

Desenvolvimento de Age of Runner: Um Jogo de Plataforma 2D em C++ utilizando SDL2

Pedro Henrique Santos Kusiak¹, Vinicius Koiti Nakahara¹,
Eduardo Henrique Molina da Cruz¹, Hélio Toshio Kamakawa¹

¹Instituto Federal do Paraná (IFPR) – Campus Paranavaí
Paranavaí – PR – Brasil

pedro01kusiak@gmail.com, viniciusnakahara@gmail.com

eduardo.cruz@ifpr.edu.br, helio.kamakawa@ifpr.edu.br

1. Introdução

O mercado de jogos eletrônicos no Brasil tem se destacado como um dos setores de maior crescimento na economia de entretenimento. Em 2021, o país se tornou o maior mercado de videogames da América Latina, superando o México, com uma receita de US\$ 1,4 bilhão, que pode dobrar até 2026, atingindo US\$ 2,8 bilhões [Sebrae 2023]. Este crescimento reflete uma base de jogadores diversificada e crescente, impulsionada por fatores como a popularização de dispositivos móveis e uma maior aceitação cultural dos jogos [PwC Brasil 2022].

Embora os jogos para celular tenham dominado o mercado, os jogos para desktop ainda mantêm uma relevância significativa. Foi com esse cenário em mente que desenvolvemos o jogo *Age of Runner*. O desenvolvimento de jogos de plataforma 2D continua sendo uma escolha relevante para desenvolvedores independentes, pois permite focar em mecânicas inovadoras mantendo custos de produção controlados [Nystrom 2014].

Neste projeto, optamos por implementar um motor de física próprio em vez de utilizar soluções prontas, permitindo maior controle sobre as mecânicas do jogo e proporcionando uma experiência de aprendizado mais profunda sobre os fundamentos da física em jogos digitais [Eberly 2010].

2. Metodologia

Este trabalho foi desenvolvido no âmbito das disciplinas de Tópicos em Computação e Projeto de Software Avançado, seguindo uma abordagem metodológica iterativa com entregas trimestrais predefinidas.

O desenvolvimento seguiu os princípios de arquitetura hexagonal [Vernon 2013], permitindo maior isolamento entre os componentes do sistema e facilitando a manutenibilidade do código. A escolha desta arquitetura foi fundamentada na necessidade de separar claramente as responsabilidades entre o motor do jogo, a lógica de negócio e a interface do usuário.

O primeiro trimestre do desenvolvimento concentrou-se na implementação da estrutura fundamental do sistema, com foco na elaboração das classes base e no desenvolvimento do motor de física proprietário. Nesta fase, priorizou-se a utilização eficiente das estruturas de dados da STL e a implementação da jogabilidade básica, utilizando representações geométricas simples para os elementos visuais do jogo.

Durante o segundo trimestre, o foco deslocou-se para a camada de apresentação e experiência do usuário. Foram implementados os sistemas de áudio, incluindo efeitos sonoros e música ambiente, além da integração de *sprites* e suas respectivas animações. Nesta etapa, também foi desenvolvido o sistema de pontuação e finalizadas as mecânicas centrais do jogo.

A implementação do motor de física próprio possibilitou o desenvolvimento de mecânicas fundamentadas em conceitos da física newtoniana, permitindo manipular as forças aplicadas aos elementos do jogo. Esta abordagem viabilizou a criação de interações únicas entre o jogador e o ambiente, as quais serão detalhadas na seção de desenvolvimento (Seção 3).

3. Desenvolvimento

O sistema de física implementado permite dois estados gravitacionais distintos: normal (9.8 m/s^2) e invertida ($-9,8 \text{ m/s}^2$). Esta mecânica afeta diretamente a movimentação do personagem e sua interação com os elementos do cenário.

O personagem principal responde aos comandos do jogador para movimentação invertendo a gravidade. A alteração da gravidade ocorre através de pressionar barra de espaço, permitindo adaptação às situações apresentadas pelo jogo.

O sistema de colisões implementado garante que o personagem interaja corretamente com as plataformas em ambos os estados gravitacionais. A detecção de colisão utiliza cálculos vetoriais para determinar a posição exata do personagem em relação aos elementos do cenário.

O paradigma de programação orientada a objetos fundamentou a estrutura do projeto, com ênfase na aplicação dos princípios SOLID e práticas de código limpo. A arquitetura do sistema foi concebida seguindo o modelo hexagonal, visando maximizar o desacoplamento entre os componentes e facilitar futuras manutenções e expansões.

4. Resultados

O projeto resultou em um jogo de plataforma 2D que explora mecânicas de inversão de gravidade. A interface inicial do jogo, apresentada na Figura 1, oferece um menu principal intuitivo com opções para iniciar um novo jogo e sair da aplicação. O design minimalista da interface busca proporcionar uma experiência de usuário direta e eficiente.

O sistema de movimentação desenvolvido integra o deslocamento tradicional com a mecânica de inversão gravitacional. Como pode ser observado na Figura 2, o jogo apresenta um ambiente com plataformas e obstáculos estrategicamente posicionados, criando desafios que exigem *timing* preciso e planejamento estratégico do jogador.

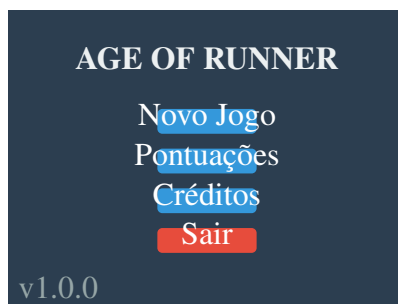


Figura 1. Menu Principal do Age of Runner.



Figura 2. Captura de Tela do Gameplay do Age of Runner.

5. Conclusão

O desenvolvimento do Age of Runner demonstrou a viabilidade de criar um jogo de plataforma 2D robusto utilizando C++ e SDL2, sem depender de frameworks específicos para jogos. A implementação bem-sucedida de um motor de física próprio, com mecânicas de inversão de gravidade, evidencia que é possível desenvolver sistemas complexos mantendo controle total sobre o comportamento do jogo. A arquitetura hexagonal adotada provou-se eficaz, proporcionando um código modular e de fácil manutenção, características essenciais para futuras expansões do projeto.

Embora nossa implementação personalizada tenha permitido um controle sobre a detecção de colisão e física do personagem, reconhecemos que motores estabelecidos como Unity e Phaser poderiam ter reduzido o tempo de desenvolvimento e oferecido ferramentas de depuração mais robustas. Esta experiência nos proporcionou conhecimento sobre as vantagens e limitações de cada abordagem para futuros projetos.

Referências

- Eberly, D. H. (2010). *Game physics*. CRC Press, Boca Raton, FL, 2 edition.
- Nystrom, R. (2014). *Game programming patterns*. Genever Benning.
- PwC Brasil (2022). Entertainment and media outlook 2022-2026. Acessado em: 25 de outubro de 2024.
- Sebrae (2023). Mercado de games: tendências e oportunidades. Acessado em: 25 de outubro de 2024.
- Vernon, V. (2013). *Implementing domain-driven design*. Addison-Wesley Professional.