
puffyCV

Mar 19, 2021

Contents:

1	About the project	3
2	Installation	5
3	Calibration	7
4	Run and Deploy	9
5	Building Example	11
6	TODOs	13
7	Changelog	15

This is the documentation of the project puffyCV. It will provide camera recognition of steel darts thrown to a dart board. It is designed to work together with the , my scoreboard project to track darts gaming score.

CHAPTER 1

About the project

Playing steel darts and counting score by hand or by a mobile app was kind of a bummer for me. So first I designed a scoreboard app to track dart games. But at that point I was just using them like any other mobile app to keep track of the score. So this project was born.

1.1 The idea

I was thinking of a system with multiple cameras pointed at the scoreboard from above. It should use video capturing and image processing to determine where the dart hit the board.

1.2 The structure

The software is supposed to run either on a mini computer like a raspberry pi or any other hardware like a real computer or laptop, running linux, next to the dart board. There are several webcams (2-4) hooked up to the system which are above the board. You are using a normal steel dart board and normal steel darts and you will not have to buy any special dart hardware. The complete system might be built into a sleek compartment housing all together with additional LED-strips for good lighting (Also refer to *the building example* for details).

1.3 A few words on the coding language

As I am using python a lot at work it came naturally to me to use python here as well. Also I think it will address a broad community if someone wants to contribute.

1.4 Ready to start?

So you feel like you want to test it out yourself? Then go ahead and check out how to *install* everything.

2.1 Git

It is recommended to clone the project with git. So you will need to install git for your specific operating system. Afterwards you can clone the project like so:

```
git clone https://github.com/patrickhener/puffyCV.git
```

2.2 Python

As the first requirements you will need *Python3* and *Python3-Pip*. You might wanna refer to the official documentation on how to install and for your specific operating system.

2.3 Virtual Environment

It is recommended to use a virtual environment to keep your pip clean. If you do not want to use one just continue with the next section.

To install and activate the virtual environment do the following:

```
pip install --user virtualenv  
python -m venv ./venv  
. venv/bin/activate
```

2.4 Install python package requirements

To install the requirements you can use *pip* again like shown in the code block below. Keep in mind that you can do this either within a virtual environment or not.

```
pip install --user -r requirements.txt
```

You should now have everything in place to *calibrate* and *run* puffyCV.

Calibration

For the calibration you can run:

```
./puffyCV.py cal <camera-id>
```

Be sure to calibrate one camera at a time. To find out which camera IDs are recognised by your host system just run the shell-script included in the root directory:

```
./enumerate_cams.sh

by Patrick Hener
patrickhener@gmx.de
Version: 0.9, 2019-2020
http://puffycv.rtf.d.io/

I 2019-12-09 14:55:38 Video Device with ID 0 is an accessible camera
E 2019-12-09 14:55:38 Video Device with ID 1 is not an accessible camera
I 2019-12-09 14:55:38 Video Device with ID 2 is an accessible camera
E 2019-12-09 14:55:38 Video Device with ID 3 is not an accessible camera
I 2019-12-09 14:55:38 Video Device with ID 4 is an accessible camera
E 2019-12-09 14:55:38 Video Device with ID 5 is not an accessible camera
E 2019-12-09 14:55:38 Video Device with ID 6 is not an accessible camera
E 2019-12-09 14:55:38 Video Device with ID 7 is not an accessible camera
E 2019-12-09 14:55:38 Video Device with ID 8 is not an accessible camera
E 2019-12-09 14:55:38 Video Device with ID 9 is not an accessible camera
```

3.1 Settings

CHAPTER 4

Run and Deploy

4.1 Running for testing purposes

4.2 Deploying for production

CHAPTER 5

Building Example

CHAPTER 6

TODOs

This chapter will describe open TODOs and will therefore illustrate some kind of a roadmap.

CHAPTER 7

Changelog

7.1 2019-11-29 - v0.9: Alpha version

Working condition: Not working at all. Just the basic project structure