

Comunicação cliente-servidor bilateral de baixa latência aplicado a Android

Guilherme Freire Silva, Marco Dimas Gubitoso (Orientador)

IME-USP - Instituto de Matemática e Estatística da Universidade de São Paulo, Departamento de Ciência da Computação



Introdução

- ▶ Dispositivos móveis, como smartphones e tables, são uma tecnologia muito nova e em constante desenvolvimento. Possuem bastante espaço para inovação, além de estarem fortemente presentes no dia a dia. Um aspecto pouco explorado deles é seu uso como parte de um sistema interativo Web, ou seja, um sistema no qual eles enviam e recebem informações de um servidor, para os mais variados fins.
- ▶ Essa arquitetura permite, por exemplo, executar um jogo em uma página Web que utiliza o smartphone como um controle, ou manter um banco de dados no servidor e fazer buscas utilizando o smartphone.

Objetivos

1. Desenvolver um aplicativo para smartphones.
2. Criar um servidor que se comunique com ele.
3. Desenvolver uma aplicação Web que receba e utilize dados vindos do smartphone.
4. Encontrar uma forma eficiente de comunicação entre eles.

Aplicativo Mobile

- ▶ Criado com o Framework Cordova.
 - ▷ Multiplataforma, incluindo versão Web.
- ▶ Conecta-se com o servidor através de um IP:Porta.
 - ▷ Recebe e envia dados utilizando Sockets.
- ▶ Sobre sua interface e utilização.
 - ▷ O formulário no início faz a conexão com o servidor, fornecendo o endereço de IP e a Porta.
 - ▷ Os botões azuis enviam mensagens pontuais ao servidor.
 - ▷ Os botões cinzas ligam ou desligam um fluxo de mensagens com dados do acelerômetro para o servidor.
 - ▷ O Log exibe mensagens vindas do servidor e outros feedbacks.

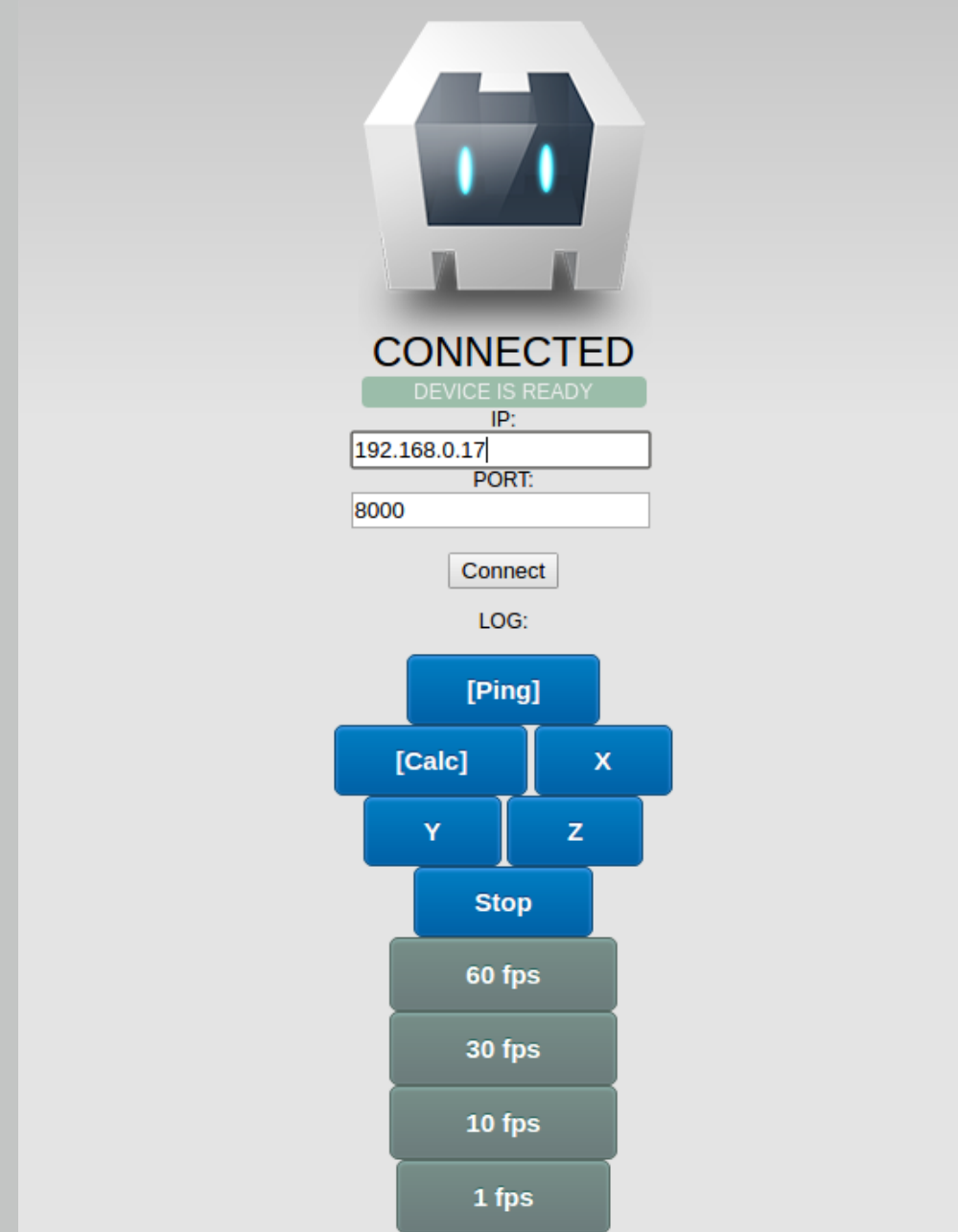


Figura 1: Aplicativo Cordova

Comunicação utilizando Socket.io

- ▶ Implementa o protocolo WebSocket.
 - ▷ WebSocket lida com os sockets de uma conexão. Ele permite a comunicação bilateral por meio de uma interface simples.
- ▶ Características do Socket.io:
 - ▷ Orientado a eventos. Cada mensagem que chega é recebida como um evento. Se o processo escuta esse evento, ele executará uma função com os parâmetros recebidos.
 - ▷ Definição arbitrária de eventos. O código define quais eventos escuta e qual o evento que é enviado.
 - ▷ Assíncrono. Uma vez enviada a mensagem, o processo continua sua execução, sem precisar esperar um evento de resposta do outro processo.
 - ▷ Conexão bilateral. Pode enviar e receber mensagens livremente.
 - ▷ Baixa latência. O protocolo facilita ao máximo o envio e recebimento de mensagens.
 - ▷ Baixo *overhead*. Possui um cabeçalho mínimo. Diferente de outros protocolos, ele possui somente o tipo de evento e o tamanho da mensagem enviada.
 - ▷ Permite múltiplas conexões simultâneas. Através de multiplexação, permite que vários Sockets estejam conectados ao servidor através de uma única porta.
 - ▷ Se o servidor não permitir o uso de WebSockets, recorre a outras técnicas menos eficientes de comunicação bilateral, com a mesma interface.

Servidor

- ▶ Escrito em JavaScript com ajuda de Node.js e ExpressJS.
- ▶ Fornece a página HTML que executa o código da aplicação.
- ▶ Com a utilização de Socket.io, aceita a conexão com o smartphone.
 - ▷ Permite a conexão com múltiplos dispositivos simultaneamente.
- ▶ Recebe eventos do smartphone e o responde ou repassa esses dados à página Web.

Exemplo simples de Aplicação Web - Esfera Interativa

- ▶ Utiliza a biblioteca 3D para JavaScript THREE.js.
 - ▷ Utiliza a GPU do usuário para renderizar o cenário 3D.
- ▶ Mantém um socket aberto para a conexão com o servidor.
- ▶ Inicia-se estática, mas recebe eventos para aumentar e reduzir sua velocidade nos eixos X, Y e Z.
- ▶ Pode receber um outro evento, que chega a intervalos constantes e define sua velocidade de acordo com o acelerômetro do smartphone.
 - ▷ Esse evento explicita a baixa latência alcançada com o uso de Socket.io para a comunicação, pois é um feedback quase instantâneo à movimentação do dispositivo pelo usuário.

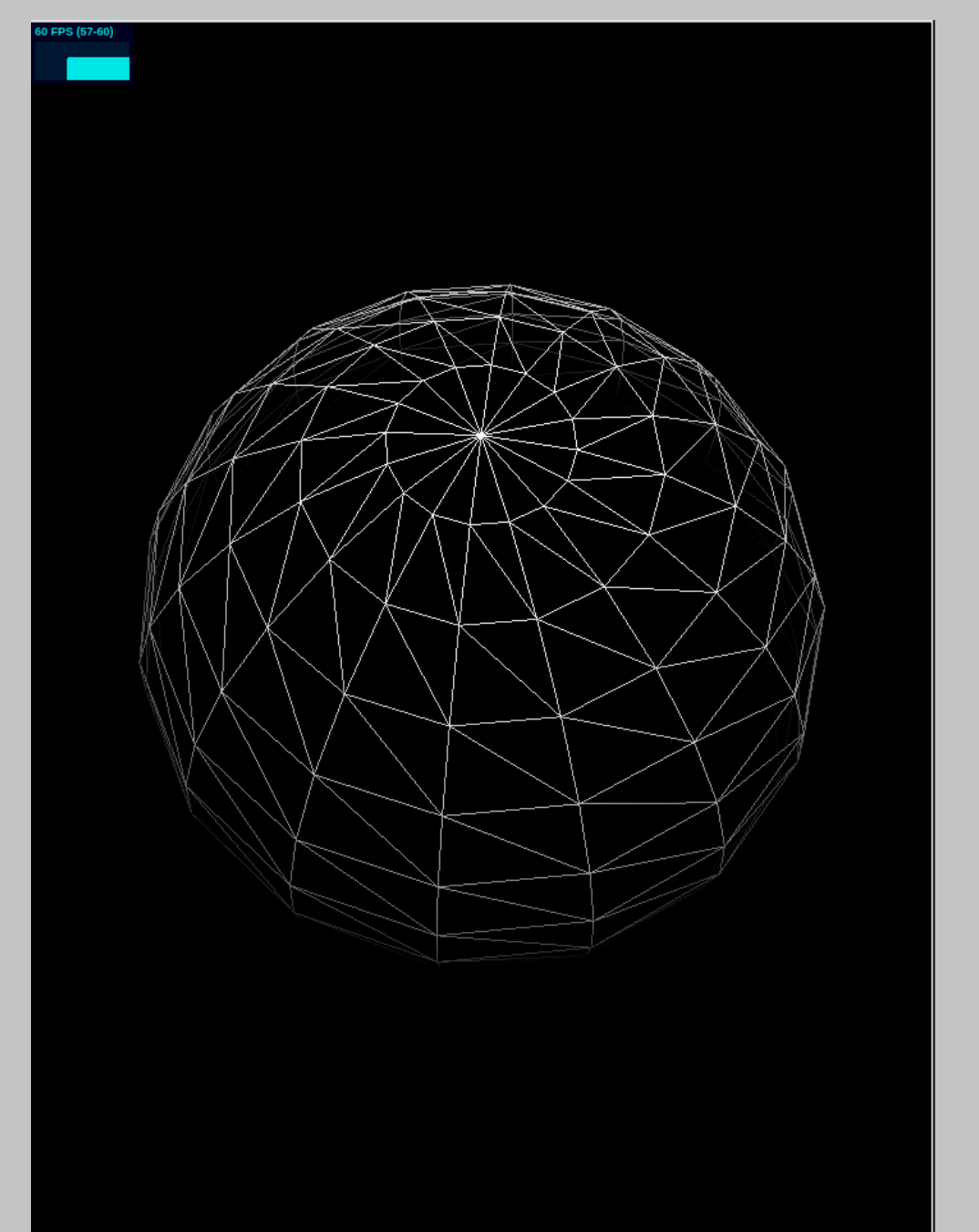


Figura 2: Esfera interativa

Conclusão

- ▶ O projeto proposto se mostrou plenamente possível e viável, com a utilização das ferramentas certas.
- ▶ As aplicações desenvolvidas e apresentadas possuem os elementos básicos para a criação de sistemas muito maiores e mais robustos.
- ▶ Em resumo, foi criado um arcabouço para a comunicação bilateral entre múltiplos clientes e um servidor, esses clientes podendo ser executados em smartphones, tirando proveito de suas funcionalidades exclusivas.

Referências

- [1] Socket.io, <http://socket.io/>
- [2] WebSocket, <http://websocket.org/>
- [3] Apache Cordova, <https://cordova.apache.org/>
- [4] Node.JS, <https://nodejs.org/en/>
- [5] THREE.js, <https://threejs.org/>



IME-USP