

Desenvolvimento de um Controle Virtual Baseado em WebSocket para jogos de computador

Title: Development of a WebSocket-Based Virtual Controller for Computer Games

Maycon V. V. Fidelis¹, Icaro S. Ferreira¹, Victor B. D. S. Rosas¹

¹Centro Universitário Mario Pontes Jucá – UMJ
Maceió – AL – Brasil

{maycon.fidelis111, icaro.ferreira, victor.rosas}@umj.edu.br

Abstract. Introduction: Interaction with digital games occurs predominantly with dedicated physical hardware, causing limited access. With the advancement of communication protocols such as WebSocket and the popularization of smartphones, it is possible to explore new forms of interaction. **Objective:** Present the development of a virtual control that allows mobile devices to simulate computer video game controls, using low latency communication. **Steps:** The system was implemented as a client-server system that sends commands via WebSocket and the server converts them into inputs for video game control. **Results:** The system presented low Wi-Fi latency, enabling its use in games and showing potential as an affordable and portable solution for digital games.

Keywords: Virtual Controller, WebSocket, gamepad, digital games, latency.

Resumo. Introdução: A interação com jogos digitais ocorre predominantemente com hardware físico dedicado, causando uma limitação de acesso. Com o avanço dos protocolos de comunicação como o WebSocket e a popularização de smartphones, é possível explorar novas formas de interação. **Objetivo:** Apresentar o desenvolvimento de um controle virtual que permite dispositivos móveis simularem controles de videogame para computador, usando comunicação de baixa latência. **Etapas:** O sistema foi implementado como um sistema cliente-servidor que envia os comandos via WebSocket e o servidor converte em entradas para controle de videogame. **Resultados:** O sistema apresentou baixa latência em Wi-Fi, viabilizando seu uso em jogos e mostrando potencial como solução acessível e portátil no contexto jogos digitais.

Palavras-chave: Controle Virtual, WebSocket, gamepad, Jogos digitais, latência.

1. Introdução

A forma dos jogadores interagir com os jogos eletrônicos evoluiu significativamente ao longo dos anos, passando de controles com fio para soluções sem fio e interfaces baseadas em toque. Apesar dos avanços, o controle tradicional ainda depende de hardware dedicado, o que pode limitar o acesso e a flexibilidade do usuário.

Segundo Salen e Zimmerman (2004), “a interação lúdica significativa em um jogo surge da relação entre a ação do jogador e o desfecho do sistema; é o processo pelo qual

um jogador toma medidas no sistema projetado de um jogo e o sistema responde a essas ações com efeitos discerníveis e integrados ao contexto do jogo” (p. 38). Por isso o controle desempenha um papel fundamental na construção de experiências imersivas. Quando essa interface é pouco responsiva ou de difícil acesso, como o caso de controles físicos, a imersão dentro do jogo pode ser prejudicada. Isso reforça a importância de alternativas como controles virtuais, que podem trazer acessibilidade para esse tipo de interação.

Com o avanço dos smartphones e das comunicações em tempo real, tornou-se viável o desenvolvimento de um controle virtual, viabilizado pelo protocolo WebSocket, que oferece conexão persistente e de baixa latência.

Este trabalho apresenta o VirtualGamePad, um sistema que permite qualquer smartphone se comunicar com um computador para simular um controle de videogame. A solução utiliza um aplicativo móvel desenvolvido em React Native e um servidor Python, responsável por processar os dados e converter para comandos compatíveis com a biblioteca vgamepad. O objetivo é fornecer uma alternativa simples, portátil e de baixo custo, mantendo uma boa experiência de uso com respostas rápidas.

Embora já existam tentativas anteriores de utilizar dispositivos móveis como controladores de jogos, como no estudo de Vajk et al. (2008), que explorou o uso de celulares com sensores de movimento para interagir com jogos em grandes telas. No entanto, essa abordagem apresentava limitações, pois essa solução dependia de dispositivos específicos, APIs de sensores pouco acessíveis e de jogos projetados para esse tipo de interação. O VirtualGamePad amplia essa proposta por utilizar tecnologias modernas, como telas sensíveis ao toque e comunicação via WebSocket, além de ser compatível com qualquer jogo com suporte a controle, ampliando o acesso e a usabilidade da solução.

2. Desenvolvimento do Sistema

O sistema VirtualGamePad é composto por quatro elementos que atuam em conjunto para transformar um dispositivo móvel em um controle de videogame funcional para computador.

2.1. Aplicativo Cliente (GamePad Mobile)

Desenvolvido usando React Native, o aplicativo funciona como um controle virtual. O aplicativo apresenta uma tela inicial para conexão com o computador, seguida por uma interface com botões e analógicos. Ao utilizar o controle os comandos são convertidos para uma estrutura em JSON contendo o ID do usuário e o comando em específico. Esses dados são enviados via WebSocket para o servidor.

2.2. Servidor (GamePad Receiver)

Desenvolvido em Python, o Servidor é responsável por gerenciar a conexão com o dispositivo móvel, receber e tratar os dados recebidos. Sendo capaz de aceitar até 8 conexões simultaneamente.

2.3. Emulador de Controle

O servidor utiliza a biblioteca vgamepad para transformar os dados recebidos em comandos de controle de videogame. Assim fazendo com que o sistema operacional

reconheça que foi adicionado um dispositivo ao sistema, como se fosse um controle físico real.

2.4. Comunicação Via WebSocket

A escolha do protocolo de comunicação WebSocket, deve-se à capacidade de estabelecer uma comunicação bidirecional e contínua entre o servidor e cliente por meio de uma única conexão TCP, reduzindo a latência e *overhead* de conexões repetidas como ocorre no HTTP tradicional. Essa arquitetura é ideal para sistemas em tempo real, como jogos, por evitar o uso de *polling* constante e garante maior fluidez na troca de dados entre o controle virtual e o computador.

2.5. Fluxo de Comunicação



Figura 1. Fluxo de comunicação

3. Testes de latência

A latência em sistemas de controle de videogame é um fator determinante para garantir a responsividade da interface e uma experiência satisfatória para o usuário. Para avaliar o desempenho do VirtualGamePad, foram realizados testes com foco na latência *end-to-end*, ou seja, o tempo entre a ação no dispositivo móvel e a resposta do computador.

3.1. Metodologia de Medição

A latência foi medida por meio de registro de timestamp no momento do envio (cliente) e do recebimento (servidor) da mensagem. O cálculo seguiu o modelo de RTT (*Round Trip Time*), considerando a diferença entre os tempos registrados. Para o teste foram realizadas 10.000 requisições no jogo Red Dead Redemption II.

Os testes foram realizados em um smartphone Android e um computador com Windows 11, com diferentes tipos de conexão: Wi-Fi 6, 5G, e 4G.

3.2. Valores da Latência

A tabela 1 apresenta os resultados obtidos nos testes de latência

Tabela 1. Valores mínimos, máximos e médios de latência e o número de requisições acima de 100ms.

Tipo de conexão	Latência Média (ms)	Latência Mínima (ms)	Variação Máxima (ms)	Requisições totais	Requisições acima de 100ms
5g	26.27 ms	3 ms	200 ms	10000	171
4g	30.39 ms	3 ms	172 ms	10000	107
Wi-Fi	14.20 ms	3ms	123 ms	10000	3

3.3. Análise Crítica

Os dados mostram que o sistema apresentou bom desempenho em redes Wi-Fi, com latência média abaixo do limite crítico de 33ms para jogos de precisão, conforme sugerido por Hoth (2022). Em conexão móvel (4G e 5G), a latência média ainda permite jogabilidade aceitável, mas o número de requisições com latência superior a 100ms aumenta – o que pode afetar a fluidez em jogos mais exigentes.

Embora o 5G tenha mostrado melhor média que o 4G, apresentou mais picos de latência alta, indicando variabilidade no desempenho. Tais flutuações podem ser tratadas em versões futuras com técnicas de compensação de latência, como *buffering* adaptativo ou interpolação de comandos.

4. Resultados Obtidos

Os resultados obtidos demonstram que o VirtualGamePad oferece uma experiência de controle responsiva em contextos com boa conectividade, especialmente em redes Wi-Fi, onde a latência média permaneceu abaixo dos 30 ms. Esse valor está dentro dos limites recomendados por estudos como o de Hoth (2022), que aponta prejuízo à jogabilidade em latências superiores a 33 ms para usuários mais experientes.

Comparado a soluções anteriores, como a de Vajk et al. (2008), que utilizavam celulares com sensores físicos e comunicação via Bluetooth para controlar jogos em grandes displays, a proposta apresentada neste trabalho representa um avanço significativo. Ao empregar smartphones modernos com interface por toque, conexão via WebSocket e compatibilidade com qualquer jogo que ofereça suporte a controles, o VirtualGamepad elimina a necessidade de hardware específicos. Enquanto a solução de Vajk exigia dispositivos, com sensores 3D e jogos especialmente adaptados para essa interação, o sistema atual amplia a acessibilidade e a aplicabilidade ao adotar tecnologias amplamente disponíveis e padronizadas.

4.1. Interfaces Obtidas



(A)



(B)

Figura 2. Interfaces desenvolvidas: Interface do Gamepad Receiver, gerenciando a Rede de conexões do computador. (A), Interface de GamePad Mobile com o gerenciador do computador. (B)

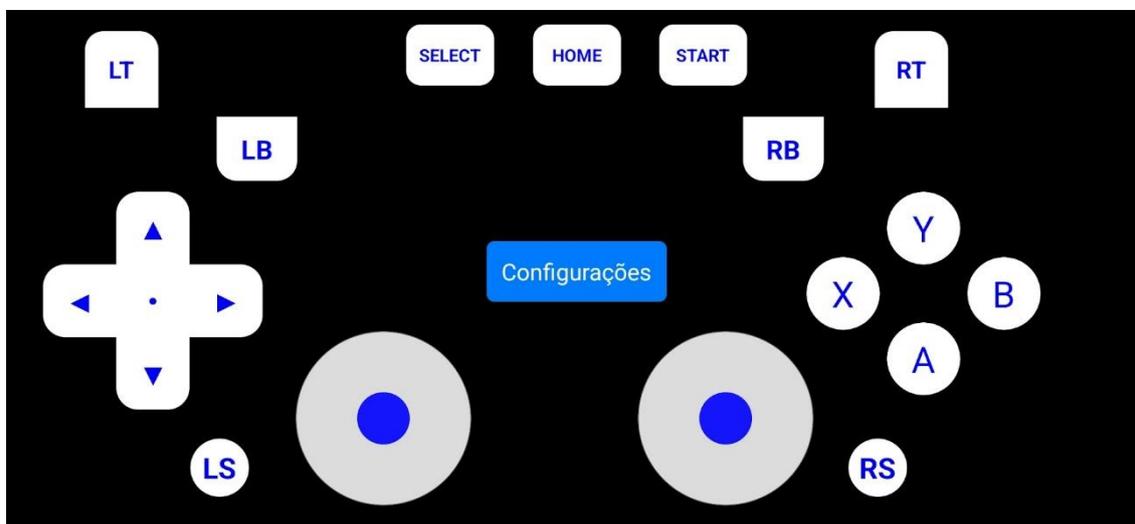


Figura 3. Interface de GamePad Mobile com o gerenciador do computador.

5. Conclusão

O desenvolvimento do VirtualGamePad demonstrou a viabilidade de um controle virtual usando WebSocket, permitindo que dispositivos móveis atuem como controles de videogame em computadores. A implementação utilizando React Native para o aplicativo cliente e Python para o servidor mostrou-se eficiente na comunicação em tempo real, garantindo uma experiência interativa fluida. Os testes de latência evidenciaram a influência do tipo de conexão na responsividade do sistema. A conexão Wi-Fi apresentou os melhores resultados, com latências médias abaixo do limite crítico para jogos de alta precisão (33 ms), enquanto redes 4G e 5G podem apresentar pequenas variações que impactam a jogabilidade.

O uso do protocolo WebSocket se mostrou uma solução robusta para esse tipo de aplicação, possibilitando comunicação bidirecional e de baixa latência. No entanto, melhorias podem ser exploradas em futuras pesquisas, como a otimização da transmissão de dados e a implementação de algoritmos de compensação de latência para cenários de rede menos favoráveis. Por fim, o VirtualGamePad se apresenta como uma solução acessível e de fácil implementação, permitindo que qualquer dispositivo móvel se transforme em um controle de videogame, ampliando as possibilidades de interação e democratizando o acesso a jogos eletrônicos.

Referências

- Hoth, T. (2022). Effects of induced latency on performance and perception in video games. In: Proceedings of the International Conference on Game Performance.
- Salen, K. e Zimmerman, E. (2004). Regras do jogo: fundamentos do design de jogos. São Paulo: Blucher.
- Vajk, T., Coulton, P., Bamford, W. and Edwards, R. (2008). Using a mobile phone as a “Wii-like” controller for playing games on a large public display. International Journal of Computer Games Technology, 2008(1), Article ID 539078, 7 pages.
- Amirkhanov, B., Amirkhanova, G., Kunelbayev, M., Adilzhanova, S. and Tokhtassyn, M. (2025). Evaluating HTTP, MQTT over TCP and MQTT over WebSocket for digital twin applications: A comparative analysis on latency, stability, and integration. International Journal of Innovative Research and Scientific Studies, 8(1), 679–694.
- Babovic, Z. B., Protic, J. and Milutinovic, V. (2016). Web performance evaluation for internet of things applications. IEEE Access, 4, 6974–6992.
- Ogundeyi, K. E. and Yinka-Banjo, C. (2019). WebSocket in real time application. Nigerian Journal of Technology, 38(4), 1010–1020.
- React Native. (2025). React Native: Official Documentation. Disponível em: <https://reactnative.dev/>. Acesso em: 27 fev. 2025.
- Python Software Foundation. (2025). vgamepad: Virtual gamepad emulation. Disponível em: <https://pypi.org/project/vgamepad/>. Acesso em: 27 mar. 2025.
- Python Software Foundation. (2025). Python Programming Language. Disponível em: <https://www.python.org/>. Acesso em: 27 mar. 2025.
- Internet Engineering Task Force. (2011). RFC 6455: The WebSocket Protocol. Disponível em: <https://datatracker.ietf.org/doc/html/rfc6455>. Acesso em 27 de mar. 2025.