

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

Abstract

L'obiettivo dello stage era lo sviluppo di moduli in diversi linguaggi che permettesse l'interazione tra uomo-macchina da parte di un dispositivo, e di un'interfaccia che permettesse di configurare quest'ultimo da una pagina web.

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 tramite voce per mezzo di un microfono.

Nello sviluppo di questi moduli si sono incontrate moderne tipologie di software come OpenCV e Google SpeechToText che sono state implementate con diverse tipologie di linguaggi.

La pagina web sviluppata aveva il compito di modificare la configurazione dei moduli (anche quelli creati da terzi) senza dover andare a modificare il file in locale rendendo più facile aggiornare le features del dispositivo.

Contents

Abstract	I
Introduzione	III
1 Scopi e Problemi	IV
1.1 Moduli del MagicMirror	IV
1.2 Interfaccia di controllo del MagicMirror	IV
2 Tecnologie Implicate	VI
2.1 Electron	VI

Introduzione

Durante

prova

prova

prova

prova

Chapter 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 e software diversi tra loro e linguaggi differenti, dal momento che ognuno era risultato pi facile da implementare con uno specifico linguaggio rispetto ad un altro.

Lo scopo dei moduli é di permettere un'interazione umana tra il software principale e l'essere umano, e di poter controllare la macchina senza bisogno di usare input meccanici come 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.

L'ambiente in cui é stato svolto lo stage é Raspian una distribuzione Debian che gira sul calcolatore Rasperry.

Chapter 2

Tecnologie Implicate

Durante lo sviluppo del progetto sono state adottate diverse tecnologie recenti e non nel campo della creazione di moduli per un'applicazione (Electron [3], OpenCV [1], GoogleSpeechRecognition [2]) e di web server (NodeJS [11], Model-View-Controller Architecture [4], Express [5], MySQL [6], Mustache [7]), utilizzando diversi linguaggi di programmazione (JavaScript [8], Python [9]) e tecnologie per visionare e condividere codici (GitLab [10]).

2.1 Electron

Electron è un Framework open source rilasciato per la prima volta nel 2013, ma la prima versione stabile è uscita di recente. È disponibile sui sistemi operativi Window, MacOS e Linux ed è scritto in C++ e Javascript. Il framework permette la creazione di interfacce grafiche (GUI) per applicazioni cross platform, utilizzando tecnologie già esistenti per lo sviluppo del backend e del frontend (Javascript, NodeJS, V8 [12]). Un'applicazione Electron ha bisogno 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 modular smart mirror platform.",
  "main": "js/electron.js"
}
```

- fdfsaf

- la flessibilità e l'ef

Bibliography

- [1] OpenCV official website, <http://opencv.org/>
- [2] Google Speech to Text API documentation,
<https://cloud.google.com/speech/>
- [3] Electron official website, <https://electron.atom.io/>
- [4] MVC Wikipedia, <https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>
- [5] Express official website, <http://expressjs.com/it/>
- [6] MySQL official website, <https://www.mysql.com/it/>
- [7] Mustache Git Site, <https://mustache.github.io/>
- [8] JavaScript, <https://www.javascript.com/>
- [9] Python website, <https://www.python.it/>
- [10] GitLab website, <https://about.gitlab.com/>
- [11] Nodejs website, <https://nodejs.org/it/>
- [12] V8 wikipedia,
[https://it.wikipedia.org/wiki/V8_\(motore_JavaScript\)](https://it.wikipedia.org/wiki/V8_(motore_JavaScript))