SPACE INVADERS MOBILE: IMPLEMENTAÇÃO DE UM JOGO 2D EM FLUTTER

Gustavo Pratti da Silva, João Paulo Domingos Barbêdo, Luciano Soares Gonçalves, Ana Paula Abrantes de Castro e Shiguemori

1Instituto Federal de Educação, Ciência e Tecnologia de São Paulo (IFSP) – Campus Jacareí, Jacareí, Brasil.
E-mail: gustavo.prati@aluno.ifsp.edu.br
2Instituto Federal de Ciência, Educação e Tecnologia (IFSP), Jacareí-SP, Brasil

Resumo: O Space Invaders Mobile é um jogo desenvolvido em Flutter e Dart, inspirado no clássico arcade. O projeto traz uma releitura moderna para dispositivos móveis, preservando as mecânicas originais em uma experiência responsiva. O sistema conta com integração em nuvem via Firebase para o armazenamento de um ranking global. O projeto une conceitos de desenvolvimento mobile e jogos digitais, englobando gerenciamento de estados, colisões e renderização gráfica de forma prática.

Palavras-chave: Flutter; Dart; jogos mobile; desenvolvimento de jogos; Space Invaders.

INTRODUÇÃO

Os jogos eletrônicos para dispositivos móveis vêm ganhando cada vez mais espaço no mercado de software, ampliando a demanda por ferramentas multiplataforma voltadas ao desenvolvimento de aplicações interativas (SAKAI et al., 2022). Nesse contexto, frameworks inicialmente criados para aplicações móveis e corporativas passaram a ser utilizados também na criação de jogos casuais bidimensionais, ampliando as possibilidades do desenvolvimento móvel (Flutter, 2026).

O desenvolvimento de jogos digitais exige uma estrutura contínua de atualização, conhecida como Game Loop, responsável pelo processamento de física, colisões, renderização gráfica e comandos do usuário em tempo real (Gregory, 2018). Para favorecer a fluidez visual e a experiência do jogador, essas atualizações precisam ocorrer de forma constante e otimizada, considerando dispositivos móveis com diferentes capacidades de hardware (Nystrom, 2014).

Além disso, serviços de computação em nuvem têm sido cada vez mais utilizados em aplicações móveis, permitindo armazenamento de dados, sincronização online e integração entre usuários de maneira prática (GUERRA, 2021). Nesse cenário, plataformas Backend-as-a-Service (BaaS), como o Firebase, contribuem para o desenvolvimento de sistemas com maior facilidade de integração, persistência de dados e escalabilidade (Firebase, 2026).

Diante desse contexto, foi desenvolvido o Space Invaders Mobile, uma releitura do clássico Space Invaders adaptada para dispositivos móveis. O projeto preserva elementos tradicionais do jogo original, como ondas progressivas de inimigos e sistema de pontuação, incorporando recursos voltados ao contexto móvel, como interface responsiva, controles otimizados para toque e integração com armazenamento em nuvem. O objetivo deste trabalho é apresentar o desenvolvimento do jogo, destacando sua arquitetura, as tecnologias utilizadas e o uso do Flutter como alternativa para o desenvolvimento de jogos bidimensionais em dispositivos móveis.

MATERIAL E MÉTODOS

A metodologia deste estudo foi organizada em três etapas principais: definição da arquitetura de software, coleta de métricas de desempenho e análise da persistência de dados. A pesquisa caracteriza-se como experimental, com abordagem quantitativa, baseada na análise do código-fonte e no monitoramento dos ciclos de execução do sistema (Gil, 2021).

O sistema analisado consiste em uma releitura funcional do jogo Space Invaders, desenvolvida em Dart, utilizando o framework Flutter. O ranking global dos jogadores foi armazenado no banco de dados NoSQL Cloud Firestore, enquanto as informações de sessão foram gerenciadas localmente por meio da biblioteca shared_preferences. As principais tecnologias utilizadas no desenvolvimento do sistema estão apresentadas na Tabela 1.

Tabela 1. Tecnologias utilizadas no desenvolvimento do sistema

Tecnologia	Finalidade
Flutter	Desenvolvimento da interface e renderização
Dart	Linguagem principal do projeto
Firebase	Integração de serviços em nuvem
Cloud Firestore	Armazenamento do ranking global
shared_preferences	Persistência local de dados

A arquitetura do jogo foi estruturada em módulos responsáveis pela interface, pelo Game Loop e pelo gerenciamento das entidades do sistema. O ciclo principal foi implementado com o recurso Timer.periodic, responsável pela atualização da física, verificação de colisões e renderização dos elementos gráficos em intervalos de aproximadamente 16 milissegundos. Esse intervalo foi adotado com o objetivo de favorecer a fluidez visual e a estabilidade da aplicação durante a execução do jogo (Gregory, 2018).

A pesquisa considerou a hipótese de que o aumento do número de entidades ativas poderia impactar o desempenho do sistema. Para verificar essa hipótese, foram realizados testes de desempenho e monitoramento do uso de memória durante múltiplas ondas consecutivas de inimigos, avaliando-se a estabilidade do jogo em diferentes cenários de execução (Haukoos; Lewis, 2005).

RESULTADOS E DISCUSSÃO

Os testes realizados demonstraram a viabilidade do motor de jogo desenvolvido em Flutter, com desempenho estável e execução fluida durante as ondas de invasores. A velocidade dos inimigos foi ajustada dinamicamente conforme o avanço das rodadas, promovendo aumento gradual da dificuldade e contribuindo para o engajamento do jogador (Nystrom, 2014). A Figura 1 apresenta a interface principal do jogo em execução.

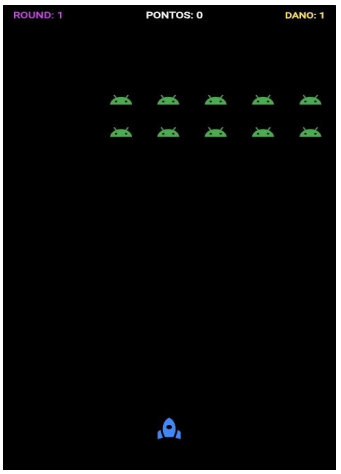

Figura 1. Gameplay principal do Space Invaders Mobile

A detecção de colisões apresentou respostas rápidas e eficientes, permitindo a remoção imediata das entidades quando sua vida era totalmente reduzida. Esse comportamento contribuiu para a estabilidade do sistema, pois evitou a permanência de objetos desnecessários em memória e reduziu o volume de processamento durante a execução.

Os testes também indicaram bom desempenho da arquitetura reativa do Flutter quando o ciclo de renderização foi mantido concentrado na tela principal do jogo (Flutter, 2026). Além disso, a integração com o Cloud Firestore possibilitou o armazenamento eficiente do ranking global, permitindo o registro das pontuações dos jogadores de forma centralizada, com organização dos dados e facilidade de consulta (Firebase, 2026). A Figura 2 demonstra o funcionamento do sistema de ranking integrado.

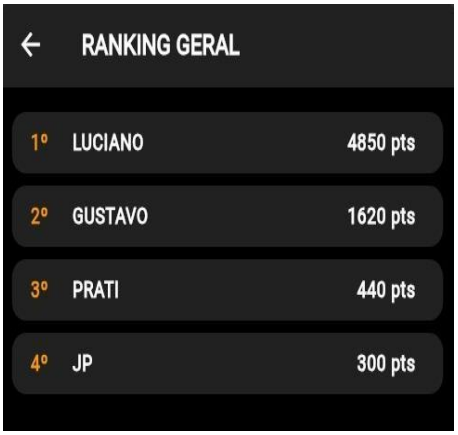

Figura 2. Sistema de ranking global integrado ao Firebase

A literatura destaca que jogos desenvolvidos em frameworks de interface exigem controle eficiente do ciclo de renderização para evitar consumo excessivo de recursos (Andrews, 2015; Gregory, 2018). Nesse contexto, o uso de proporções fixas auxiliou na manutenção da estabilidade visual da interface em diferentes tamanhos de tela, favorecendo uma experiência mais consistente ao usuário.

Por fim, observou-se que a responsividade dos comandos e a estabilidade do sistema dependem tanto do hardware utilizado quanto de uma arquitetura de software bem planejada (Pressman, 2021). Dessa forma, soluções otimizadas podem favorecer maior desempenho, melhor experiência do usuário e maior manutenibilidade do software (Sommerville, 2019).

CONCLUSÃO

Este estudo demonstrou que a integração entre um loop de física customizado em Flutter e a persistência de dados com Firebase constitui uma alternativa viável para o desenvolvimento de jogos mobile bidimensionais, mantendo desempenho estável durante a execução. Os resultados indicaram que o isolamento do ciclo de renderização e o uso de cálculos simples de colisão contribuíram para preservar a estabilidade do sistema e evitar quedas significativas de FPS.

Além disso, a pesquisa evidenciou que frameworks declarativos modernos podem ser utilizados no desenvolvimento de jogos retrô e aplicações interativas, ampliando seu uso para além das aplicações comerciais tradicionais. Entretanto, fatores como latência de rede e múltiplos acessos simultâneos ainda podem impactar o desempenho do sistema, o que indica a necessidade de estudos futuros voltados à avaliação de arquiteturas mais complexas, escaláveis e robustas.

REFERÊNCIAS

ANDREWS, Richard. Beginning Game Development with Python and Pygame. Apress, New York, 2015.

FLUTTER. Flutter Documentation: UI and State Management. Flutter Docs, Online, 2026.

GIL, Antonio Carlos. Métodos e Técnicas de Pesquisa Social. Atlas, São Paulo, 2021.

GREGORY, Jason. Game Engine Architecture. CRC Press, Boca Raton, 2018.

GUERRA, E. M. Componentes e Arquiteturas em Nuvem: Conceitos e Práticas. Casa do Código, São Paulo, 2021.

HAUKOOS, J. S.; LEWIS, R. J. Advanced statistics: bootstrapping confidence intervals for statistics with difficult distributions. Academic Emergency Medicine, v. 12, p. 360–365, 2005.

NYSTROM, Robert. Game Programming Patterns. Genever Benning, [S.l.], 2014.

PRESSMAN, Roger S.; MAXIM, Bruce R. Engenharia de Software: Uma Abordagem Profissional. AMGH, Porto Alegre, 2021.

SAKAI, L. et al. II Censo da Indústria Brasileira de Jogos Digitais. Homo Ludens / Abragames, São Paulo, 2022.

SOMMERVILLE, Ian. Software Engineering. Pearson, Boston, 2019.