Universita' del Piemonte Amedeo Avogado

Corso di Laurea in Informatica

MagicMirror

Relatore
Prof. Marco Guazzone

Relazione della prova finale di:
Riccardo Berto
Matricola: 20003275

Anno Accademico 2016/2017

Indice

Abstract

Nella prima parte dello stage sono stati sviluppati dei moduli, in diversi linguaggi di programmazione, che permettessero l'interazione tra uomo-macchina da parte di un dispositivo.

Tra i moduli sviluppati vi sono il riconoscimento di presenza umana o di uno specifico volto (tramite l'utilizzo di una telecamera) e il riconoscimento vocale, ovvero l'impartizione di comandi specifici per mezzo di un microfono.

Nella seconda parte dello stage, invece, é stata sviluppata un'interfaccia web, la quale ha il compito di modificare la configurazione dei moduli (anche quelli creati da terzi) senza dover andare a modificare il file in locale rendendo pi facile aggiornare le features del dispositivo.

Capitolo 1

Scopi e Problemi

1.1 Moduli del MagicMirror

Nello sviluppo dei moduli del MagicMirror si é dovuto affrontare il problema di far comunicare i diversi dispositivi hardware (tra cui microfono e telecamera) con il calcolatore principale, utilizzando tecnologie, software e linguaggi diversi tra loro, scelti in base alle loro caratteristiche e ai loro punti di forza. Lo scopo dei moduli é di permettere l'interazione tra il software principale e l'essere umano, e di poter controllare le funzioni della macchina senza dover utilizzare i tradizionali metodi di input (mouse e tastiera).

1.2 Interfaccia di controllo del MagicMirror

Nello sviluppo della pagina Web di controllo si é affrontato il problema di dover modificare il documento di configurazione del software principale da una pagina web senza eliminarne parti essenziali o modificarlo in un formato sbagliato (gestito tramite validatore). Lo scambio dei messaggi tra il Backend ed il Frontend avviene in formato JSON, lo stesso formato con cui é stipulato il file di configurazione del Magic Mirror.
Lo scopo dell'interfaccia di permettere ad un qualsiasi utente (con i permessi) di gestire i diversi moduli implementati nel software modificandone il JSON, senza accedere fisicamente alla macchina. La pagina, inoltre, permette di configurare moduli creati anche da terzi: inserendoli semplicemente nella directory del dispositivo il Backend li cerca e li indicizza nell'interfaccia, da dove possono essere attivati.

Capitolo 2

Tecnologie Implicate

Lo sviluppo del progetto é stato svolto nell'ambiente Raspbian[1] una distribuzione Debian[2] che gira sul dispositivo Rasperry[3]. Inoltre sono state adottate diverse tecnologie, recenti e non, nel campo della creazione di un'applicazione e i relativi moduli(Electron [6], OpenCV [4], GoogleSpeechRecognition [5]) e di web server (NodeJS [14], Model-View-Controller Architecture [7], Express [8], MySQL [9], Mustache [10]), diversi linguaggi di programmazione (JavaScript [11], Python [12]) e tecnologie per visionare e condividere codici (GitLab [13]).

2.1 Raspberry

RaspberryPi un calcolatore elettronico, montato su una singola scheda elettornica, a basso costo dal consumo ridotto e alta portabilitá. Rilasciato per la prima volta intorno al 2012 é diventato un prodotto utilizzato per una moltitudine di progetti sia aziendali che casalinghi. Il modello usato nello stage RaspberryPi 3 model B e monta:

- una porta HDMI
- porta LAN
- uscita Aux
- 4 porte USB
- 40 General Purpose Input/Output(GPIO)
- Scheda di rete wirless
- Alimentazione microUSB 5V

- un bus camera serial interface(CSI)
- ingesso per microSD

Il sistema operativo per Raspberry deve essere installato su una miscroSD opportunamente formattata e configurata con il corretto Master Boot Record (MBR).

2.2 Raspbian

Raspbian é un sistema operativo multi-architettura della distribuzione Debian, completamente libero, ottimizzato per Raspberry. Fú sviluppato da Mike Thompson e Peter Green come progetto non affiliato alla compagnia Raspberry Pi Fundation, pensato apposta per le basse prestazioni dei processori Advanced RISC Machine(ARM) montati sul dispostivo. La prima versione venne rilasciata nel 2012.

2.3 Electron

Electron un Framework open source rilasciato per la prima volta nel 2013, ma la prima versione stabile uscita di recente. É disponibile sui sistemi operativi Window, MacOS e Linux ed scritto in C++ e Javascript. Il framework permette la creazione di interfacce grafiche (GUI) per applicazioni cross platform, utilizzando teconologie gi esistenti per lo sviluppo del backend e del frontend (Javascript, NodeJS, V8 [15]). All'avvio Electron inizializza una pagina in Chromium, che renderizza una pagina web, e un server NodeJs che si occupa di trasmettere e interagire con il frontend. Un'applicazione Electorn ha bisongo di 3 componenti principali:

- Il package.json, un file JSON, che deve contenere almeno il nome dell'applicazione, la versione dell'applicazione creata, la descrizione di quest'ultima e il nome del file principale dell'applicazione (necessaria per l'avvio)

```
{
  "name": "magicmirror",
  "version": "2.1.1",
  "description": "The open source smart platform",
  "main": "js/electron.js"
}
```

- Un file html che contiene il template della pagina mostrata dall'applicazione
- Un file JavaScript che contiene il codice di esecuzione dell'applicazione: creazione della finestra, rendering della pagina, ecc...

2.4 OpenCV

OpenCV (Open Source Computer Vision Library) una libreria software sviluppata intorno al 2000 utilizzata nell'abito della visione in tempo reale da parte di una macchia per mezzo di input digitali, ottenuti tramite telecamera o fotocamera ad esempio.
La librearia supportata per i linguaggi C++(linguaggio in cui é scritta e dunque di cui ha l'interfaccia primaria), C, Python e Java e per diversi sistemi operativi, anche mobile.
OpenCV prede in input un'immagine o uno stream (come un video o una serie di immagini) e, utilizzando algoritmi di visioning implementati al suo interno, riconosce oggetti o specifiche forme, inoltre se ne pu aumentare l'efficienza applicando algoritmi di Machine Learning per individuare e riconoscere oggetti specifici.

2.5 GoogleSpeechRecognition

Negli ultimi anni Google ha ampliato sempre di pi il suo catalogo per quanto rigarda i servizi cloud e le web API. Google Speech To Text (o Google Speech Recognition) un servizio che, ricevendo in input un file audio o uno stream, ottenuto per mezzo di un dispositivo di audio input, traduce il parlato in testo scritto.
L'API supporta oltre 110 lingue, e le librerie sono disponibili nei linguaggi C#, GO, Java, Node.JS, PhP, Python e Ruby, inoltre dispone di varianti:

- Una con interfaccia REpresentational State Transfer(REST), che comunica per mezzo di URI
- Una con gRPC, un sistema di chiamata di procedura remota

Bibliografia

[1] Raspbian wikipedia, https://it.wikipedia.org/wiki/Raspbian

[2] Debian wikidia, https://it.wikipedia.org/wiki/Debian

[3] Raspberry official website, https://www.raspberrypi.org/

[4] OpenCV official website, http://opencv.org/

[5] Google Speech to Text API documentation, https://cloud.google.com/speech/

[6] Electron official website, https://electron.atom.io/

[7] MVC Wikipedia, https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller

[8] Express official website, http://expressjs.com/it/

[9] MySQL official website, https://www.mysql.com/it/

[10] Mustache Git Site, https://mustache.github.io/

[11] JavaScript, https://www.javascript.com/

[12] Python website, https://www.python.it/

[13] GitLab website, https://about.gitlab.com/

[14] Nodejs website, https://nodejs.org/it/

[15] V8 wikipedia, https://it.wikipedia.org/wiki/V8_(motore_JavaScript)