

Desenvolvimento de Archero Clone: Um Jogo de Ação 2D em C++ utilizando SDL2

Santyero Santos¹, Eduardo Henrique Molina da Cruz¹, Hélio Toshio Kamakawa¹

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

santyero@gmail.com, eduardo.cruz@ifpr.edu.br, helio.kamakawa@ifpr.edu.br

1. Introdução

O mercado de jogos eletrônicos tem se destacado como um dos setores mais dinâmicos da indústria de entretenimento global, gerando bilhões em receita e impulsionando inovações tecnológicas significativas [Telang et al. 2023]. Além do entretenimento, os jogos expandiram sua influência para educação, treinamento profissional e terapias de saúde [Hernandez and Ortega 2010]. A indústria tem sido um catalisador para avanços em computação gráfica e realidade virtual, criando milhões de empregos e formando comunidades globais de jogadores [Pavleas et al. 2013]. O impacto cultural dos jogos é igualmente notável, influenciando a arte, a música e a narrativa contemporânea, além de formar comunidades globais de jogadores que compartilham experiências e conhecimentos.

Este trabalho apresenta o desenvolvimento do “Archer0 Clone”, um jogo de ação 2D em C++ utilizando SDL2. O projeto implementa mecânicas de tiro e movimentação integradas, sistemas de colisão e comportamento de inimigos. O objetivo foi criar uma base sólida para desenvolvimento de jogos, demonstrando boas práticas de programação e servindo como referência para futuros projetos na área. A escolha do desenvolvimento de um jogo inspirado em um título existente permite focar nos aspectos técnicos e de implementação, utilizando um design de jogo já validado pelo mercado.

2. Metodologia

O desenvolvimento foi realizado sem o uso de *Engines* em C++ utilizando a biblioteca SDL2 para o gerenciamento de janelas e eventos, complementada pelo SDL2_image para manipulação de texturas e *sprites*, e SDL2_mixer para o sistema de áudio. Essa combinação de tecnologias foi escolhida por sua eficiência e amplo suporte no desenvolvimento de jogos 2D, oferecendo controle de baixo nível necessário para a implementação de mecânicas precisas de jogo. A arquitetura do jogo foi estruturada utilizando padrões de design comuns na indústria de jogos, como *Singleton* para gerenciadores globais e *State* para controle dos diferentes estados do jogo (menu, jogo, pausa).

O sistema de física do jogo foi implementado de forma simplificada, focando nas colisões entre personagens e projéteis, além de garantir movimentação fluida dos personagens através de um sistema de vetores. A renderização dos elementos visuais utiliza um sistema de camadas para correta ordenação dos elementos em tela, e o gerenciamento de recursos foi otimizado para manter boa performance durante o jogo. A interface do usuário foi desenvolvida com foco em minimalismo e clareza, garantindo que o jogador tenha sempre acesso às informações necessárias sem poluição visual excessiva.

3. Desenvolvimento

O “Archer Clone” é um jogo de ação 2D inspirado no título *mobile* Archer, incorporando mecânicas de tiro dinâmicas em um ambiente desafiador. O jogador controla um arqueiro que deve enfrentar inimigos em um nível de combate, onde cada decisão de movimento e posicionamento é crucial para a sobrevivência. A mecânica principal implementa um sistema onde tanto o protagonista quanto os inimigos disparam projéteis automaticamente quando estacionários, criando uma dinâmica única de “atirar ou mover” que força o jogador a tomar decisões táticas constantes sobre posicionamento e *timing*.

Durante o desenvolvimento, um dos principais desafios técnicos enfrentados foi a implementação do sistema de colisão. A solução adotada priorizou a simplicidade e eficiência, utilizando uma abordagem de verificação direta entre objetos. O sistema compara as posições e dimensões dos elementos para determinar se um objeto está contido dentro do espaço de outro, oferecendo um equilíbrio adequado entre precisão e performance para as necessidades do jogo.

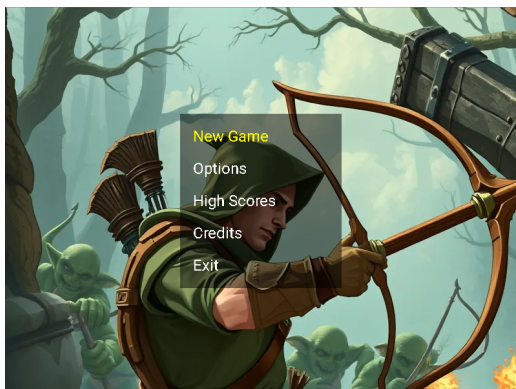
Outro desafio significativo encontrado foi no gerenciamento de memória, particularmente no carregamento e manipulação de *sprites*. A implementação inicial carregava e montava as *sprites* a cada nova animação, resultando em uso excessivo de memória e processamento. Para resolver este problema, foi implementado um sistema de pré-carregamento de *sprites* utilizando *unique_ptr* do C++, um ponteiro inteligente que garante propriedade exclusiva do recurso.

Os inimigos foram projetados em duas categorias principais: estáticos, que permanecem em posições fixas atirando em intervalos regulares, e móveis, que alternam entre movimentação e ataques. Esta variedade cria situações de combate dinâmicas que exigem diferentes estratégias do jogador. O design visual adota estilo *cartoon* 2D, priorizando clareza durante o combate, com interface minimalista e informativa. O sistema de combate incentiva o pensamento estratégico, pois o jogador deve constantemente avaliar quando é mais vantajoso atacar ou se movimentar, considerando múltiplos inimigos simultaneamente.

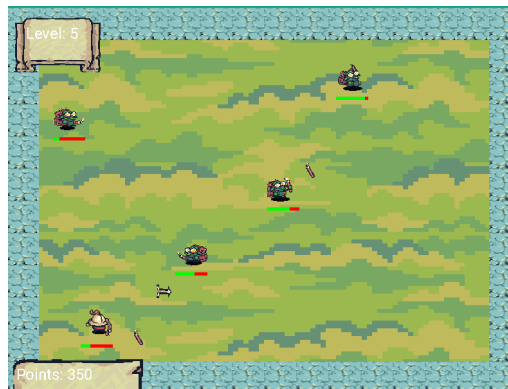
4. Resultados

O projeto resultou em um jogo que reproduz as mecânicas principais do Archer original com foco na jogabilidade e performance. A implementação do sistema de gerenciamento de *sprites* com *unique_ptr* demonstrou-se altamente eficaz, pois este ponteiro inteligente gerencia automaticamente a liberação de memória quando o objeto não é mais necessário. Esta abordagem não apenas reduziu significativamente o consumo de memória, mas também melhorou a performance geral do jogo ao evitar carregamentos redundantes de recursos. O sistema de colisão simplificado provou-se adequado para as necessidades do projeto, proporcionando detecção funcional de interações entre objetos sem comprometer a performance.

A interface do jogo foi projetada com foco em simplicidade e clareza, com um menu principal (Figura 1(a)) contendo opções como novo jogo, melhores pontuações, créditos e saída. O layout de gameplay (Figura 1(b)) foi estruturado de forma a garantir uma navegação intuitiva e fácil compreensão dos elementos visuais, priorizando a usabilidade e a minimização da poluição visual.



(a) Menu Principal do Archero.



(b) Captura de Tela do Gameplay do Archero Clone.

Os testes realizados pelo desenvolvedor e alunos de outras instituições que participaram do evento do IFTECH, realizado no dia 22 de outubro de 2024 demonstraram que as otimizações implementadas resultaram em uma experiência fluida e responsiva. A separação adequada de recursos e o gerenciamento eficiente de memória contribuíram para um desempenho consistente durante as sessões de jogo, mesmo em situações com múltiplos inimigos e projéteis na tela.

5. Conclusão

O desenvolvimento do “Archero Clone” foi bem-sucedido tanto tecnicamente quanto educacionalmente, permitindo aplicação prática de conceitos de engenharia de software e desenvolvimento de jogos. Os principais desafios superados incluíram a implementação do sistema de física próprio e o comportamento dos inimigos. A escolha de desenvolver um jogo inspirado em um título existente provou-se uma estratégia eficaz de aprendizado, permitindo focar nos aspectos técnicos da implementação. Como trabalhos futuros, identificamos possibilidades de expansão como novos tipos de inimigos, múltiplos níveis e mecânicas de progressão.

Referências

- Hernandez, F. and Ortega, F. (2010). Eberos gml2d: a graphical domain-specific language for modeling 2d video games. In *Proceedings of the 2010 ACM Conference on Computer Graphics and Interactive Techniques*, page 4:1.
- Pavleas, J., Chang, J., Sung, K., and Zhu, R. (2013). *Learn 2D Game Development with C#*. Apress.
- Telang, S., Patki, A., Kale, A., Jadhav, P., and Mujumdar, G. (2023). Shooting game using unreal engine. *International Journal for Research in Applied Science and Engineering Technology*.