

SMART MIRROR
Made by ASHWANI KUMAR

THINGS USED IN THIS PROJECT

Hardware components:

1 . Raspberry Pi 1 Model B+	1	
2 . Two-way mirror		1
3 . Camera (generic)	1	

(I'm using the NoIR but any camera should do)

4 . LED monitor	1	
-----------------	---	--

(Preferably that cover the whole surface of your two way mirror)

Software apps and online services:

OpenCV
Python
NodeJS

STORY :

Motion controlled SmartMirror showing time and temperature in background with the possibility of adding any widget to the menu. The camera is behind the two-way mirror to get the best possible look. The screen and camera are managed by a Raspberry Pi. The motion is managed by a server running in python with OpenCV, the server web is in NodeJS.

Preview:

Showing the display of the main menu and a use case on the widget Cinema (movie times of my local theater)

Capacities:

Recognition of gestures: palm, thumbs up/down, slide up/down/right/left
Time and outside temperature

Widgets:

Photo: take a photo and upload it to Dropbox
Map: display local Google Map
Cinema: movie time of local theater
DoodleJump: play the game (hard with the latency)
News: show international news

Debugging:

The motion server can't recognize gestures on a new environment : lights, hand colors... affect the process, that's why by launching the test.py file and tweak the HSV min/max values and others configs properly you can set it up for your home.

Dependencies:

Software:

OpenCV 2.x
Python 3.x
NodeJS

Hardware:

1. Two Way Mirror
 2. Raspberry pi
 3. Camera (I'm using the NoIR but any camera should do)
 4. LED Monitor (Preferably that cover the whole surface of your two way mirror)
- Building

Web server:

```
npm install
node server.js
go to http://localhost:3000
```

Motion server:

Install OpenCV 2.x and cv2 wrapper for python (should be in the opencv package)

```
pip install numpy
pip install tornado
python test.py for debug infos or python main.py for silent process
```

Notes:

The motion server was made in python to learn the language, but it should have been made in C/C++ to gain execution speed and flowness in the gestures recognitions. Also, python environment kinda sucks at the moment, the difficulty to set it up for python3 + opencv2 + windows and then linux was exhausting, I don't recommend.

Made by :

ASHWANI KUMAR
ashwanidv100@gmail.com