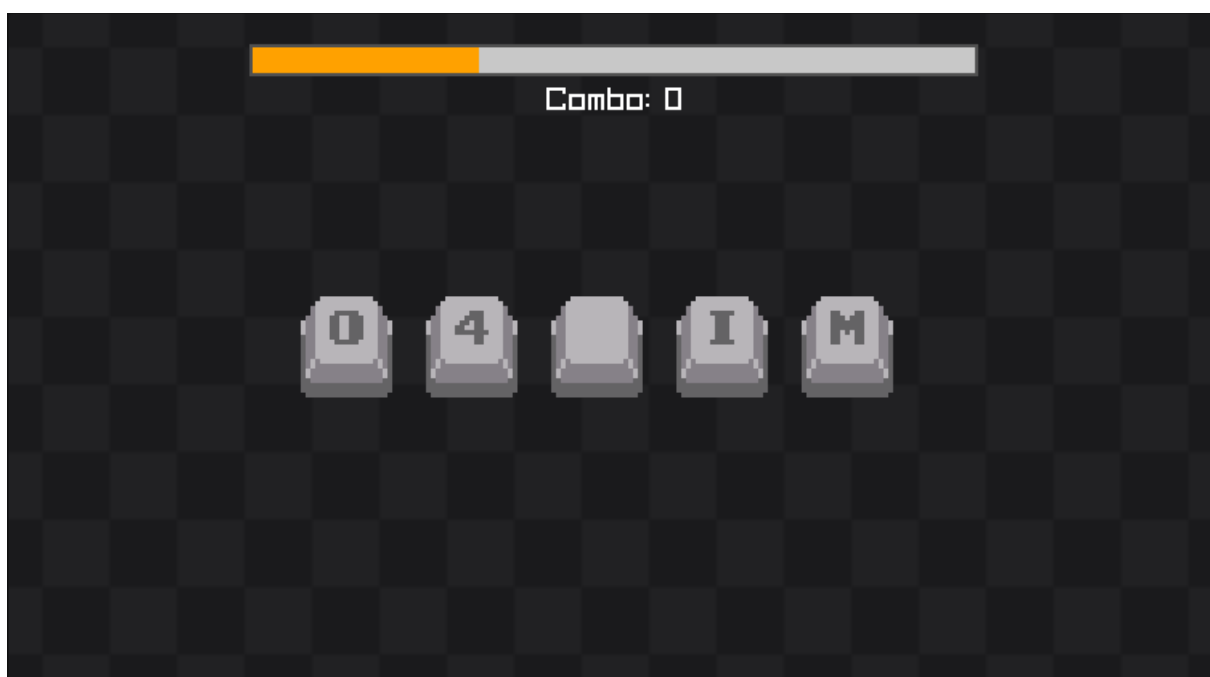


Figura 4 – Jogo Iniciado (Autor, 2025)

Jogo iniciado, exibindo no centro da tela a sequência aleatória de teclas que o jogador deve pressionar. No topo, é apresentado o contador de combos, que aumenta conforme o jogador acerta as teclas corretamente, além de um temporizador que limita o tempo de resposta, tornando a jogabilidade mais dinâmica e desafiadora.



Figura 5 – Game Over (Autor, 2025)

Game Over, exibindo a mensagem central de encerramento da partida. Abaixo, três instruções informam as ações possíveis: pressionar Enter para reiniciar o jogo, Esc para retornar ao menu principal e Backspace para encerrar a aplicação. A interface mantém o estilo visual do jogo, oferecendo uma transição clara e intuitiva para o jogador.

5 CONCLUSÃO

Com base no desenvolvimento e na avaliação realizada, conclui-se que o projeto do jogo "Combo Key" atingiu seus objetivos propostos. O jogo foi implementado com sucesso utilizando C++ e a biblioteca gráfica raylib, resultando em uma aplicação funcional, com mecânicas de jogo bem definidas e uma interface responsiva.

Durante o processo, foi possível aplicar na prática conceitos de programação orientada a objetos, lógica de jogo e manipulação de entrada do usuário. Além disso, a adaptação do jogo para a web por meio da conversão para JavaScript e sua hospedagem na Vercel ampliaram seu alcance e acessibilidade.

As funcionalidades implementadas, como a geração aleatória de teclas, a contagem de combos e os efeitos sonoros, demonstraram estar estáveis e bem integradas, proporcionando uma experiência interativa e envolvente para o jogador.

Como aprimoramentos futuros, destaca-se a possibilidade de adicionar novos modos de jogo, rankings online e desafios diários, com o intuito de aumentar o fator replay e atrair mais usuários.

Em resumo, o desenvolvimento do "Combo Key" foi uma experiência enriquecedora, que consolidou conhecimentos técnicos e práticos em programação de jogos, oferecendo uma base sólida para projetos ainda mais complexos no futuro.

REFERÊNCIAS

A. ONLINE:

BMO. *Making A Game in C With Raylib*. [S.l.]: YouTube, 2025. Disponível em: <https://www.youtube.com/watch?v=wwBktq791Qo>. Acesso em: 1 jun. 2025.

EMSCRIPTEN-CORE. *Emsdk*. Disponível em: <https://github.com/emscripten-core/emsdk>. Acesso em: 8 jun. 2025.

EMSCRIPTEN. *Docs*. Disponível em: <https://emscripten.org/docs/>. Acesso em: 8 jun. 2025.

PROGRAMMING WITH NICK. *Compile C/C++ raylib game Into WebAssembly*. [S.l.]: YouTube, 2023. Disponível em: <https://www.youtube.com/watch?v=j6akryezlzc>. Acesso em: 8 jun. 2025.

RAYLIB. *Cheatsheet*. Disponível em: <https://www.raylib.com/cheatsheet/cheatsheet.html>. Acesso em: 7 jun. 2025.

RAYLIB. *Examples*. Disponível em: <https://raylib.com/examples.html>. Acesso em: 7 jun. 2025.

RAYSAN5. *Raylib*. Disponível em: <https://github.com/raysan5/raylib>. Acesso em: 7 jun. 2025.