Documentação *Synesthesia Vision*
Manual do *Software*

Sumário

Introdução

Este documento foi criado com o intuito de fornecer as informações obtidas durante todo o desenvolvimento do projeto para auxiliar futuros “colaboradores” do código-fonte.
Neste arquivo estão especificadas as funcionalidade de cada módulo implementado e as suas respectivas explicações com o propósito de fornecer um guia de construção de novas funcionalidades.

Instalação

➢ Requisitos

- Ionic;
- Node.js;
- Android Studio;
- Gradle
- JDK

➢ Tutorial

I. Download das ferramentas:
   a) Android Studio: https://developer.android.com/studio
   b) Node.js: https://nodejs.org/pt-br/download/
   c) Gradle: https://gradle.org/releases/
   d) JDK:
https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

II. Instalação das ferramentas baixadas no passo anterior
III. Adicionar o npm ao *path* do sistema:
   a) Pressionar as teclas Win + e
   b) Botão direito do mouse em “Este computador” e clica em propriedades
   c) Selecionar a opção “Configurações avançadas do sistema” no menu lateral
   d) Selecionar a opção “Variáveis de ambiente”
   e) No campo “Variáveis do sistema” selecionar a *path*
   f) Para obter o caminho do npm, pressionar as teclar Win + r, e digitar “%AppData%”
   g) Entrar na pasta npm e copiar o caminho
   h) Voltando ao path, clicar no botão novo e colar o caminho do npm
IV. Adicionar o jdk ao path do sistema
   a) Pressionar as teclas Win + e

b) Acessar a pasta cujo JDK foi instalada
c) Acessar a pasta “bin”, localizada dentro da pasta do JDK
d) Copiar o caminho
e) Botão direito do mouse em “Este computador” e clica em propriedades
f) Selecionar a opção “Configurações avançadas do sistema” no menu lateral
g) Selecionar a opção “Variáveis de ambiente”
h) No campo “Variáveis do sistema” selecionar a *path*
i) Clicar no botão “novo” e colar o caminho do jdk

V. Adicionar o Gradle ao path do sistema
a) Extrair o zip em uma pasta da sua escolha
b) Acessar a pasta “bin”, localizada dentro da pasta do Gradle
c) Copiar o caminho
d) Botão direito do mouse em “Este computador” e clica em propriedades
e) Selecionar a opção “Configurações avançadas do sistema” no menu lateral
f) Selecionar a opção “Variáveis de ambiente”
g) No campo “Variáveis do sistema” selecionar a *path*
h) Clicar no botão “novo” e colar o caminho do Gradle

VI. Instalar o Ionic
a) Clicar no menu iniciar
b) Digitar “cmd”, e clique com o botão direito
c) Selecione a opção “Executar como administrador”
d) Digite o seguinte comando: npm install -g ionic
e) E tecle “enter” para executar

Diretórios

- App
- Pages
- Pipes/date
- Providers

Pages

As páginas são responsáveis por interagir diretamente com o usuário.
São compostas por três arquivos:

- ✓ .html: Responsável pela interface com o usuário.
- ✓ .scss: Responsável por estilizar o conteúdo html.
- ✓ .ts: Responsável pela funcionalidade do sistema.

Bluetooth Connection Verify Page

Métodos:

- **ionViewDidLoad(**): Executa o som inicial.

- **ionViewWillEnter():** Verifica se o dispositivo está conectado com algum dispositivo bluetooth sempre que a página é carregada. Caso esteja, ele irá desconectar.

- **startScanning():** Verifica dispositivos pareados e não pareados. Se um dispositivo pareado tiver o nome “Synesthesia”, o método procurará um endereço no localStorage.

- **selectDevice():** Exibe um alerta perguntando se o usuário deseja se conectar a um endereço que foi selecionado pelo usuário. Redireciona para a tela principal se o usuário desejar se conectar.
  - @param address
  - @param device

- **disconnect():** Desconecta do bluetooth.

- **synesthesia():** Irá carregar a página Synesthesia Vision Page.

- **showAlert():** Cria um alert informando um erro, com a mensagem de erro do parâmetro.
    - @param message

- **createLoading():** Cria o spinner de carregamento.

- **saveAddress(device):** Salva o endereço do dispositivo pressionado caso haja uma conexão bem sucedida.
    - @param device

- **checkAddress():** Verifica se o local storage tem o endereço do dispositivo do synesthesia.

- **autoConnect(address: string):** Se conecta a um endereço bluetooth já conhecido e redireciona para a página principal.
    - @param address

- **checkEnabledBluetooth():** Verifica se o bluetooth está ativado. Se não estiver, irá ativá-lo.

Horários Page

Métodos:

- **ionViewDidLoad():** Requisita a primeira busca de paradas, caso não tenha sido ainda executada.

- **speakData():** Realiza o tratamento de fala do tempo restante do ônibus selecionado.

- **getLinhas():** Realiza a busca pelas paradas próximas.

- **getParadas():** Exibe a lista de paradas, sendo ordenada pela distância, da mais perto para a mais longínqua.
  - @param recarregando

- **getHorarios():** Exibe a lista de ônibus da parada selecionada.
  - @param labelParada

- **transformTime():** Converte o valor da hora recebido em hora e minutos.
- @param value

- **openModalRenomear():** Exibe o quadro de renomeio de parada.
  - @param labelParada
  - @param nombreParada

Modals

Métodos:

- setRenomear(): Define o nome digitado como um apelido para a parada selecionada.

- dismiss(): Fecha o quadro de renomeio de parada.

Áudio Provider 3

Métodos:

- **setUpAudio():** Realiza a conversão para um arquivo compatível.
  - @param bufferedContent

- **playAudioTrack():** Cria um som estéreo e o carrega no aplicativo.
  - @param track

- **playSound():** Inicializa todos os serviços de sonorização, realizando as devidas checagens.

- **zeraDistancia():** Zera os valores presentes no sensor.

- **stopRunningSound():** Finaliza a repetição de execução do som.

Synesthesia Vision Page

Métodos:

- **ionViewDidLoad():** Realiza a checagem de permissões.
- **ionViewWillLeave():** Realiza a desconexão do módulo bluetooth.
- **ionViewWillEnter():** Obtém os dados do bluetooth
- **toggleStatusButton():** Realiza a troca do texto do botão ao inicializar a sonorização do aplicativo.
- **playSound():** Inicializa a vibração o telefone e a sonorização.
- **stopSound():** Encerra a vibração do telefone e a sonorização.
- **checkWeather():** Realiza a checagem de clima.
- **increaseFrequency():** Aumenta a frequência de beeps sonorizados.
- **decreaseFrequency():** Diminui a frequência de beeps sonorizados.
- **getBluetoothData():** Realiza o processamento dos dados recebidos pelos botões.
- **getFunction():** Chama a função de acordo com o parâmetro recebido.
  - @param bluetoothData
- **getParadaProxima():** Exibe uma lista com as paradas próximas.
- **speakLuminosity():** Realiza o tratamento e informa se o ambiente está claro ou escuro.
  - @param dataBuffer
- **voltarConexaoBluetooth():** Retorna à página de conexão.

Bluetooth LE Connection Verify Page

Métodos:

- **ionViewDidEnter():** executa o método de escaneamento.

- **scan():** escaneia os dispositivos bluetooth low energy e adiciona em um array de dispositivos.

- **onDeviceDiscovered()**: apresenta os dispositivos que foram encontrados.
  - @param device

- **scanError():** Caso ocorra algum erro durante o escaneamento, é exibido um aviso.
  - @param error

- **setStatus():** exibe todas as atualizações que ocorrem durante o escaneamento.
  - @param message

- **deviceSelected():** ao selecionar um dispositivo, o usuário é redirecionado para a próxima página, passando por parâmetro o dispositivo selecionado.
  - @param device

- **checkEnabledBluetooth():** realiza a verificação a fim de descobrir se o smartphone está com o bluetooth habilitado.

Choose Device Page

Métodos:

- **oculosAntigo():** Realiza a navegação para a página de verificação bluetooth da versão 1.0 dos óculos sensoriais.
- **oculosNovo():** Realiza a navegação para a página de verificação bluetooth da versão 2.0 dos óculos sensoriais.

Game Page

Métodos:

- **ionViewDidEnter():** Executa o método de escaneamento.

- **scan():** Escaneia os dispositivos bluetooth low energy e adiciona em um array de dispositivos.

- **onDeviceDiscovered()**: Apresenta os dispositivos que foram encontrados.
  - @param device

- **scanError():** Caso ocorra algum erro durante o escaneamento, é exibido um aviso.
  - @param error

- **setStatus():** Exibe todas as atualizações que ocorrem durante o escaneamento.
  - @param message

- **deviceSelected():** Ao selecionar um dispositivo, o usuário é redirecionado para a próxima página, passando por parâmetro o dispositivo selecionado.
  - @param device

- **checkEnabledBluetooth():** Realiza a verificação a fim de descobrir se o smartphone está com o bluetooth habilitado.

Game Pontuacao Page

Métodos:

- **onConnected():** Ao ter um dispositivo conectado, é iniciada a verificação de notificações do mesmo.
  - @param peripheral

- **onDataChange():** Verifica se o bluetooth recebeu o valor que simboliza que bateu em um obstáculo.
  - @param buffer

- **ionViewWillLeave():** Disconecta do dispositivo atualmente conectado.

- **startTimer():** Incrementa o tempo decorrido.

- **pontuacaoBateu():** Diminui os pontos ao bater em um obstáculo.

- **showAlert():** Caso ocorra algum erro, uma mensagem de alerta é exibida.
  - @param title
  - @param message

- **stopGame():** Para o jogo.

Rotas Page

Métodos:

- **ionViewDidLoad():** Obtém a localização atual do usuário ao entrar na página.

- **loadmap():** Carrega o mapa baseado na localização atual do usuário e na distância para a parada selecionada.

- **getInitUserLocation():** Realiza a requisição para obter a localização atual do dispositivo.

- **updateUserLocation():** Atualiza a localização atual do dispositivo.

Synesthesia Vision LE Page

Métodos:

- **onConnected():** Inicia a verificação por alterações no valor recebido pelo bluetooth.
  - @param peripheral
- **onDataChange():** Chama a função baseado no valor recebido pelo bluetooth.
  - @param buffer
- **ionViewDidLoad():** Solicita a permissão de localização do usuário.
- **ionViewWillLeave():** Desconecta o dispositivo dos óculos sensoriais.
- **checkWeather():** Inicia a verificação do tempo.
- **getFunction():** Exibe as opções de checagem de tempo, luminosidade ou de paradas próximas.
  - @param bluetoothData
- **getParadaProxima():** Redireciona para a página de paradas.
- **speakLuminosity():** Indica a luminosidade do local em que o usuário está.
  - @param dataBuffer
- **getLuminosidade():** Recebe o valor do LDR.
  - @param bluetoothData
- **voltarConexaoBluetooth():** Volta para a tela de verificação de dispositivos pelo bluetooth.
- **startGame():** Redireciona para a página do jogo.
  - @param peripheral
- **showAlert():** Exibe um alerta na tela.
  - @param title
  - @param message

- **setStatus():** Informa o estado atual da comunicação do dispostivo com os óculos.
  - @param message

Providers

Responsável por fazer conexão com as API’s, gerenciar armazenamento, autenticação e etc.

Áudio Provider:

Função: Responsável por faer a geração de som.
Importações:

- Web Audio API: Processa e sintetiza áudio em aplicativos Web.
Referências: https://www.w3.org/TR/webaudio/

Bluetooth Provider:

Função: Este provider habilita a comunicação serial via Bluetooth. É utiliado para estabelecer conexões entre Android ou iOS e um Arduino, receber e tratar os dados que os sensores do óculos enviam para o dispositivo através do bluetooth.

Importações:
- Native Storage: Armazena os dados no Local Storage do dispositivo.
Referências: https://ionicframework.com/docs/native/native-storage/

Bluetooth Serial: habilita a comunicação serial via Bluetooth.
Referências: https://ionicframework.com/docs/native/bluetooth-serial/

Bus Integration Provider

Função: Responsável por realizar a busca por paradas que estão localizadas próximas ao usuário.

Importações:

- Http: Cordova / Phonegap plugin para comunicação com servidores HTTP.

Referências: https://ionicframework.com/docs/native/http/

- Geolocation: Fornece informações sobre a localização do dispositivo, como latitude e longitude.

Referências: https://ionicframework.com/docs/native/geolocation/

- Alert Controller: Faz o controle e personalização de alerts, ou seja, diálogos onde são apresentados aos usuários informações ou coleta informações do usuário usando inputs.

Referências: https://ionicframework.com/docs/api/components/alert/AlertController/

- Loading Controller: Faz o controle e personalização de um overlay ou loading spinner que pode ser usada para indicar atividade enquanto bloqueia a interação do usuário.

Referências: https://ionicframework.com/docs/api/components/loading/LoadingController/

Permission Provider

Função: Realiza a verificação de permissões do Android.

Importações:

- Android Permissions: Este plugin é projetado para suportar o novo mecanismo de verificação de permissões do Android.

Referências: https://ionicframework.com/docs/native/android-permissions/

Text To Speech Provider

Função: Responsável por transmitir informações em áudio para o usuário.

Importações:

- Text To Speech: Plugin Text to Speech.

Referências: https://ionicframework.com/docs/native/text-to-speech/

Weather Forecast

Função: Verifica e informa ao usuário as condições climáticas atuais.

Importações:

- HTTP: Responsável por fazer a comunicação com servidores http.

Referências: https://ionicframework.com/docs/native/http/

- Alert Controller: Faz o controle e personalização de alerts, ou seja, diálogos onde são apresentados aos usuários informações ou coleta informações do usuário usando inputs.

Referências: https://ionicframework.com/docs/api/components/alert/AlertController/

- Text To Speech Provider: Responsável por transmitir informações em áudio para o usuário.

- Geolocation: Fornece informações sobre a localização do dispositivo, como latitude e longitude.

Referências: https://ionicframework.com/docs/native/geolocation/

- Location Accuracy: Solicita a ativação/alteração dos Serviços de Localização ao acionar um diálogo nativo (alert) a partir do aplicativo, evitando a necessidade de o usuário deixar seu aplicativo para alterar as configurações de local manualmente.

Referências: https://ionicframework.com/docs/native/location-accuracy/

- Mobile Accessibility: Expõe o status de vários recursos de acessibilidade de sistemas operacionais móveis. Usado também para que o aplicativo envie uma string para ser falada pelo leitor de tela ou um comando para impedir que o leitor de tela fale.

Referências: https://ionicframework.com/docs/native/mobile-accessibility/