Final Report

Game Development Project

sean khanna – Q11279516

Level 6

Contents

[What? 2](#_Toc6911438)

[How? 2](#_Toc6911439)

[Who? 2](#_Toc6911440)

[Does the product meet the set goals and objectives? 2](#_Toc6911441)

[Process used to verify goal completion 2](#_Toc6911442)

[Why some goals might not have been met? 2](#_Toc6911443)

[What I could do next time to complete those goals 2](#_Toc6911444)

[If I was to do it again what changes to my schedule would I make 2](#_Toc6911445)

[What went right and what went wrong? 2](#_Toc6911446)

[Right 2](#_Toc6911447)

[Wrong 3](#_Toc6911448)

[Improvements 3](#_Toc6911449)

[Improvements made by request (with proof) 3](#_Toc6911450)

[What I am going to do with the project in the future 3](#_Toc6911451)

[What did I learn 3](#_Toc6911452)

[What could I have done differently during development 3](#_Toc6911453)

[References 4](#_Toc6911454)

[Appendix 5](#_Toc6911455)

[Appendix A – HacknPlan 5](#_Toc6911456)

[Appendix A.1 – HacknPlan Backlog 5](#_Toc6911457)

[Appendix B – GitHub 6](#_Toc6911458)

[Appendix C – Black Box Testing 6](#_Toc6911459)

[Appendix D – White Box Testing 6](#_Toc6911460)

# What?

This project started out as a new tool for facial recognition detection, were I would use certain algorithms and have ways in which the user or developer could manipulate and image or frame with filters and other effects. However, it turns out that creating such a tool meant that the project size was just too great for the time frame allowed. Therefore, the projects goal altered slightly, instead of creating a facial recognition tool, a premade tool was used and applied to the own program.

This projects scope allowed enough time to learn the new library and apply some of its features for a tech demo, with prospects for future use in actual games and apps.

# How?

The project uses a library called OpenCV which is a pre-made library with all sorts of functions methods to manipulate and detect features. For its face detection, the Local Binary Pattern (LBP) was employed rather than Haar-Cascades; the Local Binary Pattern is three times faster at doing calculations because it uses integers rather than floats which Haar-Cascades do. However, Haar-Cascades can be more accurate than the Local Binary Pattern but, for real time detection the Local Binary Pattern surpasses Haar-Cascades.

On the other hand, Haar-Cascades was used to apply eye detection as there was no trained data using the Local Binary Pattern nevertheless, I have prospects for new trained data which will be spoken about later in this report.

## Haar-Cascades

Haar-Cascades simply use an XML document that has been trained for a specific feature with a set of negative and positive images and with these you can input an image into this classifier, superimposing it to the image a return the features locations.

Figure 1 Haar-Cascades, 2019

## Local Binary Pattern

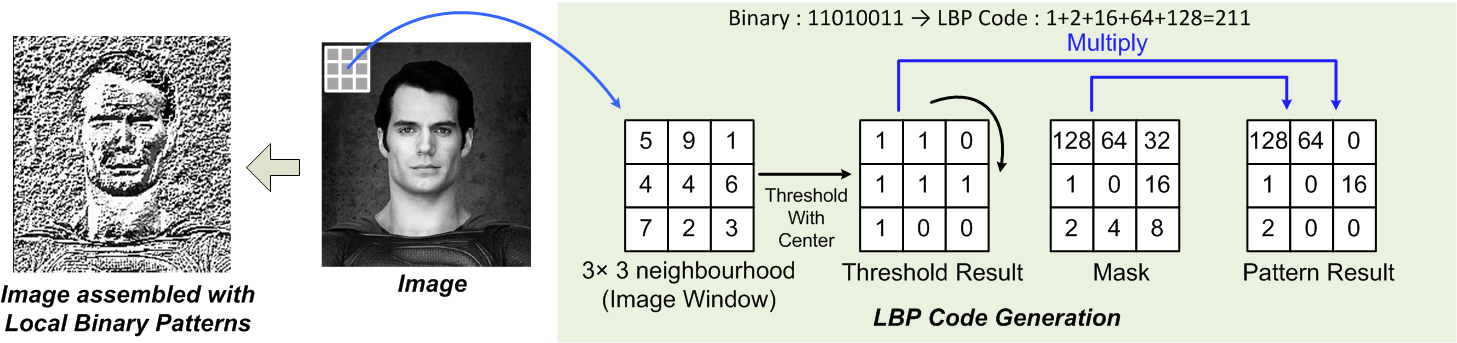
The local binary pattern begins by converting the greyscale image into an image using binary. The first step is to take a 3x3 neighbourhood and take the centre pixel as the threshold, with this you can convert the surrounding pixels into 1s and 0s; apply the mask; eventually you will have an image consisting with multiple binary numbers which can then be used to extrapolate features such as a face or eye outline using histograms.

Figure Local Binary Pattern, 2019

# Who?

This project aims to become a program that people with disabilities can use just by using there face and eyes, simply by using a normal web camera rather than, for example, purchasing a camera from [tobii](https://www.tobii.com/) (2019), making this whole experience a lot cheaper.

# Goals, Objectives and Processes

Within this

## Why some goals might not have been met?

* rotating the head
  + Found a solution called solvePnP
* Training my own haar/LBP cascades
  + Dlib and open cv
    - Using points on the face

### What I could do next time to complete those goals

### If I was to do it again what changes to my schedule would I make

# What went right and what went wrong?

## Right

* Use the gpu

## Wrong

## Improvements

# Improvements made by request (with proof)

# What I am going to do with the project in the future

* Areas of improvement
  + TBB

# What did I learn

# What could I have done differently during development

# References

tobii, 2019. The world leader in eye tracking [viewed 23/04/2019]. Available at: <https://www.tobii.com/>

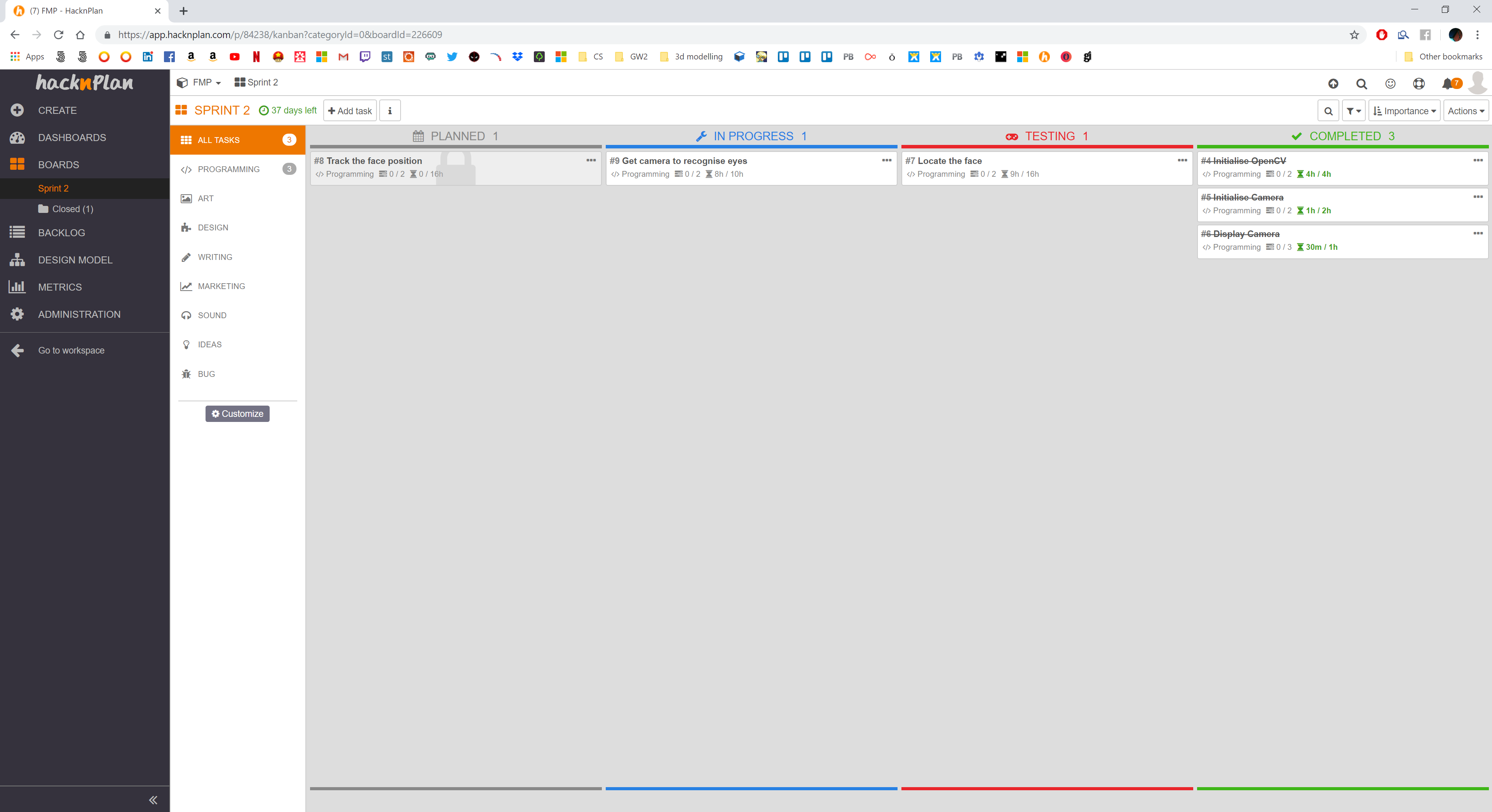
OpenCV, 2019. Cascade Classifier Training [viewed on 23/04/2019]. Available at: <https://docs.opencv.org/3.3.0/dc/d88/tutorial_traincascade.html>

Viola P., Jones M., 2004. Object Detection Using a Boosted Cascade of Simple Features [viewed on 23/04/2019]. Available at: <http://www.merl.com/publications/docs/TR2004-043.pdf>

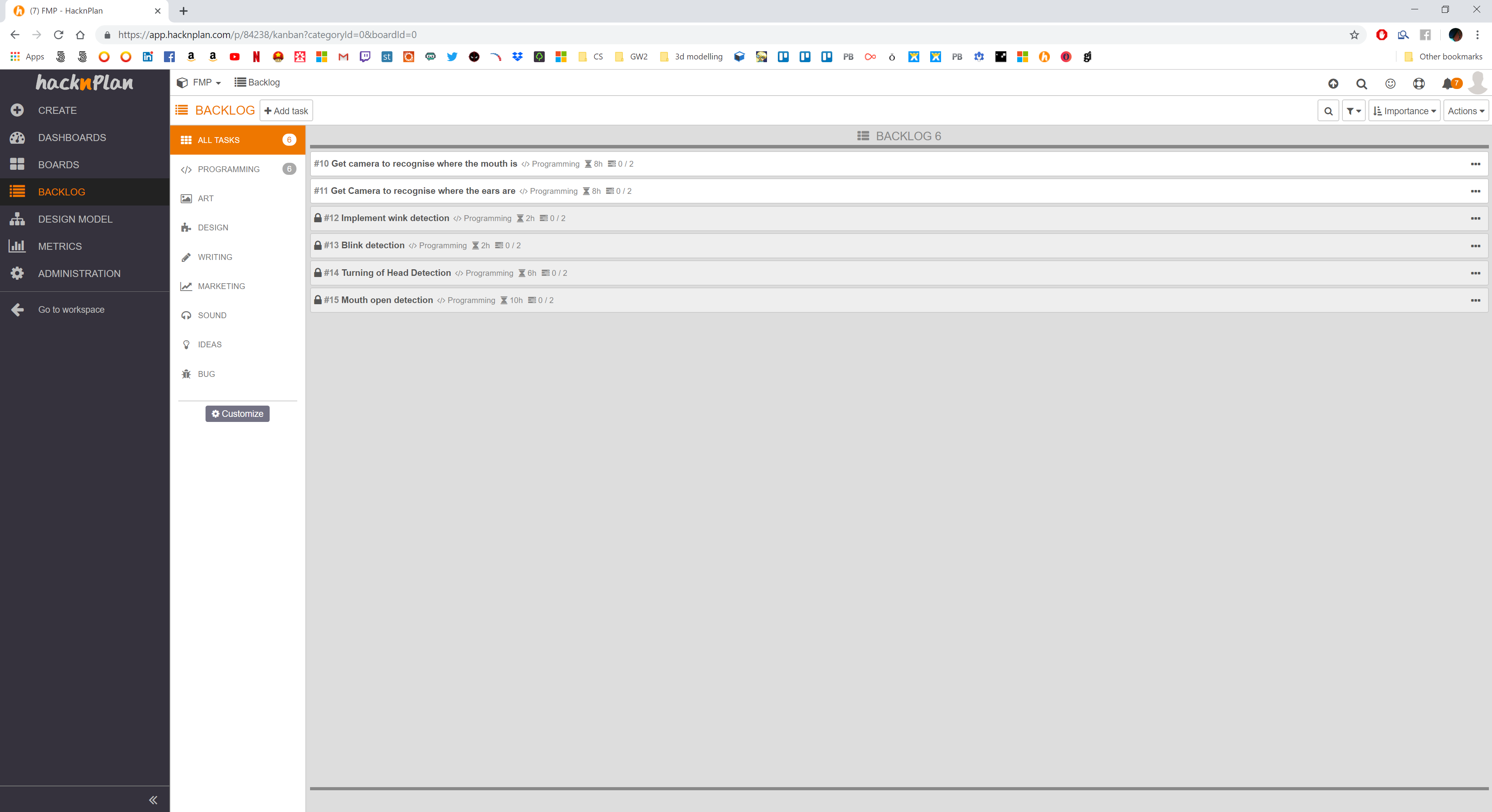
Christos Kyrkou, 2017. Object Detection Using Local Binary Patterns [viewed on 23/04/2019]. Available at: <https://medium.com/@ckyrkou/object-detection-using-local-binary-patterns-50b165658368>

# Appendix

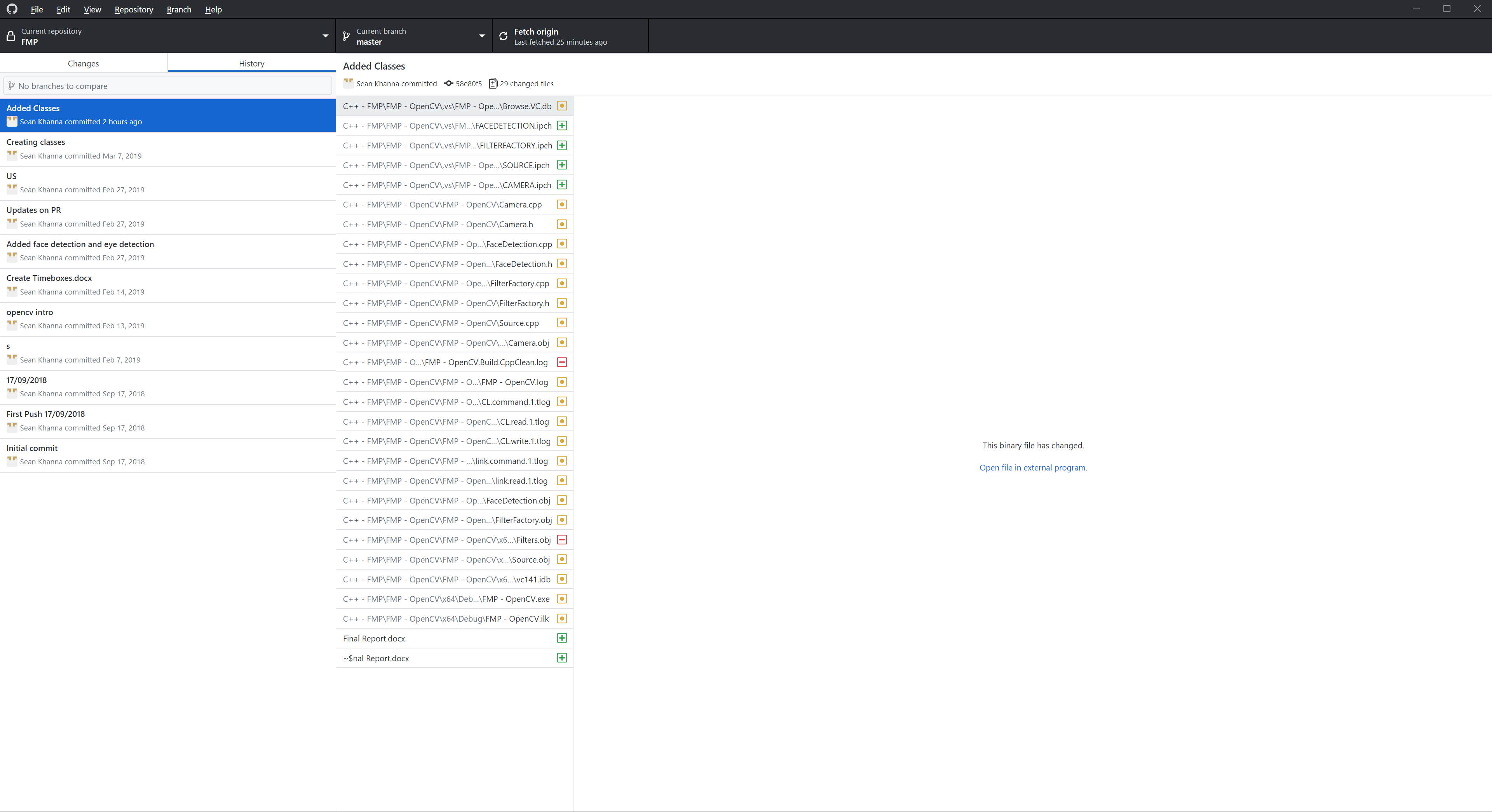
## Appendix A – HacknPlan



### Appendix A.1 – HacknPlan Backlog



## Appendix B – GitHub



## Appendix C – Black Box Testing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test ID | Feature | Test | Expected | Actual | Solution |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## Appendix D – White Box Testing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |