QR Code Challenge:

Hardware: Raspberry pi 3B + Logitech C270

Environment: Raspbian GNU/Linux 9 (stretch) + OpenCV 3.3.0 + Python 3.5.3

Step1: Install opencv3: <https://github.com/Tes3awy/OpenCV-3.2.0-Compiling-on-Raspberry-Pi>

<https://www.pyimagesearch.com/2016/04/18/install-guide-raspberry-pi-3-raspbian-jessie-opencv-3/>

Step2: Install Zbar: <https://www.pyimagesearch.com/2018/05/21/an-opencv-barcode-and-qr-code-scanner-with-zbar/>

<https://github.com/NaturalHistoryMuseum/pyzbar/blob/master/pyzbar/pyzbar.py>

2 methods solve decode QR code:

1. Decode QR code directly with pyzbar:

pyzbar\_decode.py: python3 pyzbar\_decode.py

1. Detect QR code with OpenCV then decode with pyzbar:

detect\_decode.py: python3 detect\_decode.py

2 methods solve angle facing camera:

1. Calculate angle in proportion
2. Find 4 point correspondences -> find homography matrix -> decompose homograph get rotation matrix -> find euler

Reference:

<https://docs.opencv.org/3.4/d9/d0c/group__calib3d.html#ga7f60bdff78833d1e3fd6d9d0fd538d92>

<https://www.learnopencv.com/homography-examples-using-opencv-python-c/>

<https://www.learnopencv.com/rotation-matrix-to-euler-angles/>

<http://answers.opencv.org/question/89786/understanding-the-camera-matrix/>

<https://stackoverflow.com/questions/43364900/rotation-matrix-to-euler-angles-with-opencv>

<https://www.learnopencv.com/rotation-matrix-to-euler-angles/>

<http://dsynflo.blogspot.com/2014/10/opencv-qr-code-detection-and-extraction.html>