Final Report

Game Development Project

sean khanna – Q11279516

Level 6

Contents

[What? 2](#_Toc7525020)

[How? 2](#_Toc7525021)

[Haar-Cascades 2](#_Toc7525022)

[Local Binary Pattern 2](#_Toc7525023)

[Market 3](#_Toc7525024)

[Goals, Objectives and Processes 3](#_Toc7525025)

[Why some goals might not have been met? 3](#_Toc7525026)

[What I could do next time to complete those goals 3](#_Toc7525027)

[If I was to do it again what changes to my schedule would I make 3](#_Toc7525028)

[What went right and what went wrong? 3](#_Toc7525029)

[Right 3](#_Toc7525030)

[Wrong 3](#_Toc7525031)

[Improvements 3](#_Toc7525032)

[Improvements made by request of client 3](#_Toc7525033)

[Tracking 3](#_Toc7525034)

[Future Prospects 4](#_Toc7525035)

[What did I learn 4](#_Toc7525036)

[What could I have done differently during development 4](#_Toc7525037)

[References 5](#_Toc7525038)

[Appendix 6](#_Toc7525039)

[Appendix A – HacknPlan 6](#_Toc7525040)

[Appendix A.1 – HacknPlan Backlog 6](#_Toc7525041)

[Appendix A.2 – HacknPlan Activity Logs (per sprint) 7](#_Toc7525042)

[Appendix B – GitHub 9](#_Toc7525043)

[Appendix B.1 – GitHub Commits 10](#_Toc7525044)

[Appendix C – Black Box Testing 10](#_Toc7525045)

[Appendix D – White Box Testing 10](#_Toc7525046)

[Appendix E – CPU and GPU transfer 11](#_Toc7525047)

[Appendix E.1 CPU 11](#_Toc7525048)

[Appendix E.2 GPU 11](#_Toc7525049)

# 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

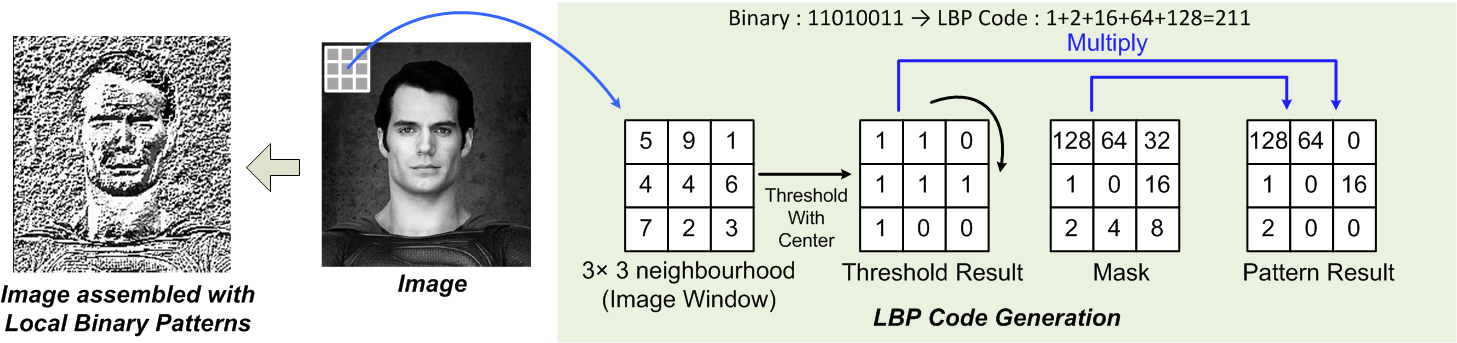
The local binary pattern begins by converting the greyscale image into an image using binary. The first step is to take a 3x3 neighbourhood and take the centre pixel as the threshold, with this you can convert the surrounding pixels into 1s and 0s; apply the mask; eventually you will have an image consisting 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

# Market

This project aims to be 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 of client

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Request | Success or Fail? | Evidence | Notes |
| 1 | Move workload over to GPU | Success and Fail | See [Appendix E.1](#_Appendix_E.1_CPU) for CPU and [Appendix E.2](#_Appendix_E.2_GPU) for GPU. | Some of the work was easily transferrable to the GPU however, the algorithm calculations are still being calculated on the CPU. |
|  |  |  |  |  |
|  |  |  |  |  |

# Tracking

Throughout any project backups are always key, making sure you save your work is one thing but, what happens when something goes wrong and you lose all your work, if you don’t have a backup then you must start from scratch. As BullGuard (2019) would put PC security, “Having duplicate copies of your most important information saved in a remote location keeps it safe in case anything goes badly wrong with your computer.”. Therefore, this project uses an open source program called GitHub. A place where it can be swiftly uploaded to, safely and securely, and with a means to get access to previous iterations of uploads, as Laurence Bradford (2019) puts it “…like a cloud for code”. Tracking is extremely important in making sure that a project stays on track (see [Appendix B.1](#_Appendix_B.1_–)) however, GitHub doesn’t offer any tracking apart from commits therefore to further aid in the project

# Future Prospects

As I have said previously, this program needs to be optimised greatly, all of the calculations, algorithms, filters, rendering and so on, requires massive amounts of computing power to swiftly return results. In turn, the project managed to be split between the GPU as well as the CPU which helped greatly with the optimisation however, it did not solve the problem. After thorough research, I learned a lot about another template library, widely used by many and most programs, Threading Building Blocks (TBB, 2017), it allows programs to “take full advantage of multicore performance”. Using this library will allow for code to be written easily and which adapts, based on the number of cores the computer has.

Unfortunately, one of the aims for this project was to eliminate the use of extra hardware being bought and just utilise what consumers already own, this however, is proving to be very difficult with the number of optimisations needed to make it run smoothly. This is one of the reasons why such companies like tobii(2019) use cameras built by their company, as they have the hardware and software built into the camera to make the workload on the computers less.

# 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>

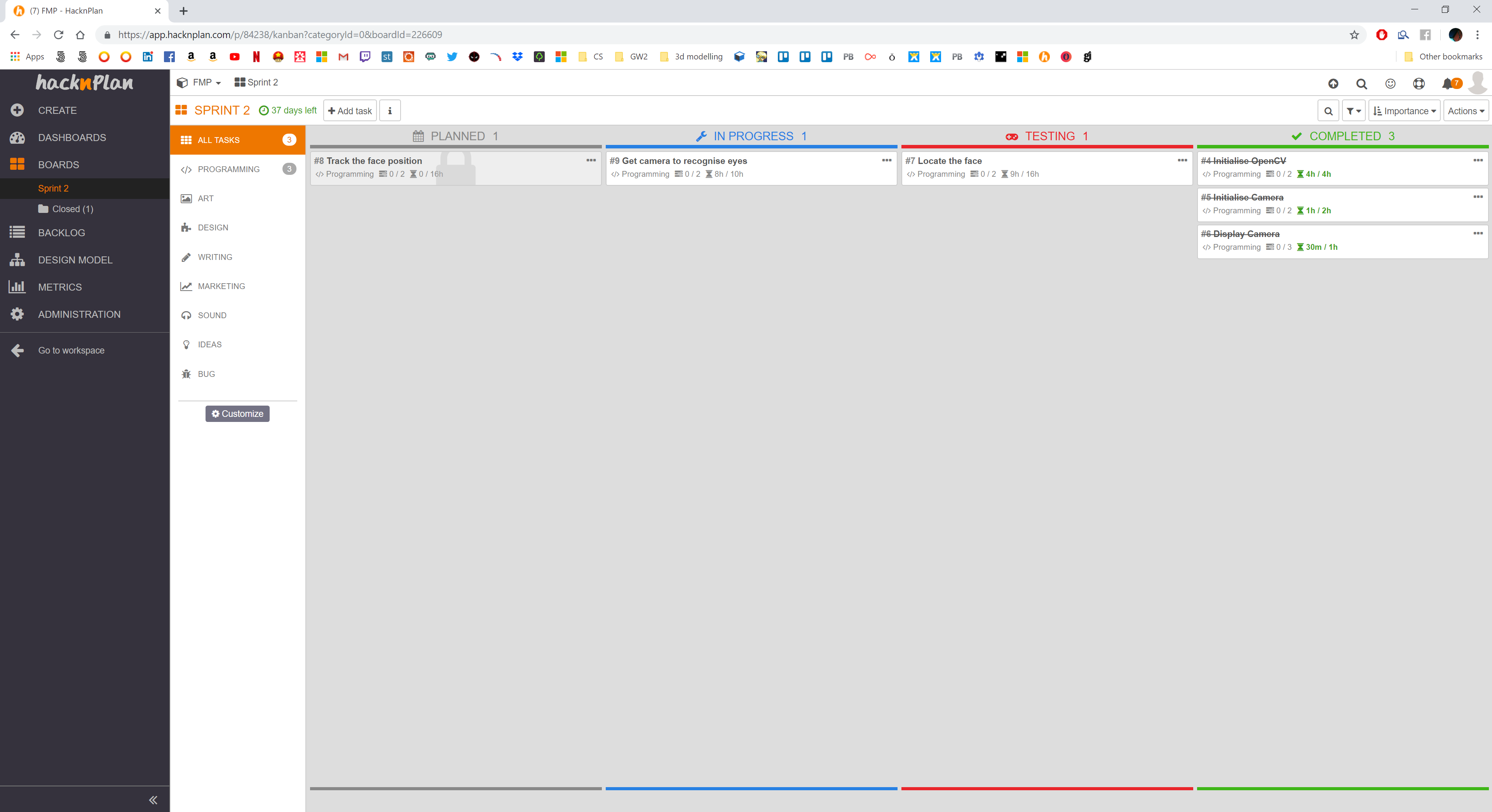
Intel®, 2017. Threading Building Blocks [viewed on 26/04/2019]. Available at: <https://www.threadingbuildingblocks.org/>

BullGuard, 2019. Backup of Data & Files – Why it is Important? [viewed on 26/04/2019]. Available at: <https://www.bullguard.com/bullguard-security-center/pc-security/computer-threats/backup-of-data-files-why-it-is-important.aspx>

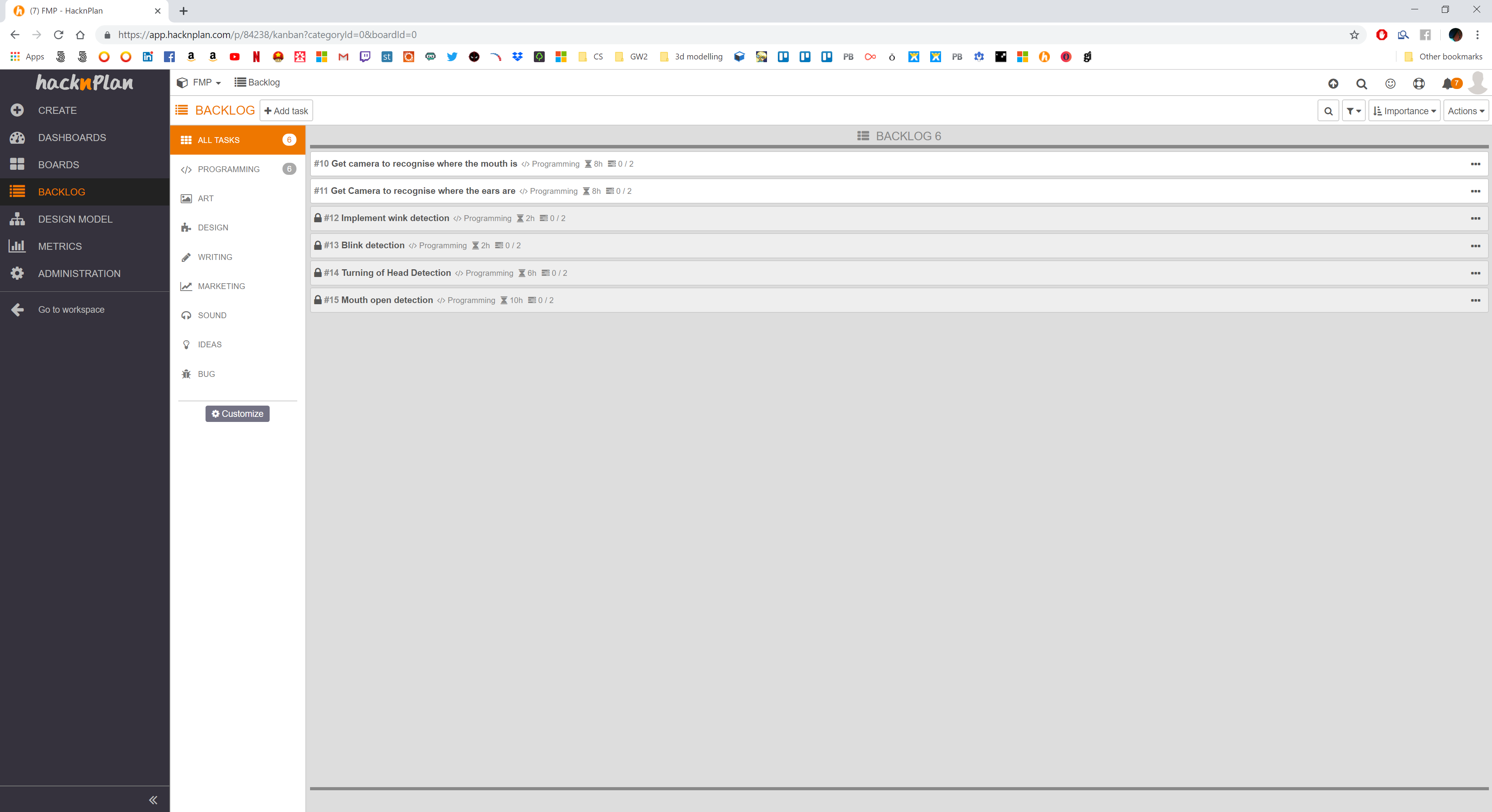
Laurence Bradford, 2019. What Is GitHub, and Why Should I Use It? [viewed on 26/04/2019]. Available at: <https://www.thebalancecareers.com/what-is-github-and-why-should-i-use-it-2071946>

# Appendix

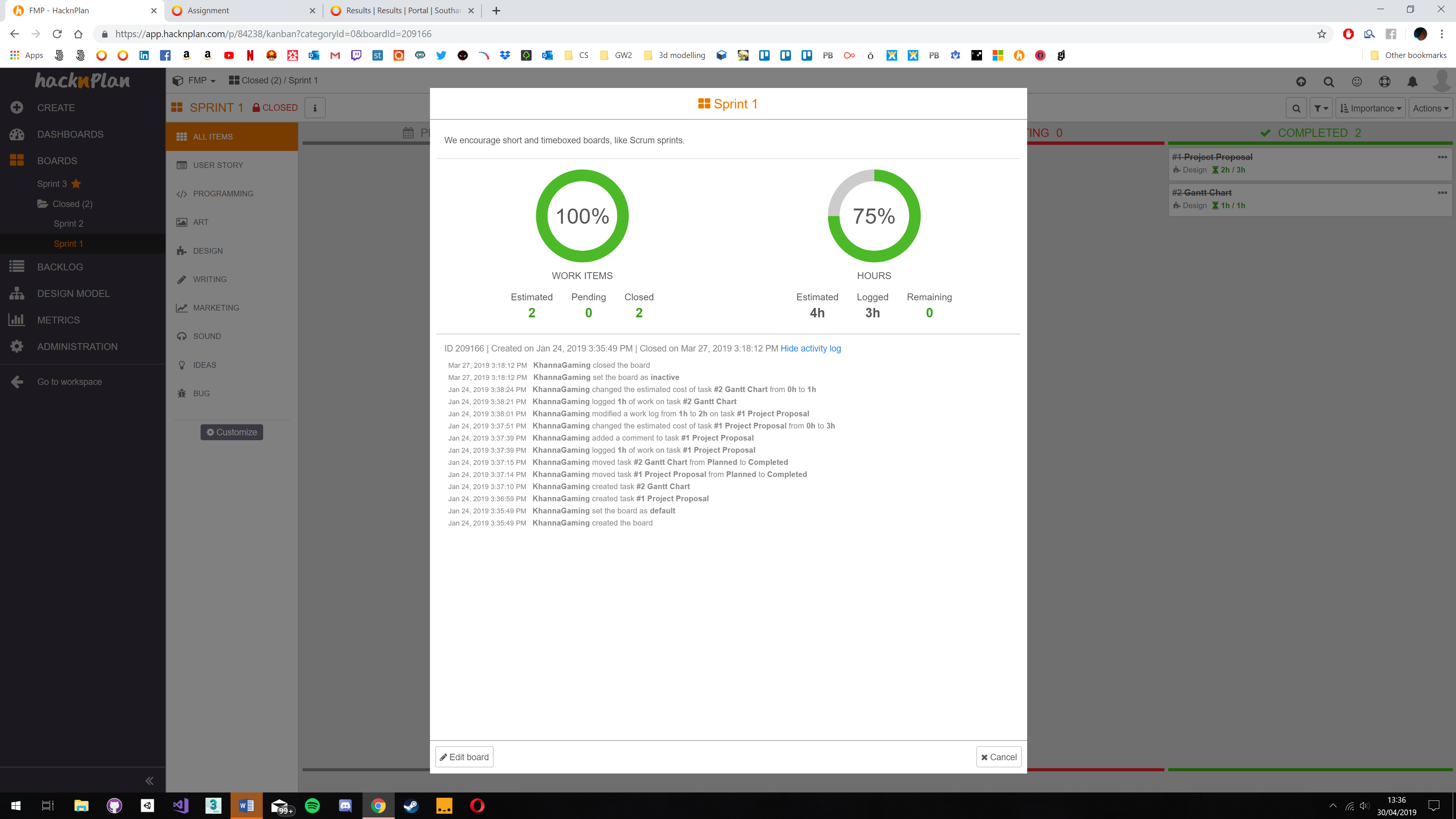
## Appendix A – HacknPlan

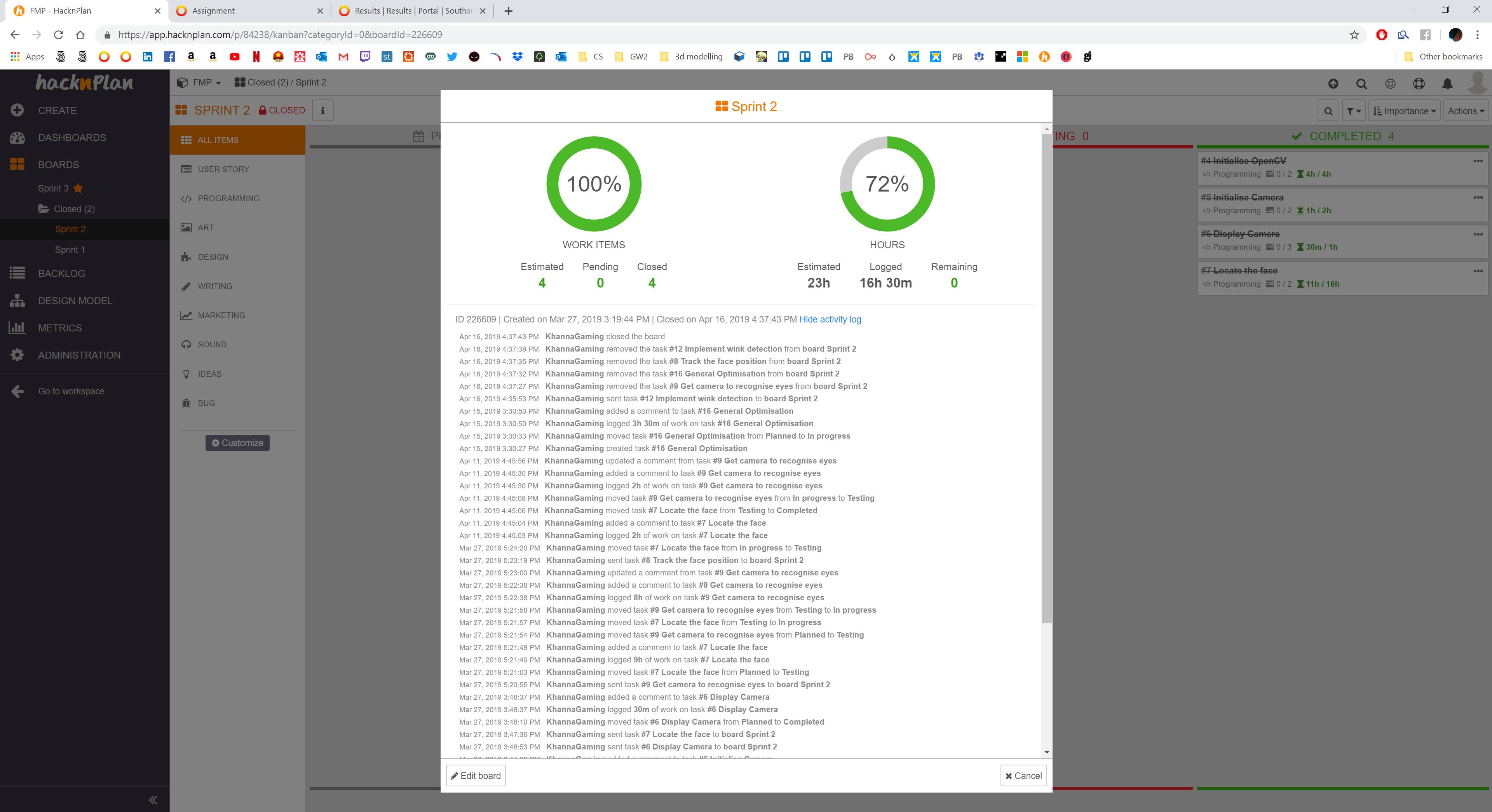


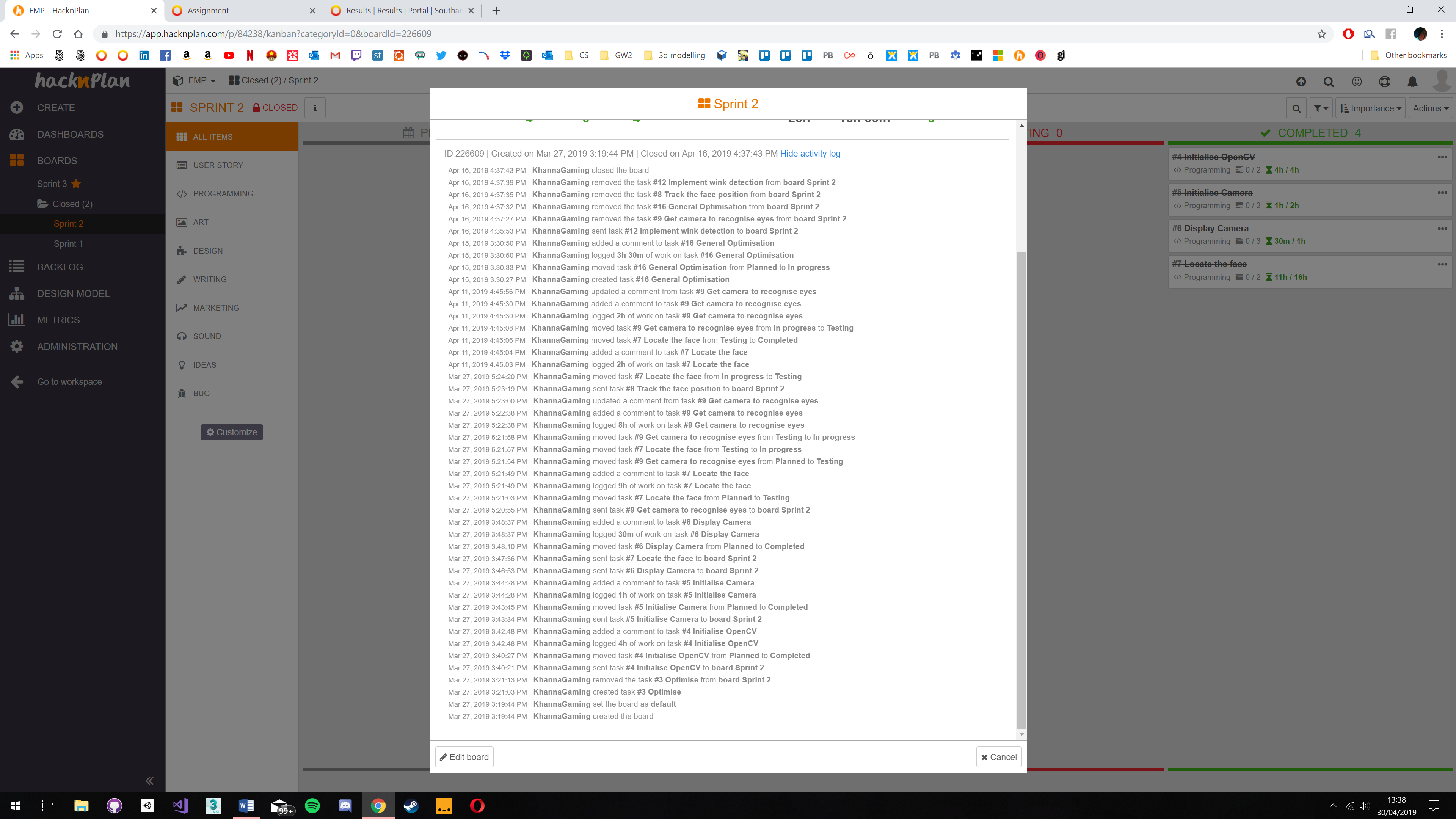
### Appendix A.1 – HacknPlan Backlog

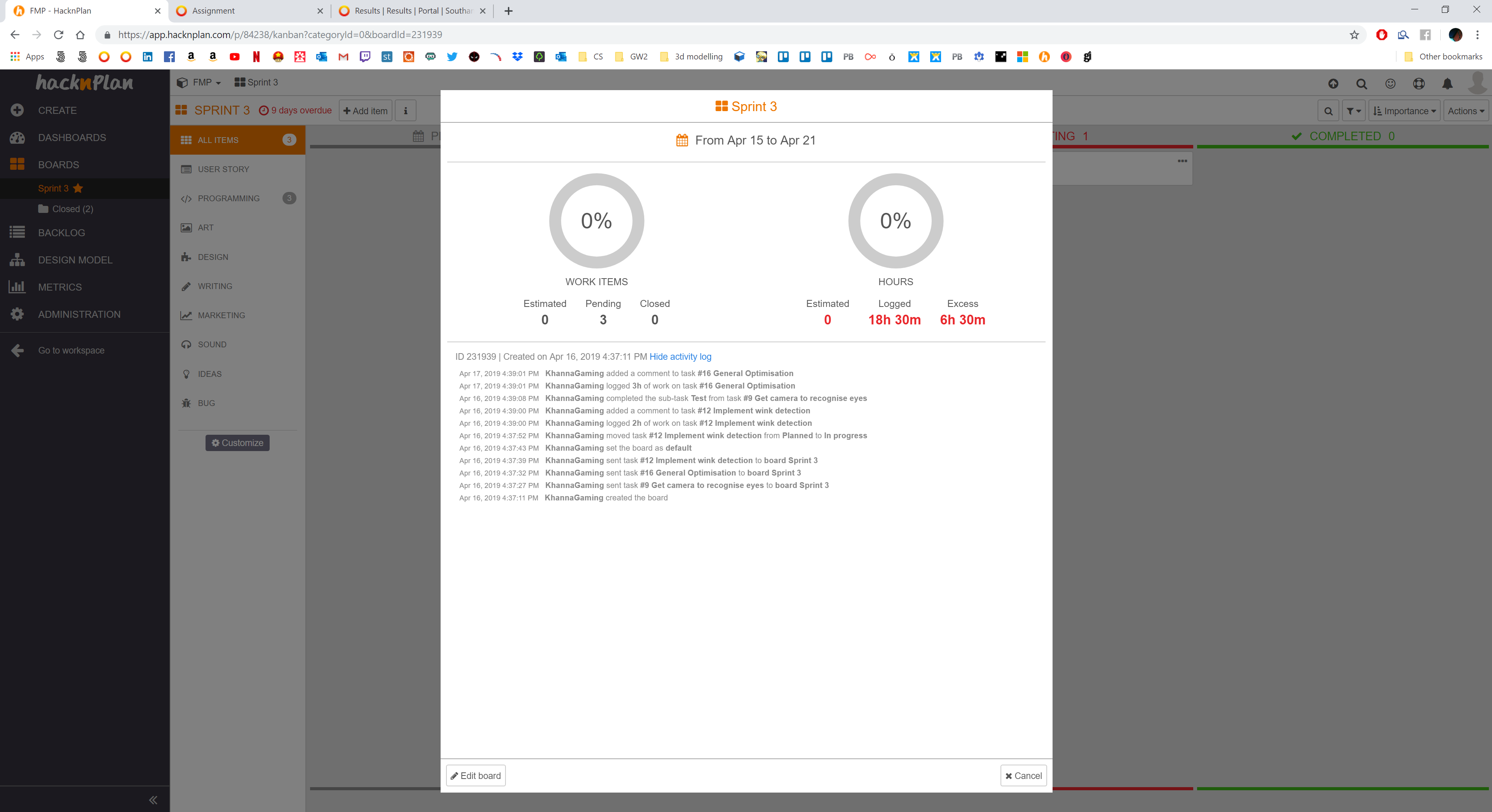


### Appendix A.2 – HacknPlan Activity Logs (per sprint)

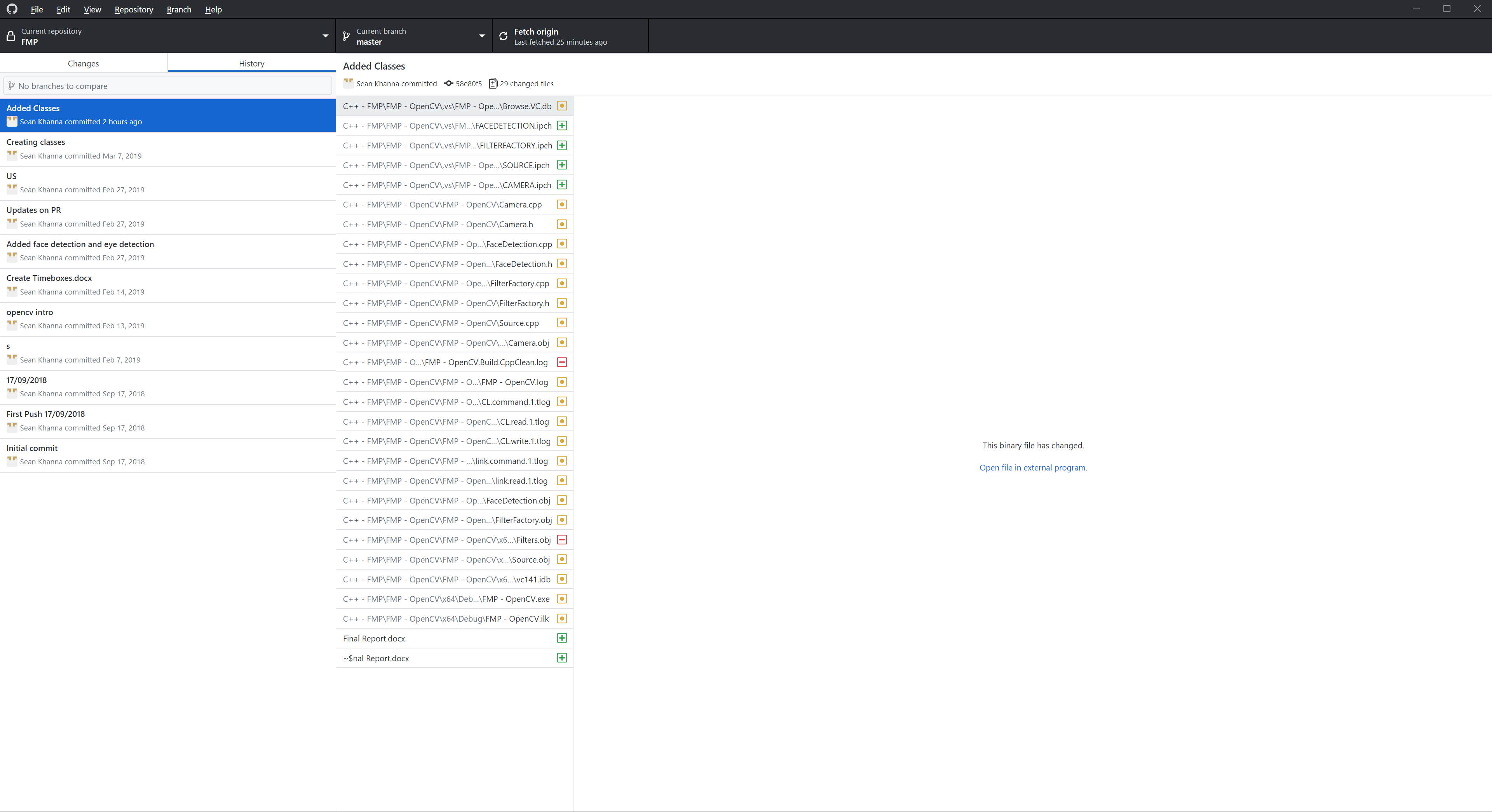




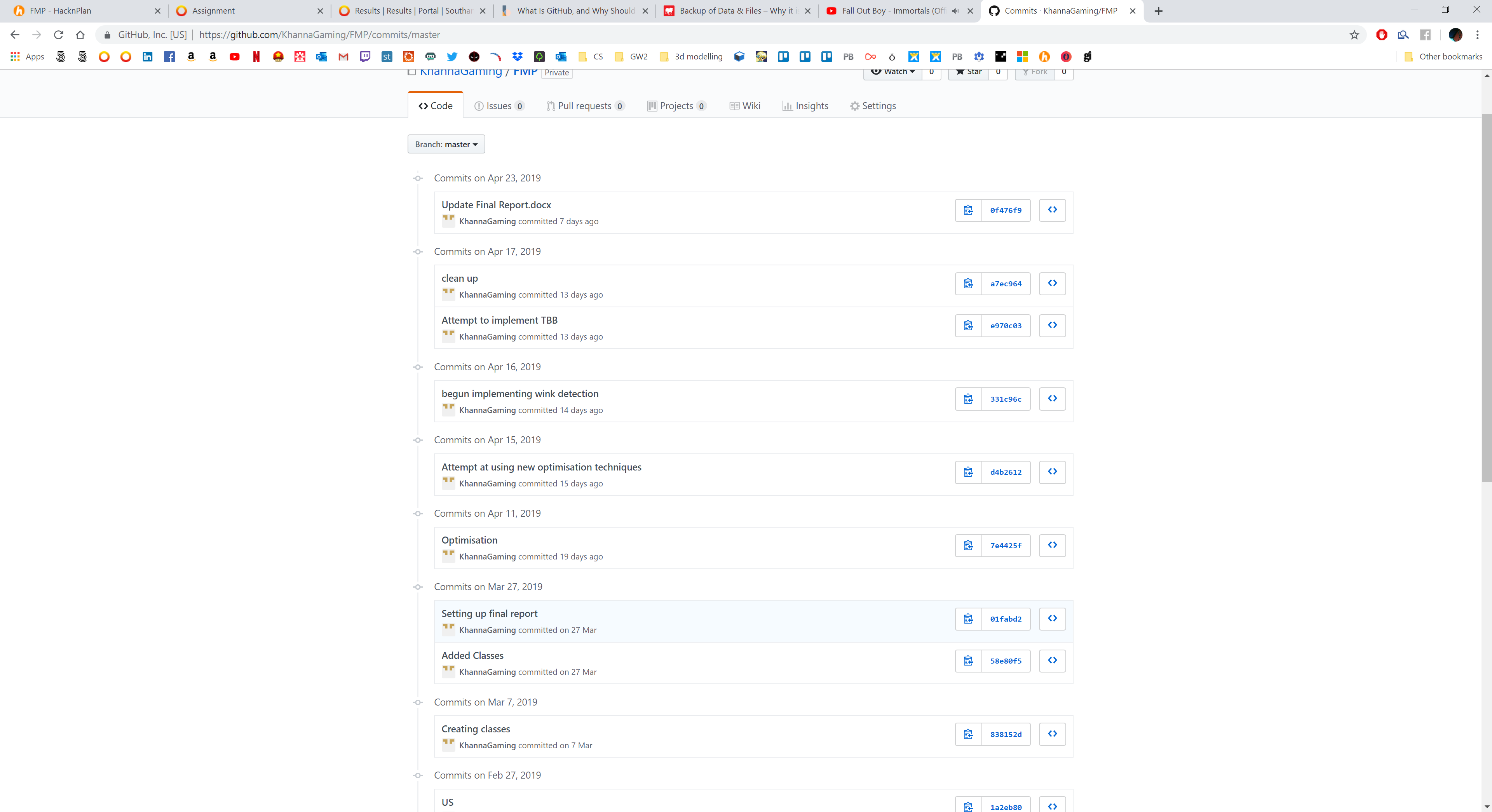


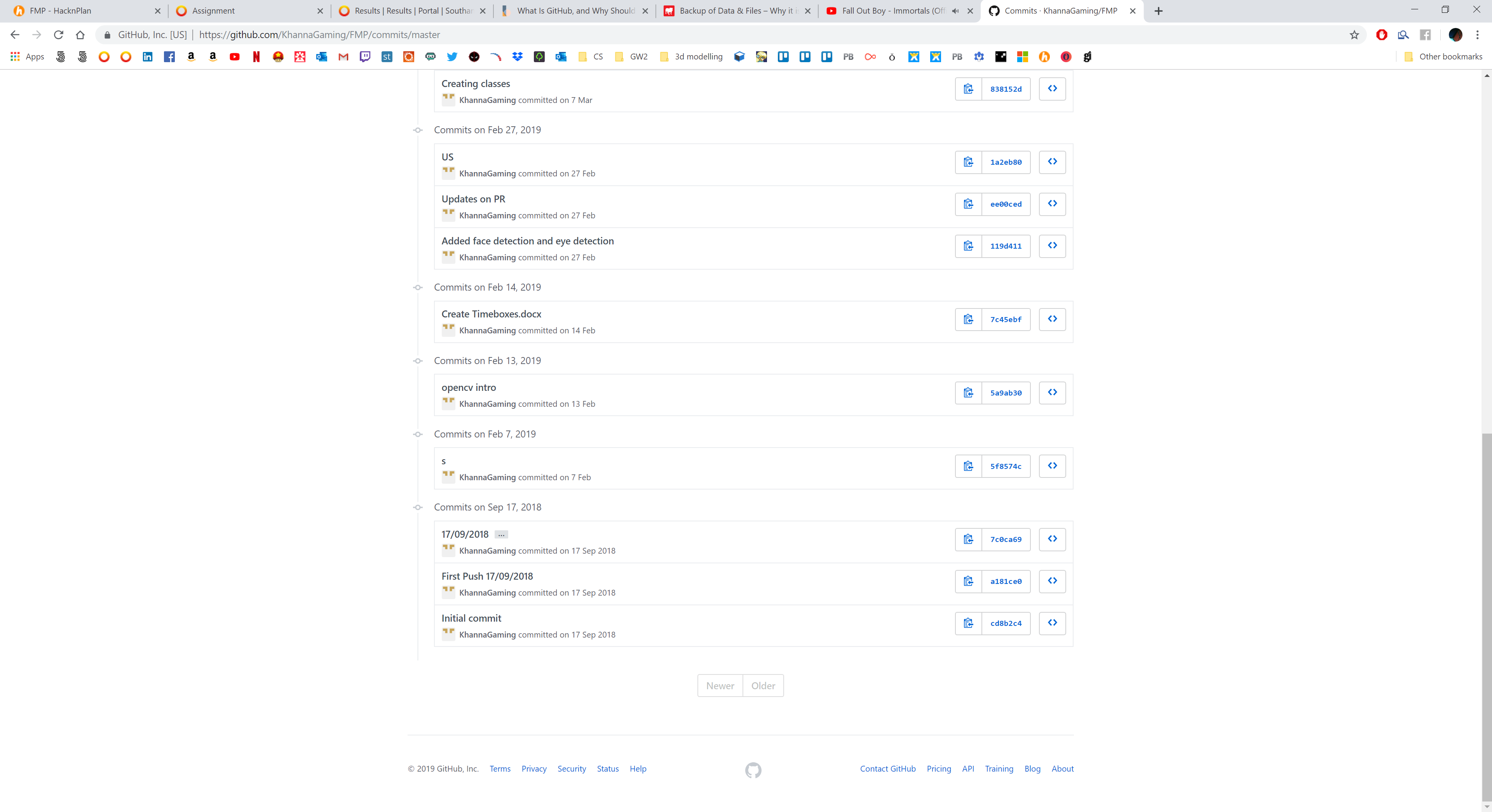


## Appendix B – GitHub



### Appendix B.1 – GitHub Commits





## Appendix C – Black Box Testing

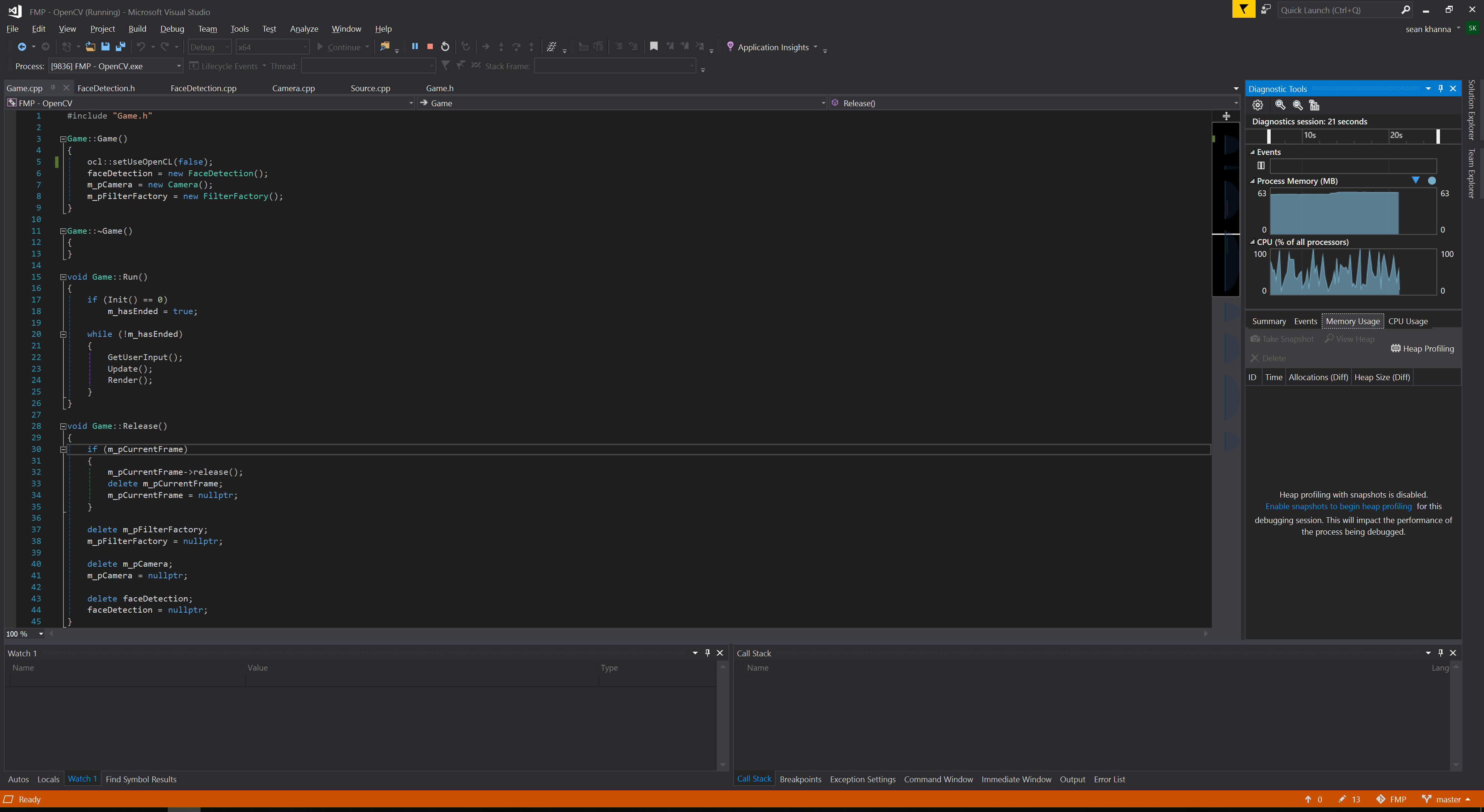
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test ID | Feature | Test | Expected | Actual | Solution |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## Appendix D – White Box Testing

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

## Appendix E – CPU and GPU transfer

### Appendix E.1 CPU



### Appendix E.2 GPU

