

# Tower Building: Projeto e Desenvolvimento de um Jogo em C++

Yuri Gauze<sup>1</sup>, Eduardo Henrique Molina da Cruz<sup>1</sup>, Helio Toshio Kamakawa<sup>1</sup>

<sup>1</sup>Instituto Federal do Parana - Campus Paranavaí (IFPR)  
Paranavaí – PR – Brasil

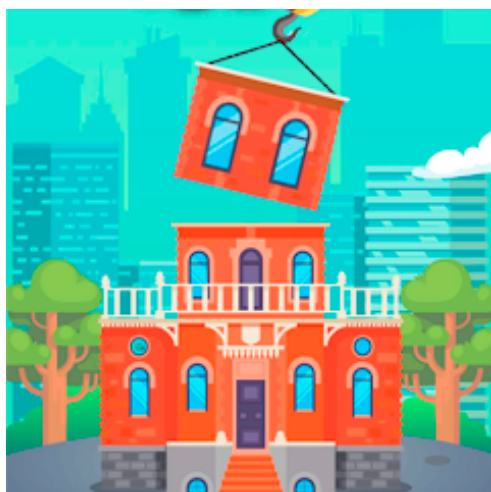
yuri159gauze@gmail.com, {eduardo.cruz, helio.kamakawa}@ifpr.edu.br

O mercado de desenvolvimento de jogos eletrônicos é um dos setores mais dinâmicos e lucrativos da indústria do entretenimento, tendo alcançado receitas que superam as da indústria cinematográfica e musical combinadas. Em 2023, o mercado global de jogos foi avaliado em aproximadamente US\$ 384 bilhões e deve crescer ainda mais nos próximos anos, com projeções de atingir US\$ 485 bilhões até 2028 [Newzoo 2023]. Este crescimento é impulsionado não apenas pela evolução tecnológica, que possibilita experiências de jogo mais imersivas e acessíveis, mas também pela diversificação de plataformas, incluindo consoles, dispositivos móveis e realidade virtual [Statista 2023]. Além disso, a indústria de jogos eletrônicos também tem um impacto significativo na economia, gerando milhões de empregos em diversas áreas, desde programação e design até marketing e produção [Canário et al. 2024][GamesIndustry.biz 2023].

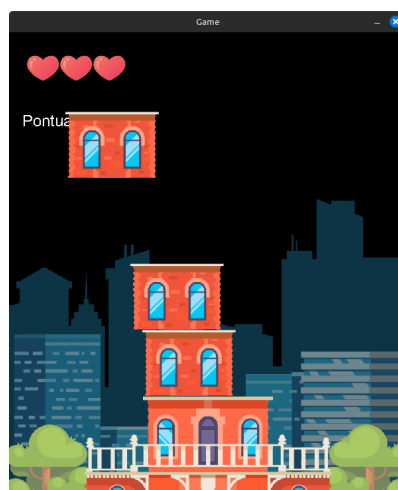
Este trabalho tem como objetivo desenvolver um jogo usando uma linguagem de programação orientada a objetos (POO), seguindo os princípios de POO e SOLID. O código será modular e claro, utilizando herança e polimorfismo apropriadamente. Também se aplicará o princípio de código limpo, focando na legibilidade e manutenção, e adotando arquiteturas robustas e escaláveis, como a Arquitetura Hexagonal ou Limpa.

*Tower Building* é um jogo de construção de torres em 2D que combina estratégia, física realista [Yuri Gauze 2024]. Os jogadores são desafiados a construir estruturas verticais utilizando blocos, enquanto enfrentam a gravidade e a dinâmica de equilíbrio. Com uma interface simples, intuitiva e controles simples, o jogo permite que os usuários empilhem os blocos de forma criativa, buscando a altura máxima possível sem que a torre desmorone. À medida que os níveis avançam, novos desafios e obstáculos são introduzidos, como a aceleração da velocidade, tornando cada partida única. *Tower Building* oferece uma experiência divertida e simples para jogadores de todas as idades.

Na Figura 1 e à Subfigura 1a ilustra a visão estética e funcional esperada para o jogo *Tower Building*. Embora a versão atual do jogo, que pode ser observada na Subfigura 1b, não apresente exatamente esse visual, essa referência demonstra as características de design e a atmosfera desejadas. O objetivo é criar um ambiente de jogo mais imersivo e visualmente atraente, que complemente a mecânica de construção e os desafios que os jogadores enfrentarão. Essa imagem reflete a evolução planejada para o jogo e as metas futuras de desenvolvimento.



(a) Resultado esperado.



(b) Resultado Atual.

Figura 1: Comparação entre o resultado esperado e o resultado atual.

A linguagem de programação escolhida para o desenvolvimento do jogo Tower Building é o C++, que suporta plenamente o paradigma de orientação a objetos (POO). Essa escolha é fundamental para atingir os conceitos de modularidade e reusabilidade do código, garantindo uma estrutura sólida e escalável. A implementação segue os princípios do SOLID, que ajudam a manter o código coeso e fácil de manter [Martin 2002]. O SOLID é composto por cinco princípios:

- **Single Responsibility Principle:** cada classe tem uma única responsabilidade, facilitando a manutenção;
- **Open/Closed Principle:** as classes estão abertas para extensão, mas fechadas para modificação, permitindo maior flexibilidade;
- **Liskov Substitution Principle:** classes derivadas podem substituir suas classes base sem alterar a funcionalidade;
- **Interface Segregation Principle:** as interfaces são específicas e não impõem métodos desnecessários;
- **Dependency Inversion Principle:** o código depende de abstrações, não de implementações concretas, promovendo maior desacoplamento.

O projeto, que está sendo desenvolvido nas disciplinas de Tópicos em Computação e Projeto de Software Avançado no curso de Engenharia de Software do Instituto Federal do Paraná - Campus Paranavaí, utiliza princípios de código limpo e a Arquitetura Hexagonal para garantir boa organização e manutenção. A implementação inclui bibliotecas como SDL para gráficos e eventos, SDL\_image para imagens, SDL\_mixer para áudio e Box2D para simulação física, visando uma experiência de jogo envolvente.

A Arquitetura Hexagonal, proposta por Alistair Cockburn em 2005, visa separar a lógica de negócios do núcleo da aplicação das dependências externas, como bancos de dados e interfaces de usuário. Utilizando "portos" para definir pontos de interação e "adaptadores" para integrar as tecnologias externas, essa abordagem facilita a testabilidade e manutenção, além de tornar a aplicação mais flexível e adaptável a mudanças tecnológicas [Cockburn 2005].

SDL (Simple DirectMedia Layer) é uma biblioteca de desenvolvimento de jogos que fornece uma abstração de baixo nível para acessar recursos de hardware, como gráficos, som e entrada do usuário, de forma simples e eficiente [SDL Community 2024]. Com suporte a múltiplas plataformas, a SDL facilita a criação de jogos e aplicações multimídia, permitindo que os desenvolvedores se concentrem na lógica do jogo sem se preocupar com as particularidades de cada sistema operacional.

Box2D é uma biblioteca de física 2D amplamente utilizada que simula o comportamento físico de objetos em jogos, permitindo a implementação de mecânicas realistas, como colisões e movimentos dinâmicos [Erin Catto 2024]. A combinação de SDL e Box2D oferece uma base poderosa para o desenvolvimento de jogos, possibilitando a criação de experiências imersivas e interativas.

O uso dessas bibliotecas foram fundamentais para a criação do jogo Tower Building, pois proporcionou os recursos necessários para o gerenciamento eficiente de gráficos, áudio e física. A SDL facilitou a implementação de uma interface interativa e envolvente, enquanto o Box2D permitiu a simulação realista de colisões e movimentos, elevando a qualidade da jogabilidade e garantindo uma experiência mais imersiva para os jogadores.

## Referências

- Canário, R. A., Oliveira, F. B. M., Gomes, I. S., and Silva, K. V. P. (2024). Influência do avanço do mercado de jogos eletrônicos para a economia brasileira. *Revista de Informação Contábil*.
- Cockburn, A. (2005). *Hexagonal Architecture: Ports and Adapters*. Independent Publisher, Chicago, IL.
- Erin Catto (2024). Box2d. Disponível em: <https://github.com/erincatto/box2d>. Acesso em: 19 out. 2024.
- GamesIndustry.biz (2023). The economic impact of the video game industry. Disponível em: <https://www.gamesindustry.biz/cheap-money-and-bad-bets-how-the-games-industry-turned-pandemic-success-into-disaster>. Acesso em: 20 out. 2024.
- Martin, R. C. (2002). *Agile Software Development, Principles, Patterns, and Practices*. Prentice Hall.
- Newzoo (2023). Global games market report. Disponível em: <https://newzoo.com/resources/trend-reports/newzoo-global-games-market-report-2023-free-version>. Acesso em: 19 out. 2024.
- SDL Community (2024). Simple directmedia layer. Disponível em: <https://www.libsdl.org/>. Acesso em: 20 out. 2024.
- Statista (2023). Video game industry statistics. Disponível em: <https://www.statista.com/topics/868/video-games/#editorsPicks>. Acesso em: 20 out. 2024.
- Yuri Gauze (2024). Tower building. Disponível em: <https://github.com/yurigauze/Tower-Building>. Acesso em: 19 out. 2024.