UAL CCI Diploma [22/23]

Creative Practice: Open Sourcing + Guides

George Symonds

CCI: Computational Environments

Contents:

Series of Motion Detection Explorations (3)

Intro:

Welcome to my Computational Environments Project!

Coming out of our last unit - Physical Computing, I got started on thinking on how I could expand my computing projects further, upscaling them in 3D Exhibition Space and conxtextualising them in education + creation.

I started thinking about what computational space is for me - a creative; in the frame of one's ability to make their computational ideas come to reality.

I realised that my ability to make my projects come to life are only as good as the pre-existing libraries and tutorials that came before it. Therefore my project is about spreading what I've learnt, so that the next person can develop on my ideas and make their own - improved outcomes.

I will present my project in 2 parts - **The Projects** (Series of Arduino Machine Vision Explorations) and their accompanying **Guidebooks/Tutorials**.

SERIES OF ARDUINO MACHINE VISION EXPLORATIONS

The Physical Computing Outcomes

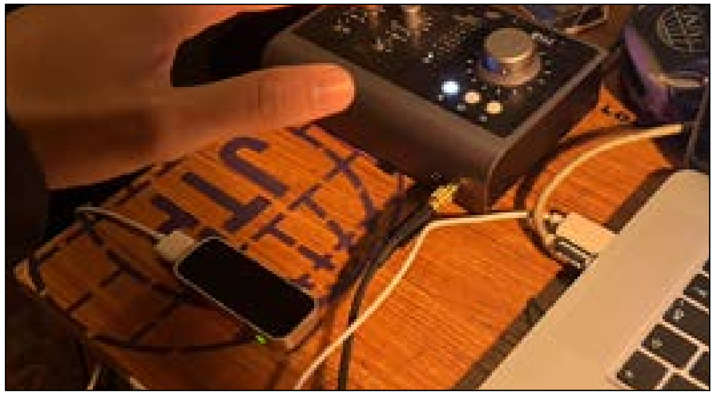
03

NERF FUN SENTRY TURRET

ML Powered home defense system

KEYSCAPE

Gesture Controlled Music Creation

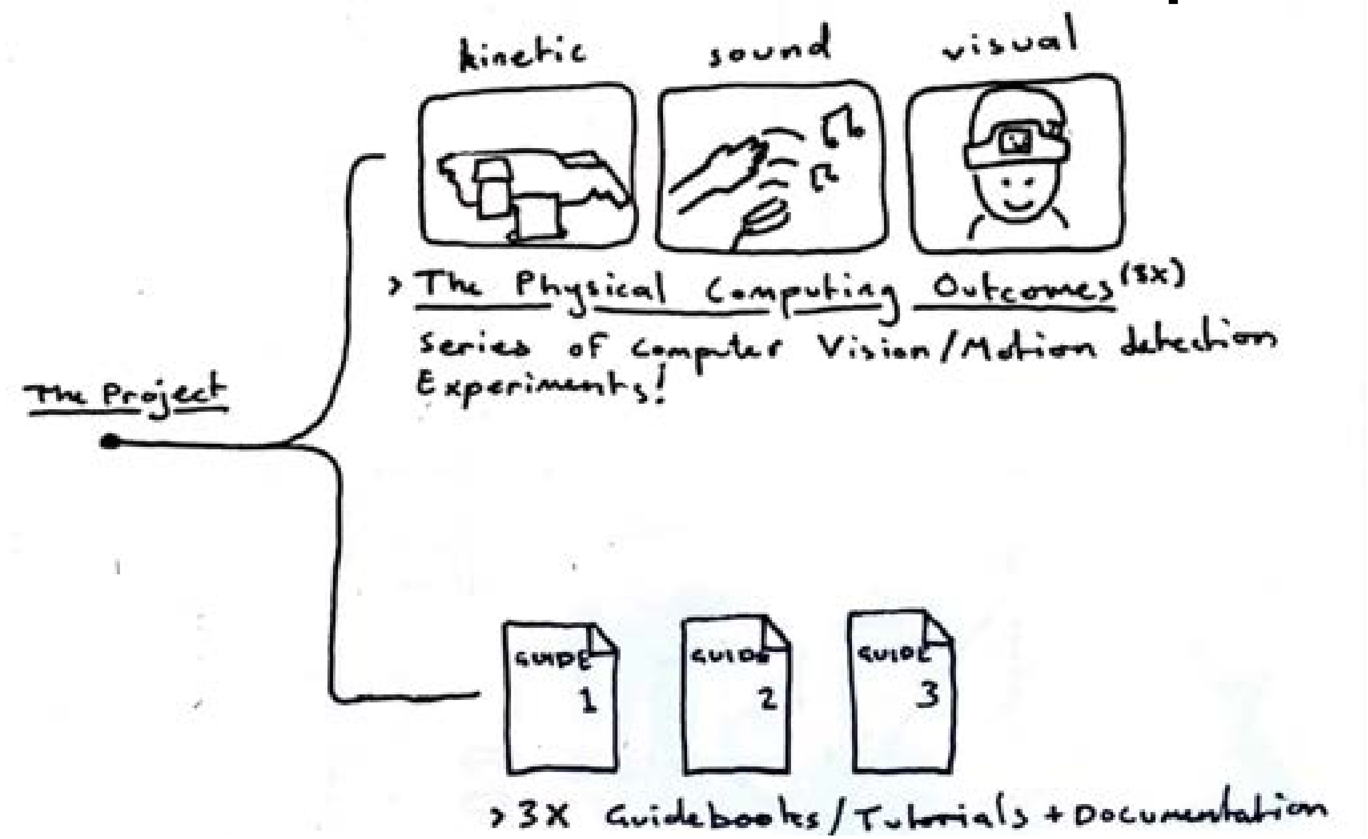


SMILEY BEANIE

The Beanie that smiles back at you

Concept:

The Series of Arduino Machine Vision Explorations (3)

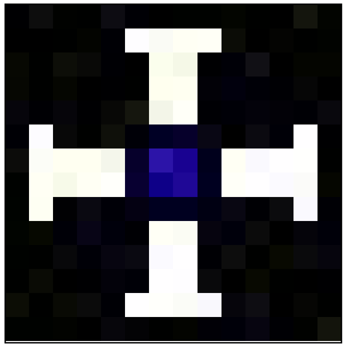

kinetic
sound
visual
> The Physical Computing Outcomes (3x)
Series of Computer Vision/Motion detection Experiments!
The Project
GUIDE 1
GUIDE 2
GUIDE 3
> 3X Guidebooks/Tutorials + Documentation

Open Sourcing Code + Creating Guidbooks/Tutorials:

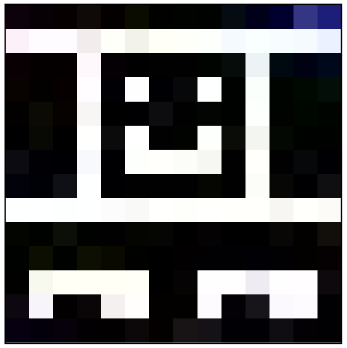
The Guidebooks/Tutorails will contain easy-to-follow comprehensive instructions on how to make my projects, and will be accessible to anyone through my publications or available on my website, all the code will be open-source so that inspect, modify, and enhance it.

THE FINAL MODUAR-BRANDING SYSTEM

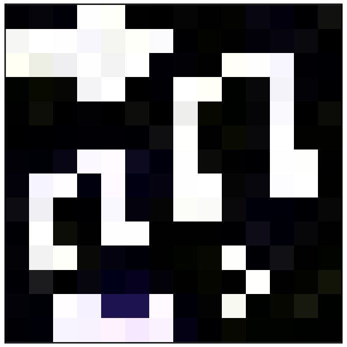
**More on this later*



Kinetic



Visual



Audio

The Projects:

Series of Arduino Machine Vision Explorations (3)

The projects are 3 fun experiments (***right)*** that combine cheap/ accessible hardware with different methods of motion detection + machine learning to solve complex problems.

The Projects Are:
Nerf Gun Sentry Turret, an autolocking and mounted nerf gun that locks on targets using ml5 face-detection and fires. It uses a webcam, p5, servos, led indicators, and requires a laptop to be plugged in.

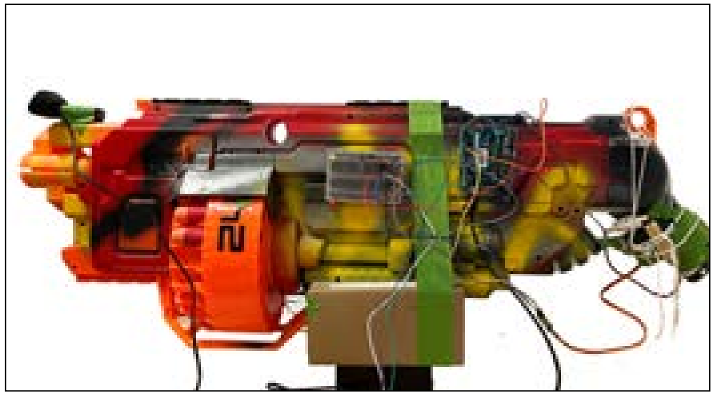
KeyScape, A modified leap motion detector that reads the users' hand (hand x-position + number of extruded fingers) and plays different chords in the chosen scale. Made in processing, it's to be used as a tool in music production. You can play music straight into your DAW or plug in into a keyboard to hack the way notes are inputted.

Smiley Beanie (Prototype). A wearable beanie that has an webcam and an LCD display, which smiles back at you when it detects a smile (using ml5 emotion classification). For now it has to be plugged in but the aim is that It will be able to run independently without a laptop.

I had to make a prototype or this hand-in,due to challenges ill mention later. I aim to fix this for the final degree show

NERF FUN SENTRY TURRET

ML Powered home defense system



KEYSCAPE

Gesture Controlled Music Creation

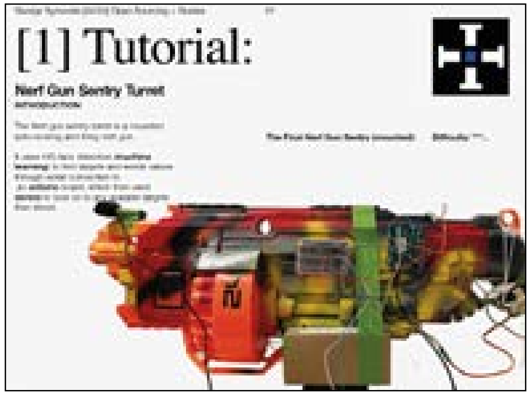
SMILEY BEANIE (Prototype)

The Beanie that smiles back at you

The Guidebooks:

THE FINAL FRONT PAGE OF EACH GUIDEBOOK
The Physical Computing Outcomes

Open Sourcing Code + Creating Guidbooks/Tutorials:


[1] Tutorial:

Nerf Gun Sentry Turret

INTRODUCTION

The Final Nerf Gun Sentry (mounted)

Difficulty

[2] Tutorial:

KeyScape: Gesture Controlled Music Creation

INTRODUCTION

KeyScape

Difficulty

[3] Tutorial:

Smiley Beanie: The Beanie That Smiles Back At You

INTRODUCTION

SmileyBeanie (WIP)

Difficulty

The main key of this project is that it encourages other pople to create.

Therefore I designed easy to follow guides that I will be handing out at the exhibition and available on my website! all the code will be open too n available

The Audience:

people who haven’t coded before/ just got into it

Aims:

The aim is to inspire ideas, and allow for people with no background in coding to get started with Arduino.

Context + Narrative:

Initial Inspiration:

Throughout my journey in learning how to code, I've learnt 2 things,

1. I'll never understand eveything.
2. The projects that I endevour will only be as good as the libraries and documentation that already exists on it.

Inspired by all the countless guides and youtube videos I've watched, I've made my Computational Environments projects on building guides as a framework for further

Open Sourcing + Guides

The main key of this project is that it encourages other people to start using arduino and create.

Which is why the easy-to-follow guides are neccesary and and all the code will be available for anyone to make and develop further!

Keywords:
Education, Documentation, Creativity, Frameworks

Contextualisation
Open-sourcing code and promoting socially engaged, inclusive, and accessible technological development are important aspects of creating a positive and transformative impact in the field of physical computing. By open-sourcing code, you can encourage collaboration, knowledge sharing, and the democratization of technology. This shift in sharing code empowers individuals and communities to actively participate in the development process, fostering a culture of collective learning and innovation.

Open-sourcing code allows others to build upon existing projects, modify them to suit their specific needs, and contribute back to the community. In a world where technological advancements are increasingly shaping our daily lives and becoming closed/ monopolised for profit, it is crucial to ensure that the benefits of these technologies are accessible to everyone, regardless of their background or abilities.

Moreover, open-sourcing code aligns with the principles of transparency and accountability. It enables the public to scrutinize and validate the functionality, security, and ethical implications of technology. This transparency helps to build trust and ensure that technological advancements are driven by the common good rather than solely for self-beneficial gains.

Inspiration:

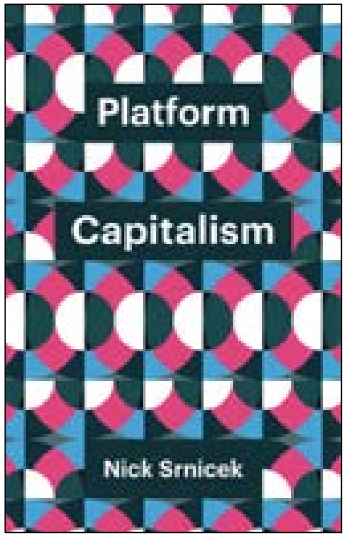
Open Sourcing Code + The Future of technolody:

PIANO STAIRS
The Fun Theory
2022

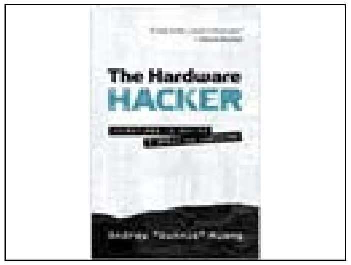
THE WORLDS DEEPEST BIN
The Fun Theory
2022

PLATFORM CAPITALISM
Nick Srnicek
2016

“Platforms, in sum, are a new type of firm; they are characterised by providing the infrastructure to intermediate between different user groups”
Today more and more of technology is being monopolised and as a product of capitalism are all about profit.



Hannah Perner-Wilson (Plusea): is an artist, designer, and educator known for her work in wearable technology and e-textiles. She actively shares her projects, code, and tutorials on platforms like GitHub, encouraging others to learn, remix, and build upon her work.



Andrew “bunnie” Huang: Andrew Huang is an engineer, entrepreneur, and hardware hacker. He has contributed significantly to the open-source hardware movement, advocating for the sharing of schematics, designs, and documentation. His book, “The Hardware Hacker,” delves into the importance of open-source hardware for innovation.

Tutorial :TinyML with OV7670 Camera module
Vishwa
hackster.io

Vishwa
Published September 2, 2022 © GPL3+

Tutorial : TinyML with OV7670 Camera module!

Want to build your own TinyML image recognition application? This blog contains all the necessary details for you to get started

Journey + Development:

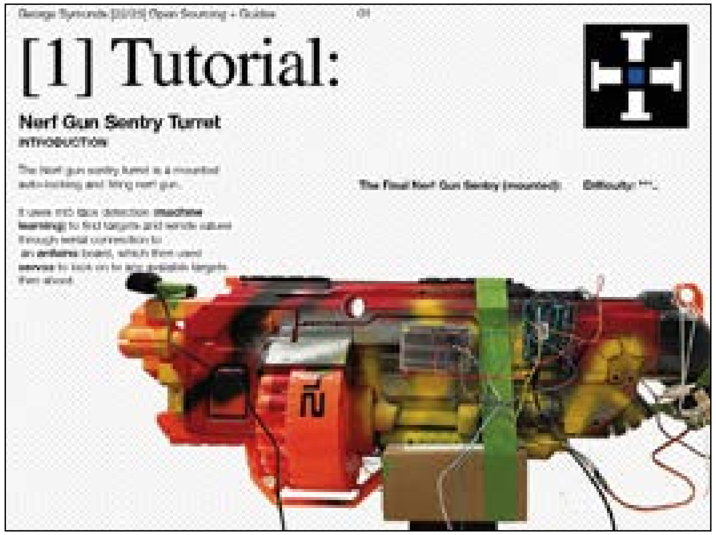
The Process of this project went as follows:

1. The Project Contextual development

2. Creation of the Physical Computing Outcomes
    2.1. The Nerf Gun Sentry Turret
    2.2. KeyScape
    2.3. SmileyBeanie

3. Bringing the Project Back Together
    3.1. Project Branding
    3.2. Creation of the Guidebooks

The order of “Journey + Development” will follow this ^



George Symonds [22/23] Open Sourcing + Guides 01

[1] Tutorial:

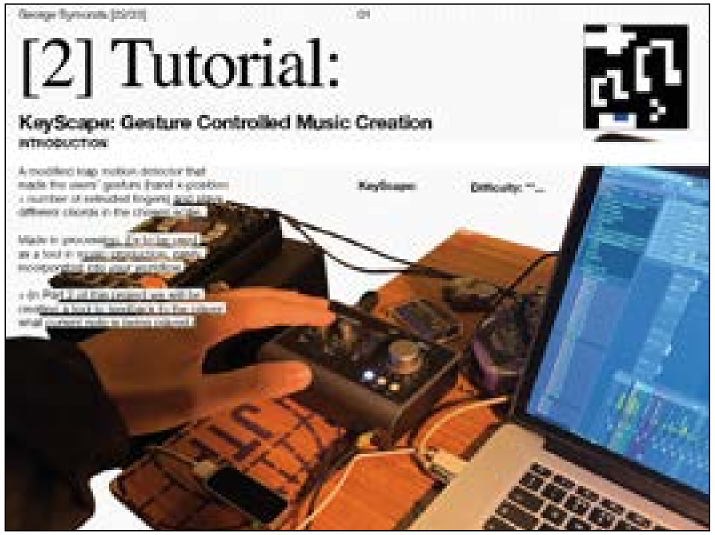
Nerf Gun Sentry Turret

INTRODUCTION

The Nerf gun sentry turret is a mounted auto-locking and firing nerf gun.

It uses ml5 face detection (**machine learning**) to find targets and sends values through serial connection to an **arduino** board, which then used **servos** to lock on to any possible targets then shoot.

The Final Nerf Gun Sentry (mounted): Difficulty: **..



George Symonds [22/23] 01

[2] Tutorial:

KeyScape: Gesture Controlled Music Creation

INTRODUCTION

A modified leap motion detector that reads the users' gesture (hand x-position + number of extended fingers) and plays different chords in the chosen scale.

Made in processing, it is to be used as a tool in music production, easily incorporated into your workflow.

+ (in Part 2 of this project we will be creating a tool to feedback to the player what current note is being played)

KeyScape: Difficulty: **..

[1] Contextual Development:

[1.1] The Start:

Coming out of our last unit - Physical Computing, I got started on thinking on how I could expand my computing projects further.

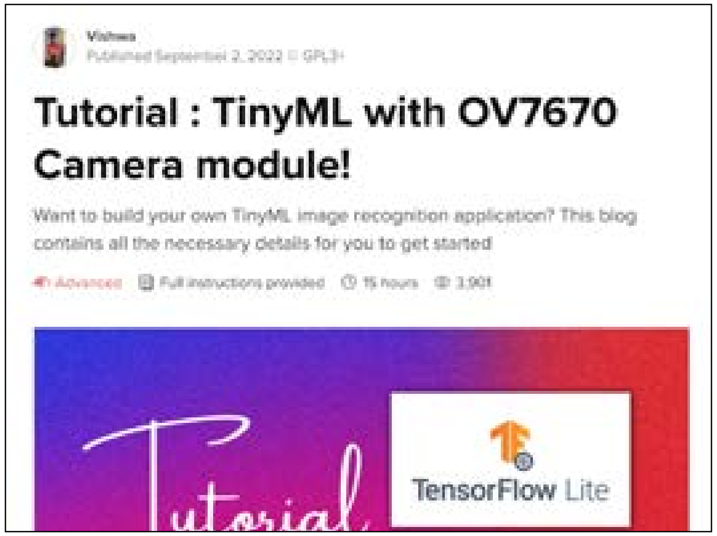
I started thinking about computational space in the frame of one's ability to make their computational ideas come to live and those who cant, and realising the importance of open sourcing code in the future of technological development.

I therefore aimed to start project that would aim to kickstart people's journey into code - with fun ideas.

[1.2 The Guide Books:

Comprehensive guidebooks will be available for any passer-by in the exhibiton and on my website. and accessible to anyone on my website
I was heavily inspired by a lot the the online tutorials I've been reading. I've sampled and adapted their templates/

Tutorial :TinyML with OV7670 Camera module
Vishwa
hackster.io



Vishwa
Published September 2, 2022 © GPL3+

Tutorial : TinyML with OV7670 Camera module!

Want to build your own TinyML image recognition application? This blog contains all the necessary details for you to get started

Advanced · Full instructions provided · 15 hours · 3,901

[2] Creation of the Physical Computing Outcomes (Outcome 1)

[2.1] The Nerf Gun Sentry Turret

I started with The Nerf Gun sentry turret by determining what all the inputs + outputs will be, and the necessary processesing inbetween:

Input

Human Targets (-> Webcam)

Processesing

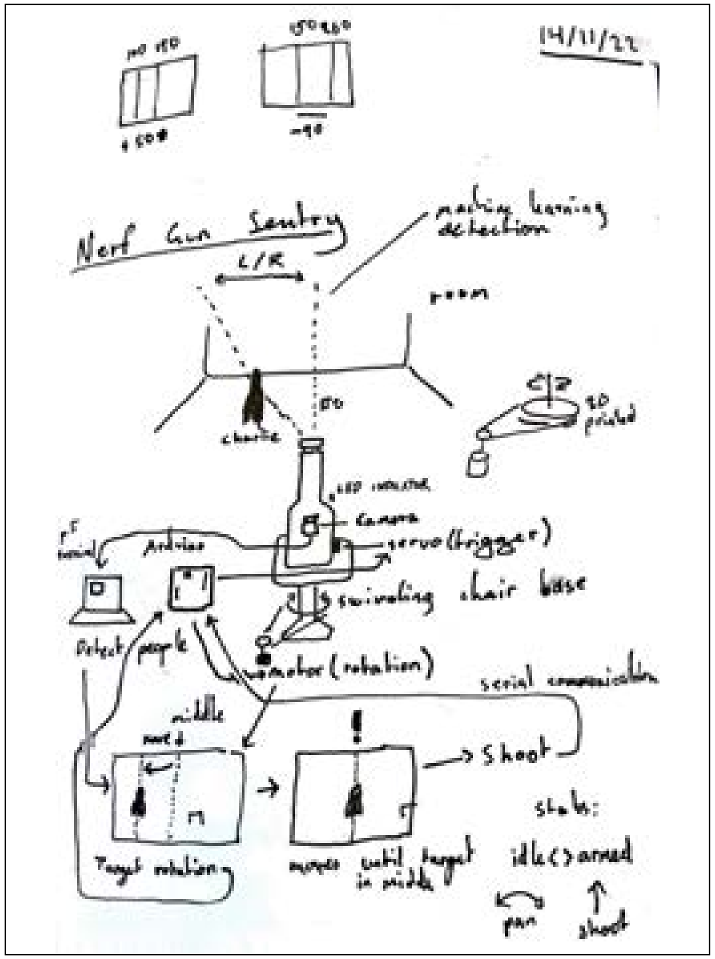
face detection terminal (coded in p5)

Output

Trigger (servos

Armed/Firing Indication (LEDs)

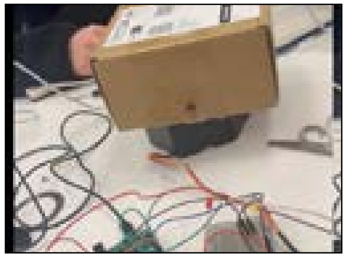
I drew up a diagram to break it all down ***(Right)***



[2.1.2] TESTING SERIAL CONNECTION TO SERVO BASE PLATE

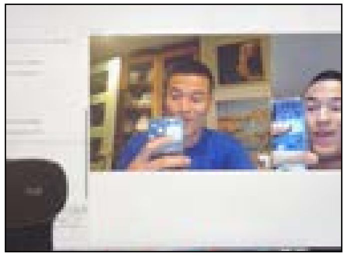
To spin the gun

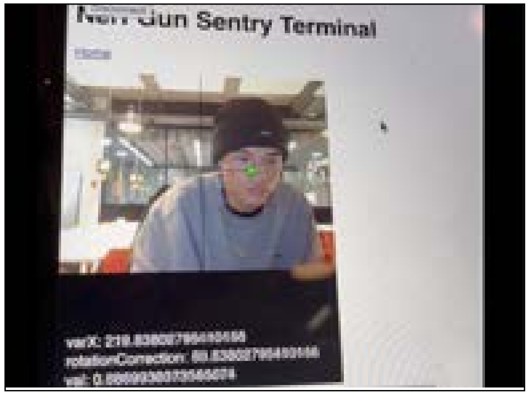
Prototype



[2.1.2] TESTING DIFFERENT WEBCAM INPUT

For barrel mounted webcam





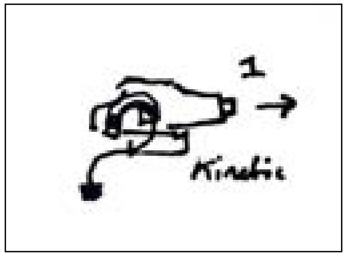
Nerf Gun Sentry Terminal

```
varX: 219.838027954[illegible]
rotationCorrection: 69.838027954[illegible]
val: 0.99999380[illegible]
```

[2.1.1] THE FIRST TRACKING TERMINAL

Tracking face position + detecting when target is in the middle

Prototype



Then I started coding the tracking terminal. I did this by adapting a program Irti gave us earlier in the year where you train + save the model right in the p5 app.

The Program detects the face, and calculates the distance between that and the middle of the screen, giving a number you'd have to spin the camera to correctly align on target. I later

Going Forward

Progression:

Face Tracking

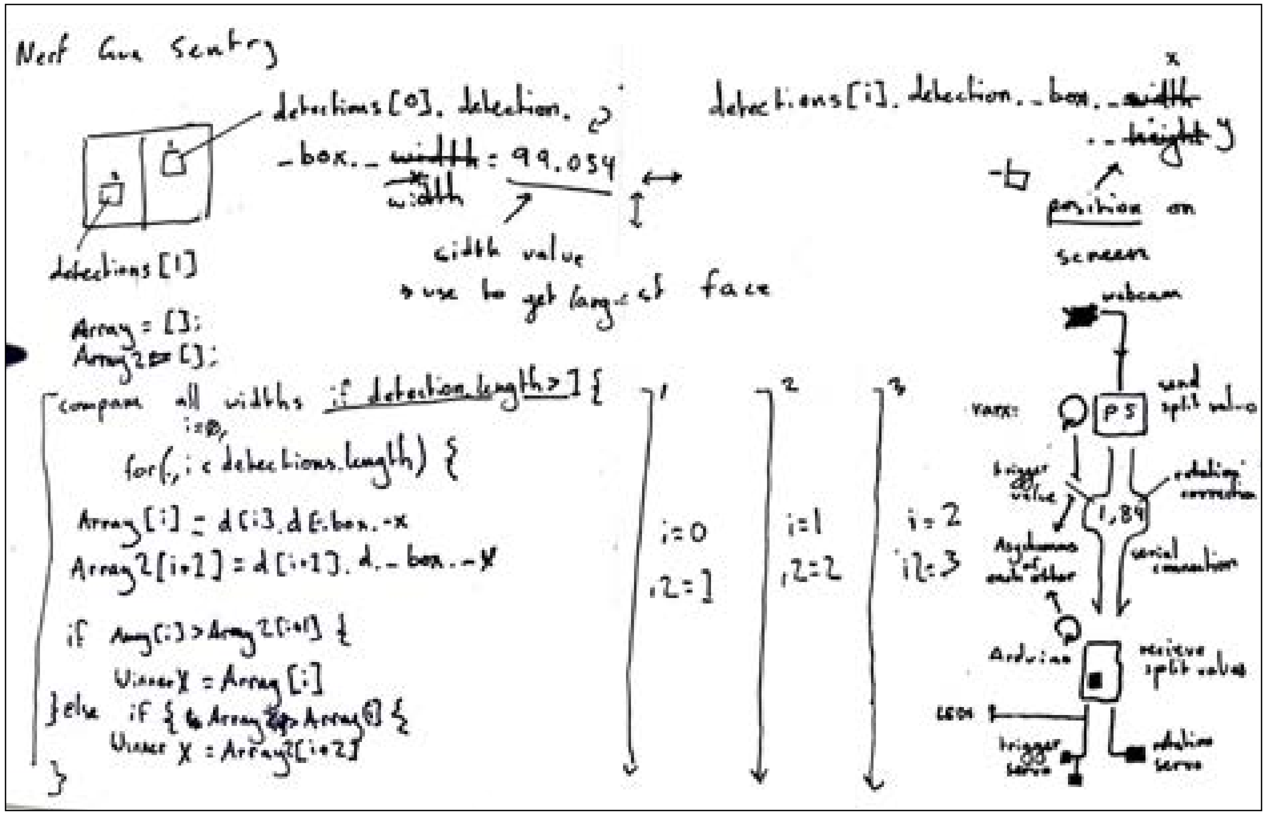
I then realised that the tracking terminal wasn’t good enough, it struggled to detect faces if the background moved and wouldn’t work if multiple faces were present.

After doing some research I came across to ml5 face detection library, which was perfect!

I worked it into my existing code and it was great! I now though had the problem of how do I get it to only target the closet target - I did this by reading the json data the ml5 library which would have the size of each box drawn around each face inside of it. and creating if statements to assign that face to varibles that the tracking terminal looks for!

Serial Connection

I also realised I had to send multiple values through serial connection. Luckily Jen gave us some code on how to do this and I incorporated it into my arduino and p5 code.



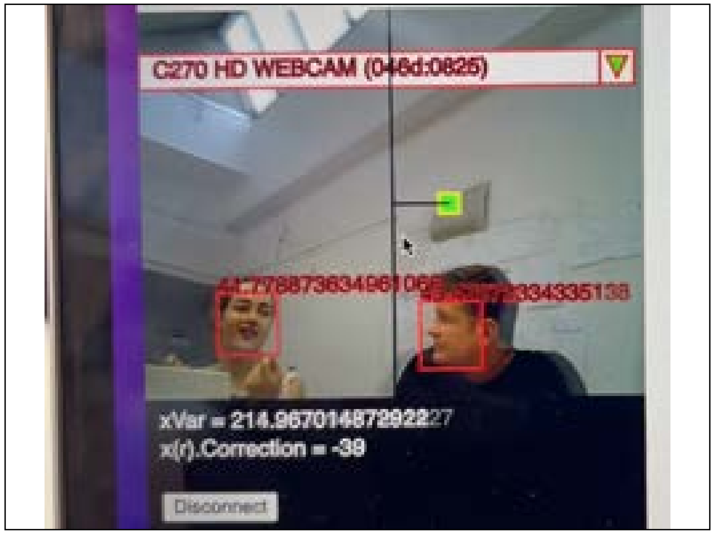
```
Array = [];
Array2 = [];
compare all widths if detection.length>1 {
    i=0;
    for(;i < detections.length) {
Array[i] = d[i].d.box._x
Array2[i+1] = d[i+1].d._box._y

if Array[i] > Array2[i+1] {
    WinnerX = Array[i]
}else if { Array < Array2 {
    Winner X = Array2[i+2]
}
```

[2.1.5] MECHANISM FOR HOW TO TARGET MULTIPLE FACES

It was a big challenge to figure out how to do thsi but I eventually got there

Notes



```
xVar = 214.96701487292227
x(r).Correction = -39
```

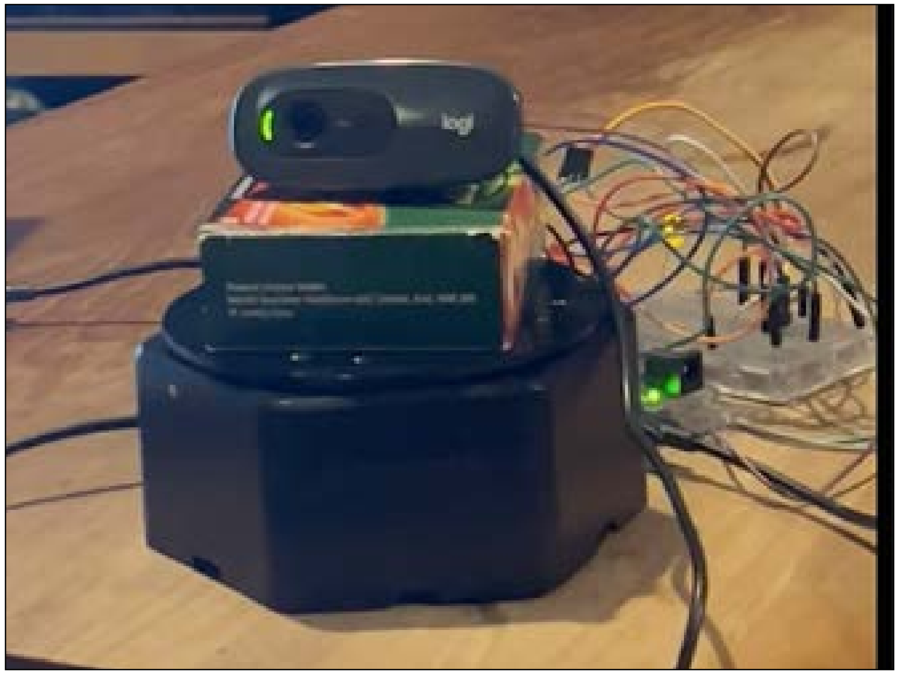
[2.1.6] THE FINAL TRACKING TERMINAL

Updated to be able to detect multiple faces + target the closest one, aswell as having a dropdown menu to select webcam

P5.js

Prototyping + Final Outcome

[2.1.7] Face Tracking Prototype:

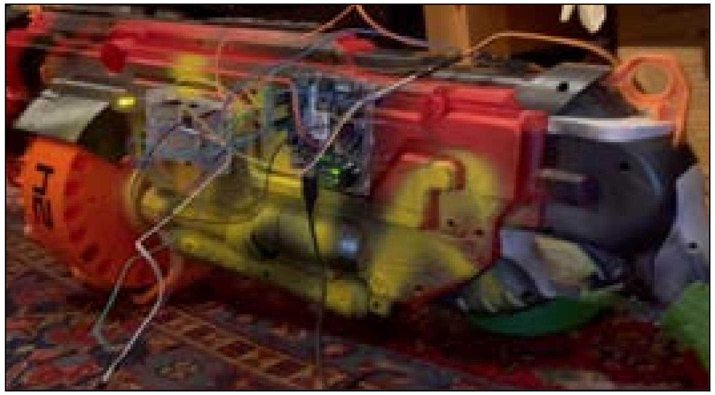


(Above) I mounted the webcam to the spinning base plate with a lensip box as a prototype while I tweaked my code to turn left/right in short intervals until it's facing straight at you

Progression:

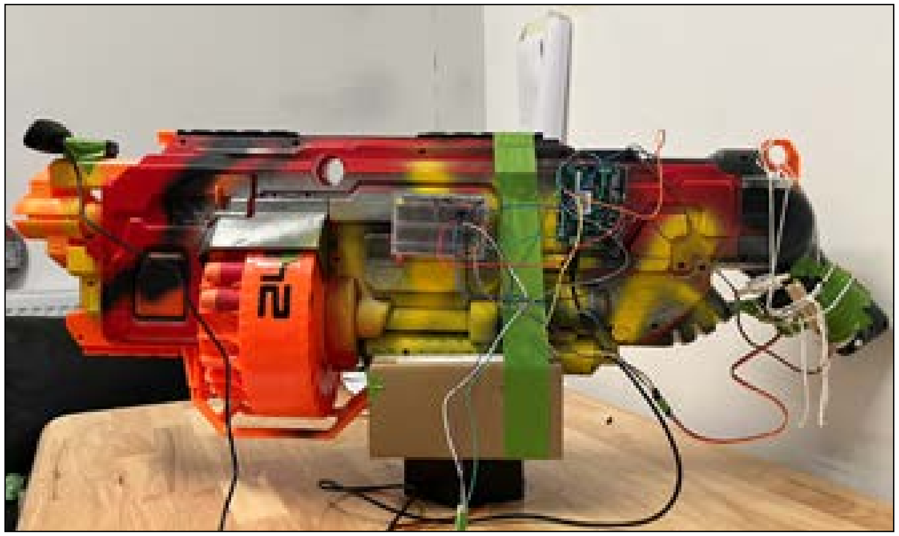
From here, I just had to mount the electronics on the gun ***(Right)*** and mount the gun onto the spinning base plate:

I also updated the p5 tracking terminal to use the ml5 face detection library so that It could detect multiple faces - I also wrote additional code to only track the closet face. I did this by detecting the largest square (around targets face).



[2.1.8] ELECTRONICS MOUNTED ON GUN
multiple faces + target the closest one
Prototype

[2.1.9] THE FINAL NERF GUN SENTRY TURRET!



[2] Creation of the Physical Computing Outcomes (Outcome 2)

[2.2] KeyScape

I started with KeyScape by understanding how to use the Leap Motion Detector with Processessing.

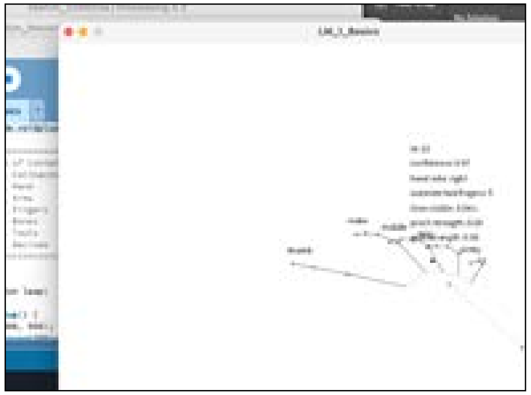
Input

Hand Gesture/Position

Processessing

Output

Sound (Chords)



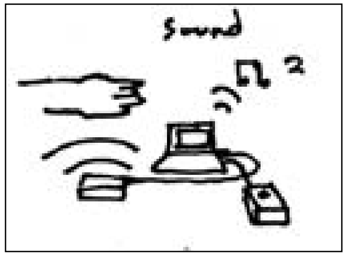
LEAP MOTION HAND POSITION EXAMPLE CODE

saves the position of the hand to variables

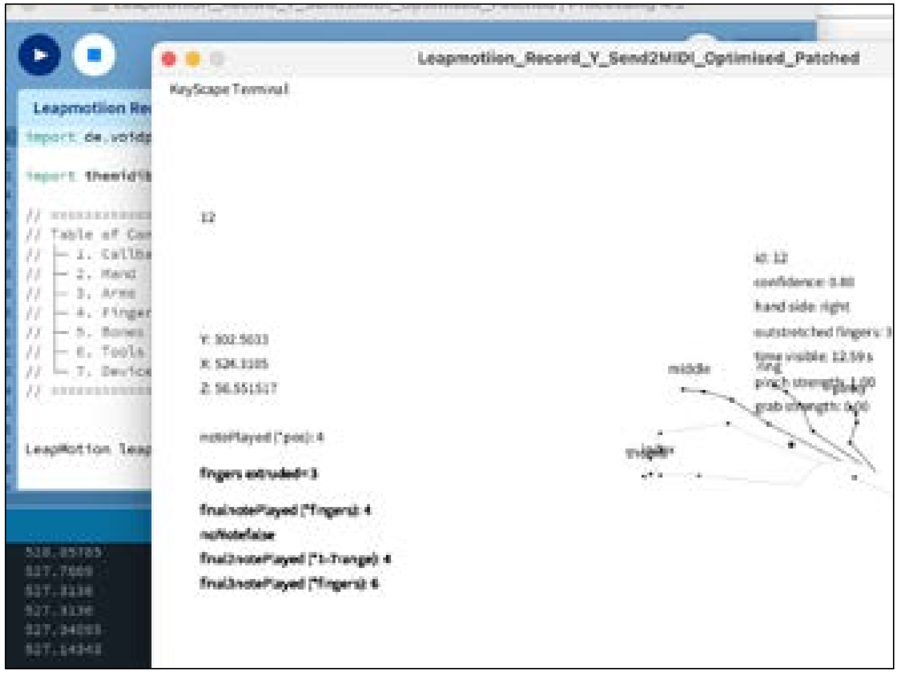
Processessing

[1] TESTING OUT THE INFRARED CAMERAS IN PROCESSESSING

I was getting the image coming through.



The Final Modified Processessing Terminal:



```
KeyScape Terminal

12

Y: 302.5033
X: 526.3105
Z: 56.551517

notePlayed (*pos): 4

fingers extruded=3

finalnotePlayed (*fingers): 4
noNotefalse
final2notePlayed (*1-Finger): 4
final3notePlayed (*fingers): 6
```

```
id: 12
confidence: 1.00
hand side: right
outstretched fingers: 3
time visible: 12.59 s
pinch strength: 1.00
grab strength: 0.00
```

(Above) I then adapted the program to output different numbers based on the hand's X-position and amount of extruded fingers (to be later translated into different chords):

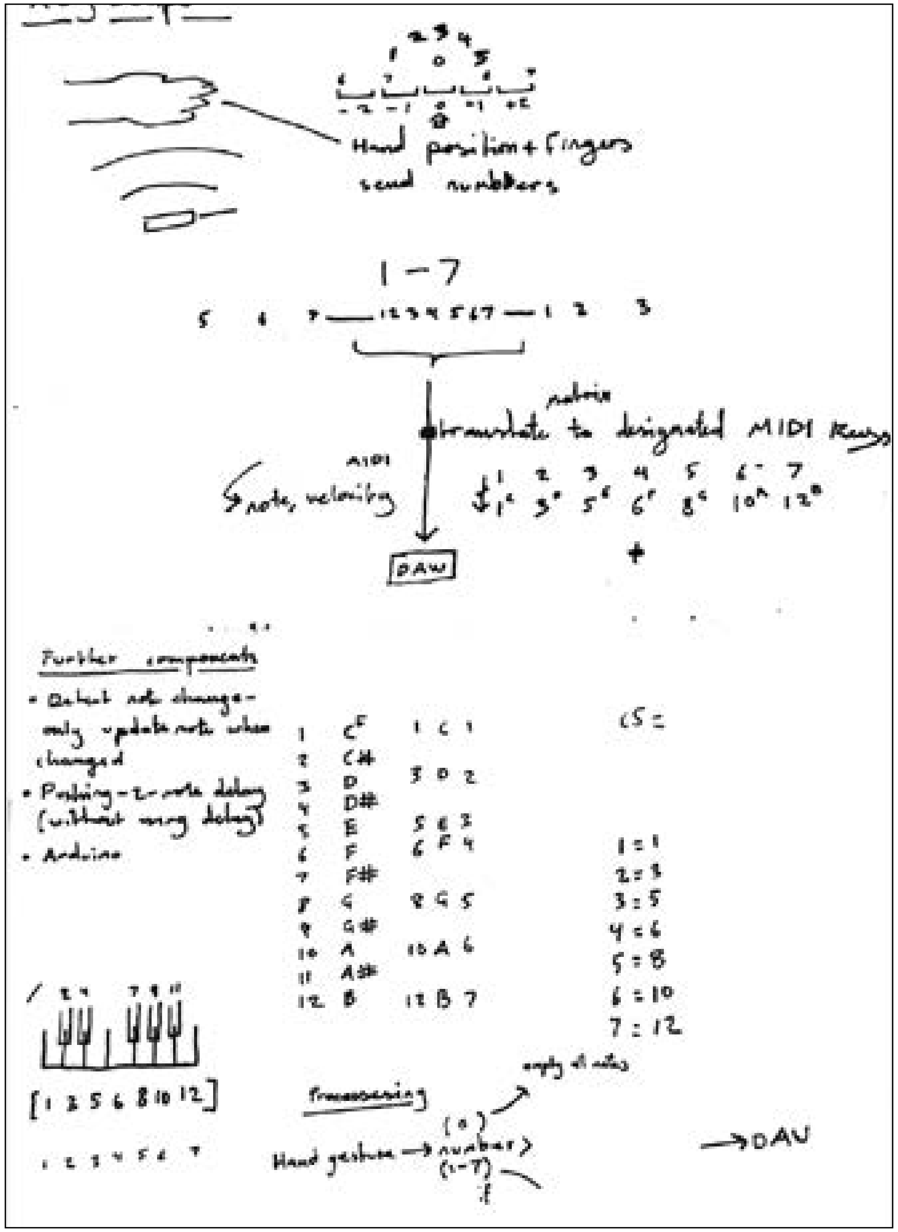
Imagine an invisible grid (on the X-Axis):

1	2	3	4	5	6	7

Where in each zone a different chord gets played, I did this because it would be easy for the user to move their hand and figure out what areas play what note. The number of extruded fingers modifies what zone you're in - I did this to allow the user to play zone one then skip to zone 3, without having to play 2 inbetween.

Going Forward

The Mechanism:



Further components

- Detect note change - only update note when changed
- Pushing - 2 - note delay (without using delay)
- Arduino

Gesture Controlled Music Is Working!

(Above) The Grove Vision AI Module Sensor features a built in a thumb-sized camera, and some customized sensors.

The company that makes them already has really well documented code on ML algorithms for people detection, as well as other customized models. So I will be able to easily to edit to detect what I want.

It also works under ultra-low power model. Which is perfect.

[2] Creation of the Physical Computing Outcomes (Outcome 3)

[2.3] Smiley Beanie:

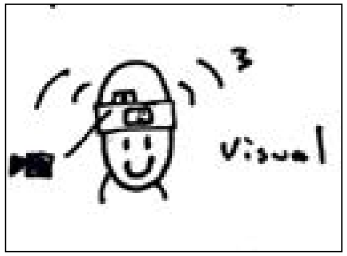
This project was by far the most advanced out of the 3, and had the most bumps and errors: here's how it went:

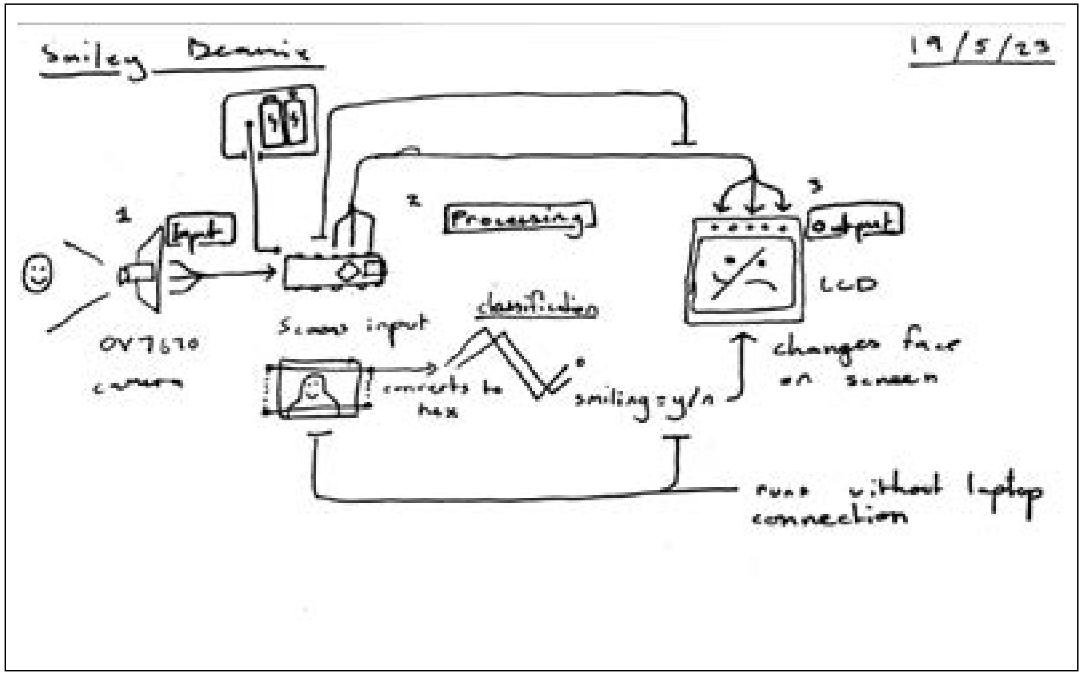
Like my other projects I started with breaking down the:

Input
	Human Emotion (-> Camera)

Processesing
	ML-Emotion
	Classification

Output
	LCD Display (Display happy/neutral face)





(Above) The aim of this project was to get all of the machine learning classification to happen on-board the Arduino chip.

That way with a battery pack the beanie could be run withough having to be cabled into a computer.

[2.3.1] BREAKING DOWN HOW IT'LL WORK

Inputs + Outputs

Notes

Initial Development

Progression:

I came across this tutorial ***(Right)*** on how use TinyML on the arduino nano 33 BLE Sense with the OV7670 Camera Module.

It looked really promising! and all the componenets are cheap - which goes with my context of all these projects being cheap & accessible.
I got going and ordered the camera

Tutorial 1:

Tutorial :TinyML with OV7670 Camera module
Vishwa
hackster.io

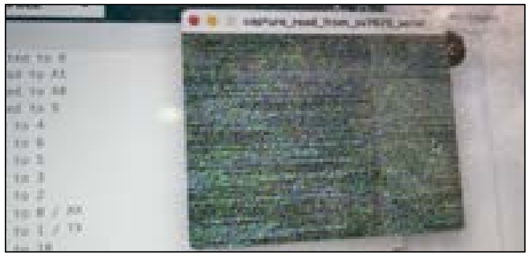
Vishwa
Published September 2, 2022 © GPL3+

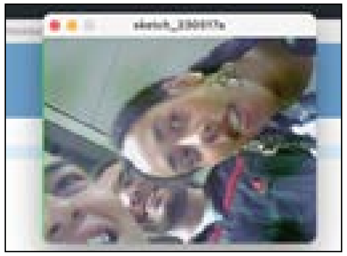
Tutorial : TinyML with OV7670 Camera module!

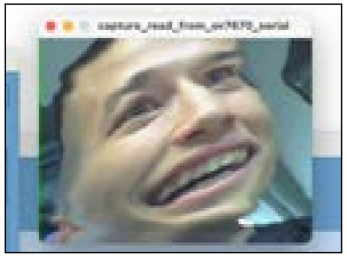
Want to build your own TinyML image recognition application? This blog contains all the necessary details for you to get started

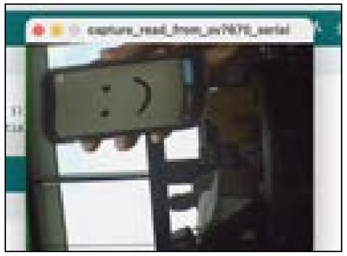
Progression:

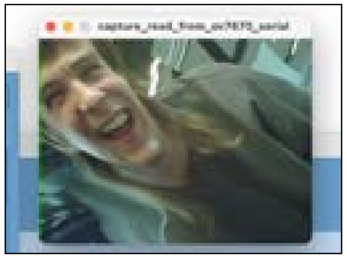
After a lot of tinkering with the cabling and following some other tutorials, I got the camera to send footage live into processesing ***(Right).***



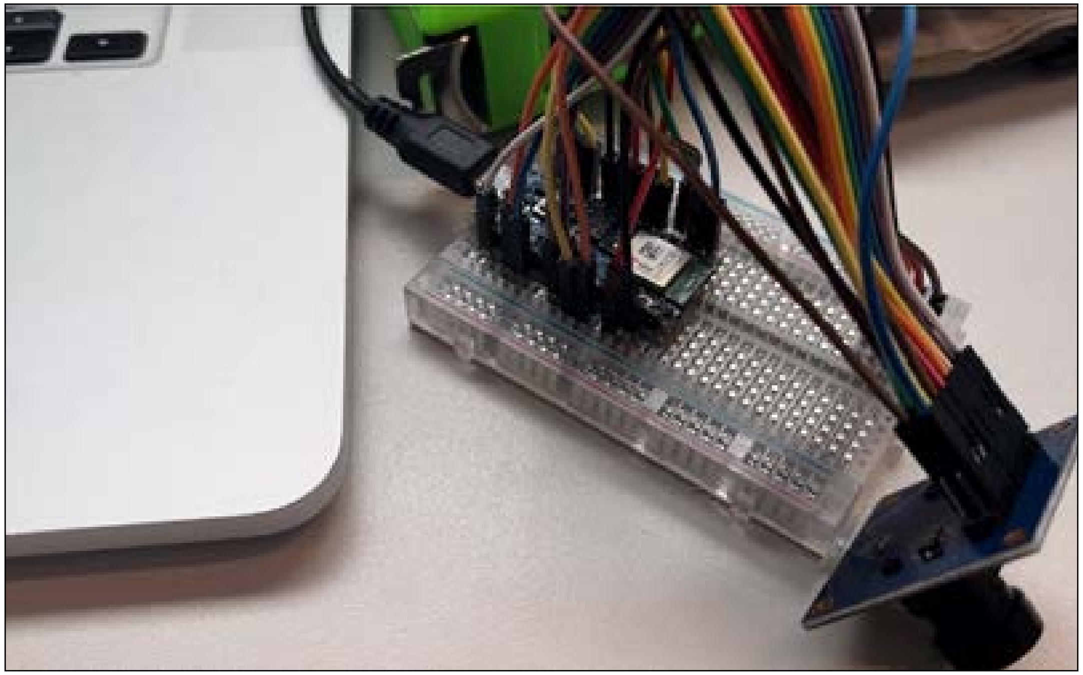








The Failed Arduino Nano 33 BLE + OV7670 Camera Module:



(Above) However I came to the realisation that with the components that I was using - the project is not feasible.

- The Nano microcontroller doesnt have enough ports to power both the camera module and the LCD Display
- The framerate of the OV7670
- ***(Left)*** The video recieved through serial would be warped if the camera was moving - which will be happening when worn on the head

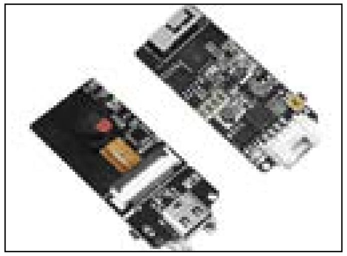
Overall I felt like I could better use my time trying to figure out another way to do this.

Going Forward

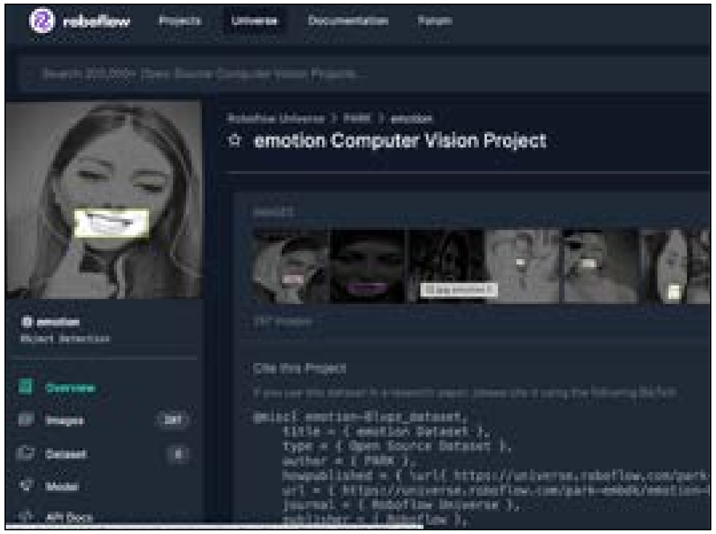
Progression:

With less time left I did some reasearch on other ways I could achieve what I want to do.

I came across this github tutorial that does object recognition with the ESP32-Cam ***(Right)***, which looked promising, but then Matt showed me the Grove Vision AI Module Sensor.



(Right) With the Grove AI Vision Module I though I could either create a dataset myself using the camera on the board, or use the datasets on roboflow



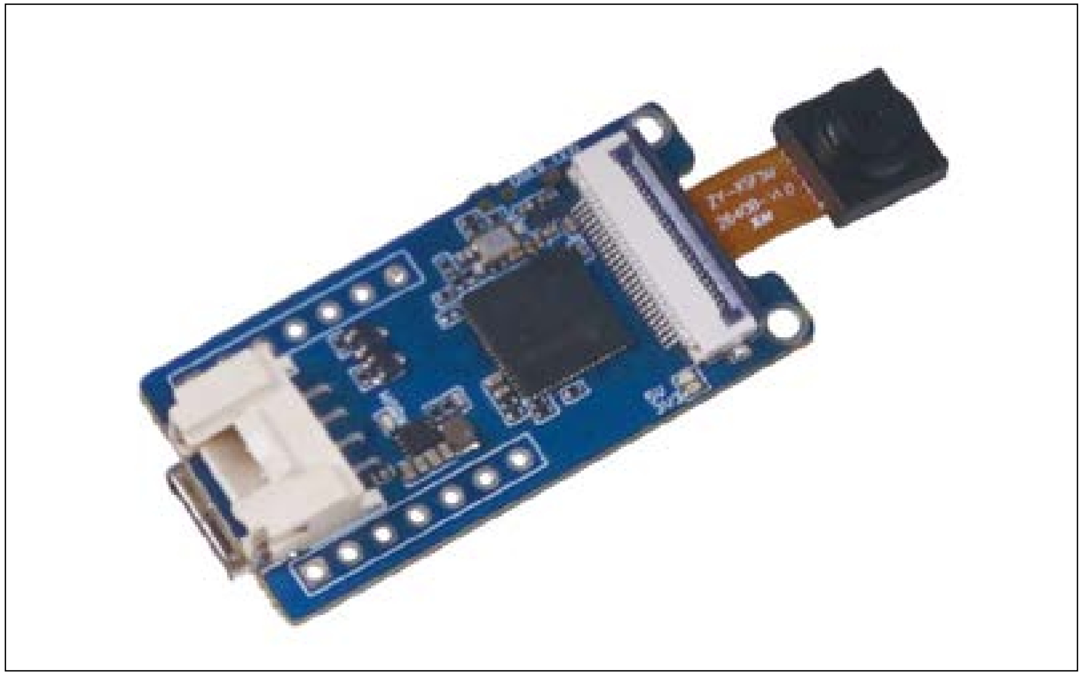
emotion Computer Vision Project

Cite this Project

If you use this dataset in a research paper, please cite it using the following BibTeX:

```
@misc{ emotion-8lypz_dataset,
    title = { emotion Dataset },
    type = { Open Source Dataset },
    author = { PARK },
    howpublished = { \url{ https://universe.roboflow.com/park-
    url = { https://universe.roboflow.com/park-mmbdk/emotion-
    journal = { Roboflow Universe },
    publisher = { Roboflow },
```

The Seeed Grove Vision AI Module Sensor:



(Above) The Grove Vision AI Module Sensor features a built in a thumb-sized camera, and some customized sensors.

The company that makes them already has documented code on ML algorithms for people detection, as well as some other customized models. It also works under ultra-low power model. So it was looking perfect...

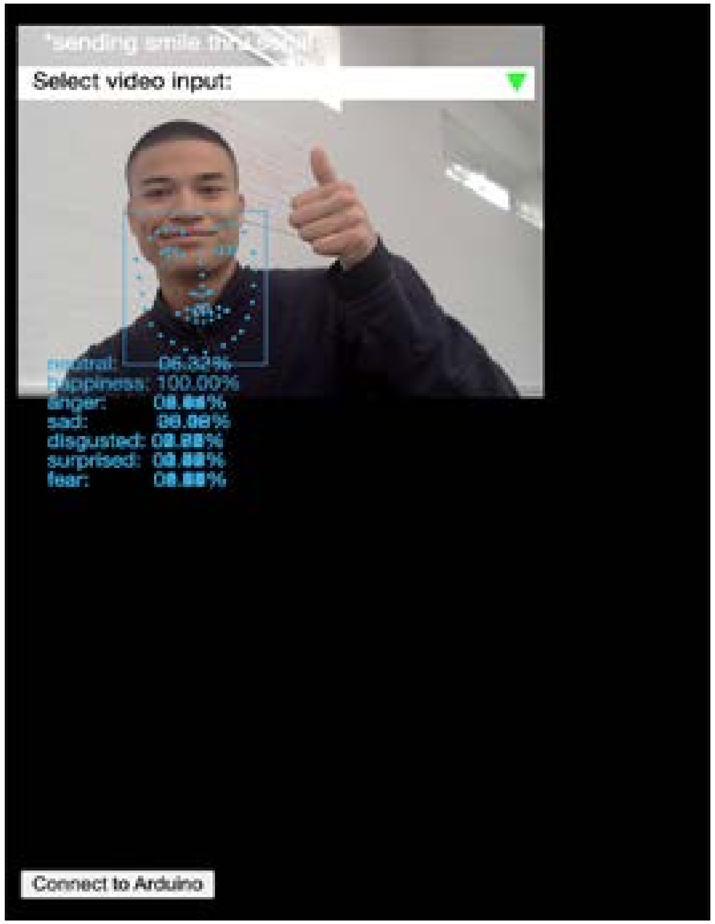
ML Training + Datasets

Progression:

Now unfortunatly after week of tinkering with the grove ai vision module, it just didnt work.

The code wasn't well documented, a lot of pages on their website weren't up anymore - so I just coudn't get it to detect human emotion and it didn't seem to even be coded in the arduino ide so i had no idea how to make do something on an LCD display

So with the 5 days I had left I decided to put together a working cabled prototype.

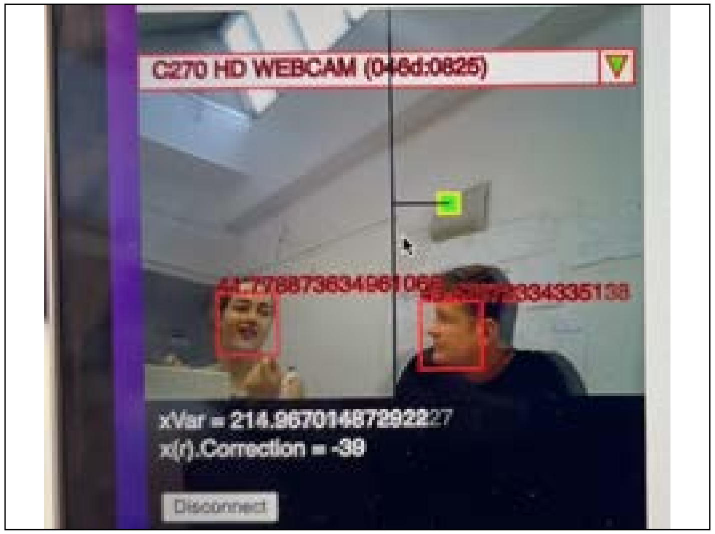


```
neutral:    06.32%
happiness:  100.00%
anger:      00.00%
sad:        00.00%
disgusted:  00.00%
surprised:  00.00%
fear:       00.00%
```

[2.1.6] THE FINAL EMOTION DETECTION TERMINAL

If Happiness is detected to 50%, it'll send a signal through serial connection

P5.js



```
41.778873634961066
xVar = 214.967014872922227
x(r).Correction = -39
```

[2.1.6] THE NERF GUN SENTRY TERIMNAL

I apadted my old termianl with open ml5 code from a youtube tutorial

P5.js

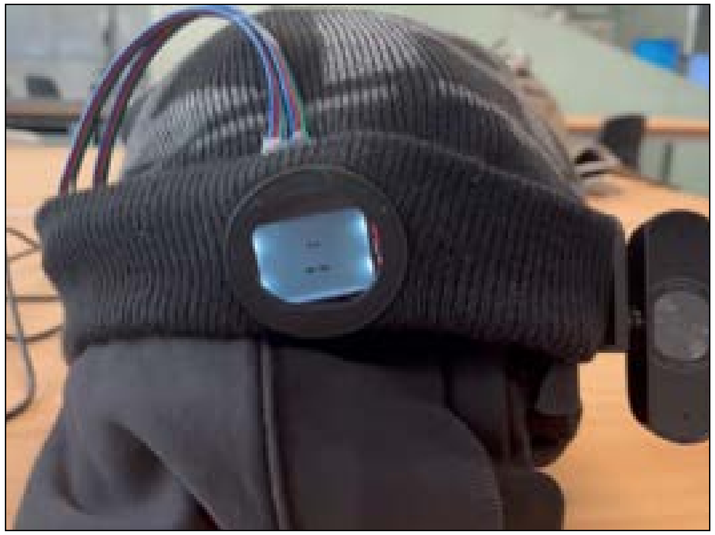
Going Forward

Progression:

I did some reasearch on other ways I could achieve what I want to do.

I came across this github tutorial that does object recognition with the ESP32-Cam ***(Right)***, which looked promising, but then Matt showed me the Grove Vision AI Module Sensor.

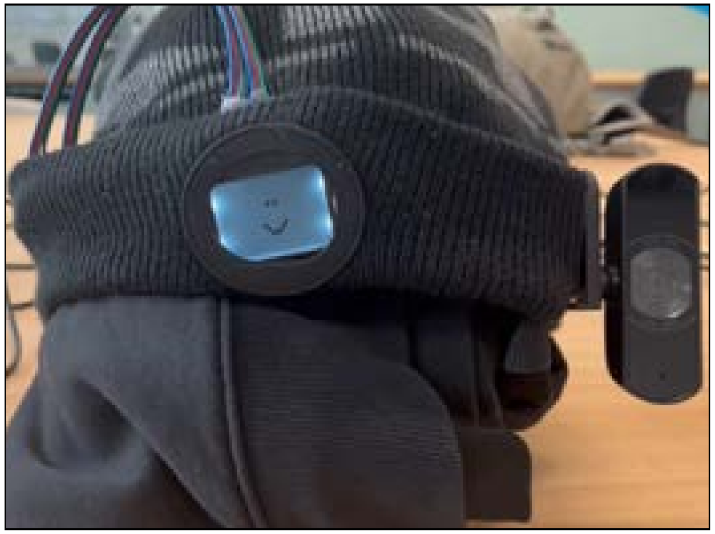
The Final Prototype:



[2.1.6] LCD DISPLAYING NEUTRAL FACE

If no smile detected

Arduino + Nokia 5110 LCD



[2.1.6] LCD DISPLAYING HAPPY FACE

If smile is detected

Arduino + Nokia 5110 LCD

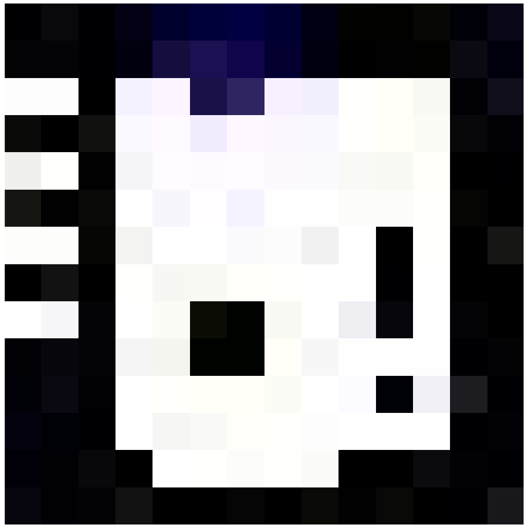
(Above) The original aim of the beanie was that has an in-built camera, and an LCD display, which smiles back at you when it detects a smile (using ML classification) It will be able to run independently without a laptop.

I will continue to work on getting closer to there

[3] Bringing The Project Back Together

[3.1] Project Branding

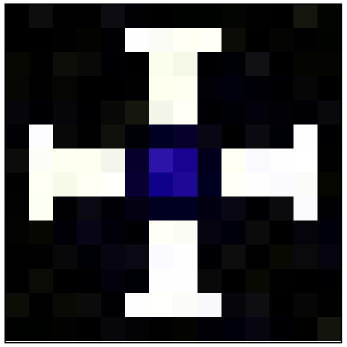
I wanted there to be a visual theme across all the 3 outcomes. So I design a modular branding system suing a grid and pixel art.



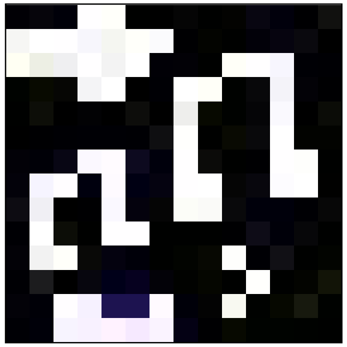
THE MAIN LOGO
Pixel Rendition of an Arduino Board

System:
All are on a 14x14 grid

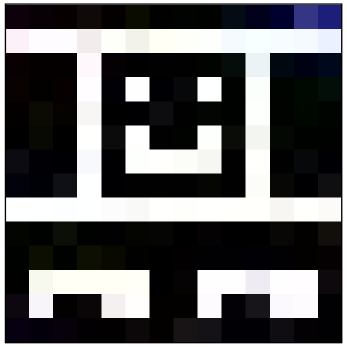
Colours:
Black + White
2 Blue Pixels for highlights



THE NERF GUN SENTRY LOGO
Target crosshairs



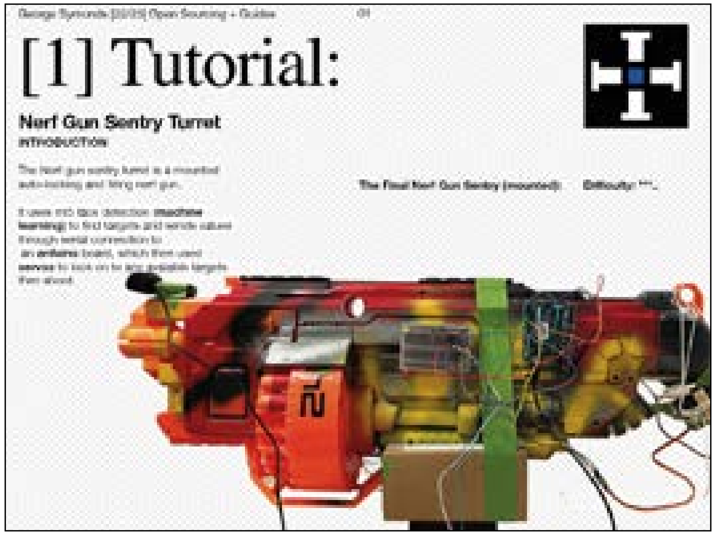
KEYSCAPE
A Waving hand creating musical notes



SMILEYBEANIE
Beanie on head (+LCD display and camera)

[3.3] The GuideBooks

After reflecting on where people would find my tutorials, I decided that the format will also have to look good and function when read on a webpage.


George Symonds [22/23] Open Sourcing + Guides 01

[1] Tutorial:

Nerf Gun Sentry Turret

INTRODUCTION

The Nerf gun sentry turret is a mounted auto-locking and firing nerf gun.

It uses [illegible] detection **machine learning** to find targets and sends values through serial connection to an **arduino** board, which then used **servos** to lock on to any available targets then shoot.

The Final Nerf Gun Sentry (mounted): Difficulty: **-.

THE NERF GUN SENTRY TURRET
Tutorial Guidebook

George Symonds [22/23] 01

[2] Tutorial:

KeyScape: Gesture Controlled Music Creation

INTRODUCTION

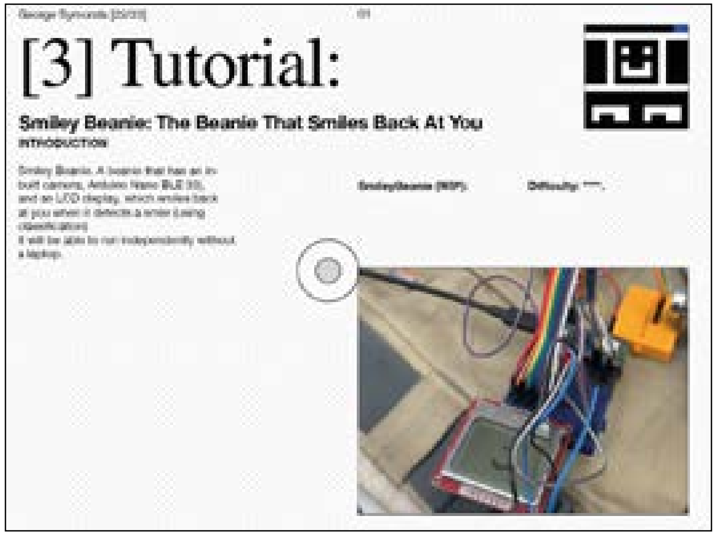
A modified leap motion detector that reads the users' gesture (hand x-position + number of extended fingers) and plays different chords in the chosen scale.

Made in processing. [illegible]

[illegible]

KeyScape: Difficulty: **-.

KEYSCAPE
Tutorial Guidebook


George Symonds [22/23] 01

[3] Tutorial:

Smiley Beanie: The Beanie That Smiles Back At You

INTRODUCTION

Smiley Beanie. A beanie that has an in-built camera, Arduino Nano BLE 33, and an LCD display, which smiles back at you when it detects a smile (using classification)
It will be able to run independently without a laptop.

SmileyBeanie (WIP): Difficulty: ***.

SMILEYBEANIE
Tutorial Guidebook

I then started on getting to work designing all the neccesary accompanying guidebooks for each project.

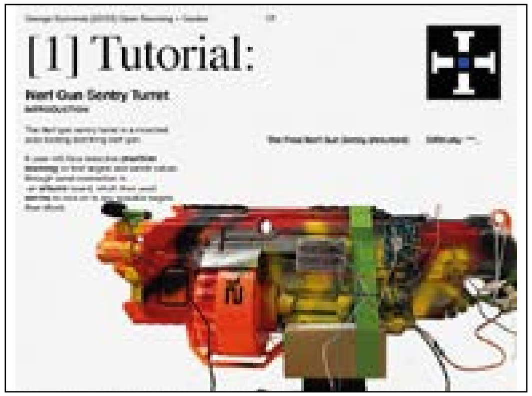
The aim for the guidebooks are to be accessible to everyone so they will be free, and availble and free publications or on my website.

For it to be accessible they have to be able to be understood by someone who doesn't have any prior knowledge of coding, so they have to go in-depth on every part of the project.

Neccessary Components:
Hardware + Software needed to start creating (+Cost of project)
Clear + Concise Instructions
Breaking down all softwares and how to use them
  Explaining how p5.js/processesing works
    the files (index.html, sketch.js, style.css) and how they interact with each other
  Explaining The Arduino IDE
  How to install libraries + how to use them
Wiring Breakdown Schematics + Pin in instructions
  Troubleshooting
Extra relevent references

Open sourcing + Guides:

Guidebook Pages



[1] Tutorial:

Nerf Gun Sentry Turret

GUIDE-BOOK 1

Full guide available here

[2] Tutorial:

KeyScape: Gesture Controlled Music Creation

INTRODUCTION

GUIDE-BOOK 2

Full guide available here

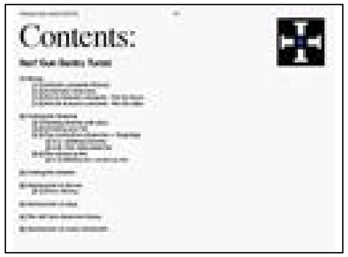
[3] Tutorial:

Smiley Beanie: The Beanie That Smiles Back At You

GUIDE-BOOK 3

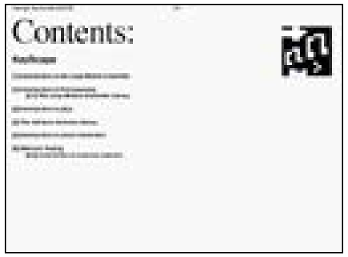
Full guide available here

Whats Needed:



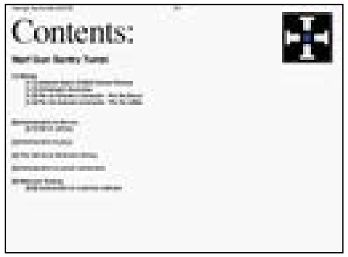
Contents:

Whats Needed:



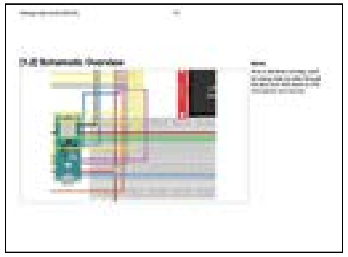
Contents:

Whats Needed:

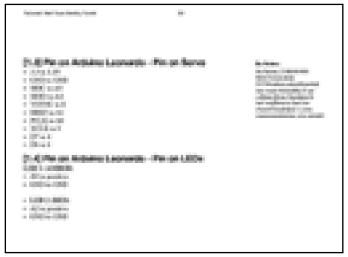


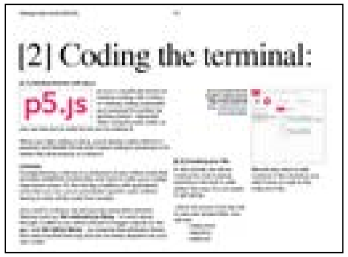
Contents:

[1] Wiring:



1. Wiring:





[2] Coding the terminal:

p5.js

The Vision:

The main key of this project is that it encourages other pople to create.

Inside the tutorials are links to all the code needed to

The Code:

https://github.com/gsymonds1/ComputationalEnvironmentsProjects/tree/main

[All ANNOTED CODE FOR 3 PROJECTS HERE](https://github.com/gsymonds1/ComputationalEnvironmentsProjects/tree/main)

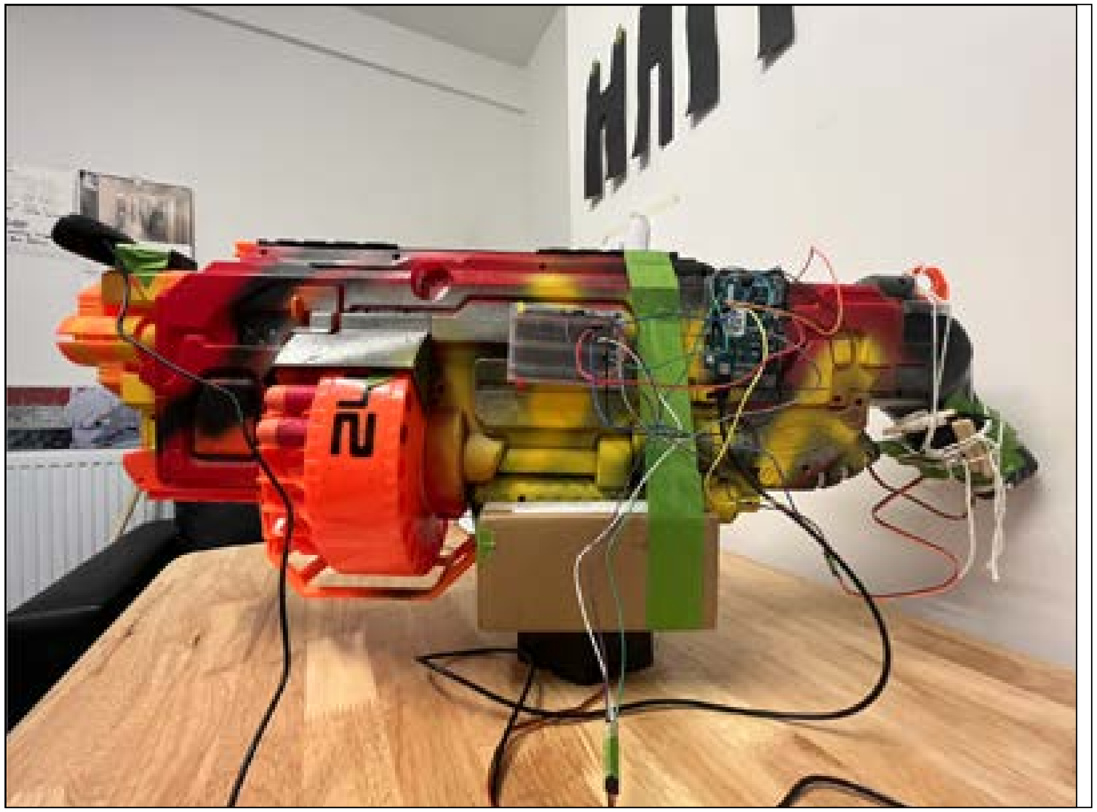
P5.js

Processsessing

Arduino (C*):

Final Outcomes:

Working Physical Outcomes In Action

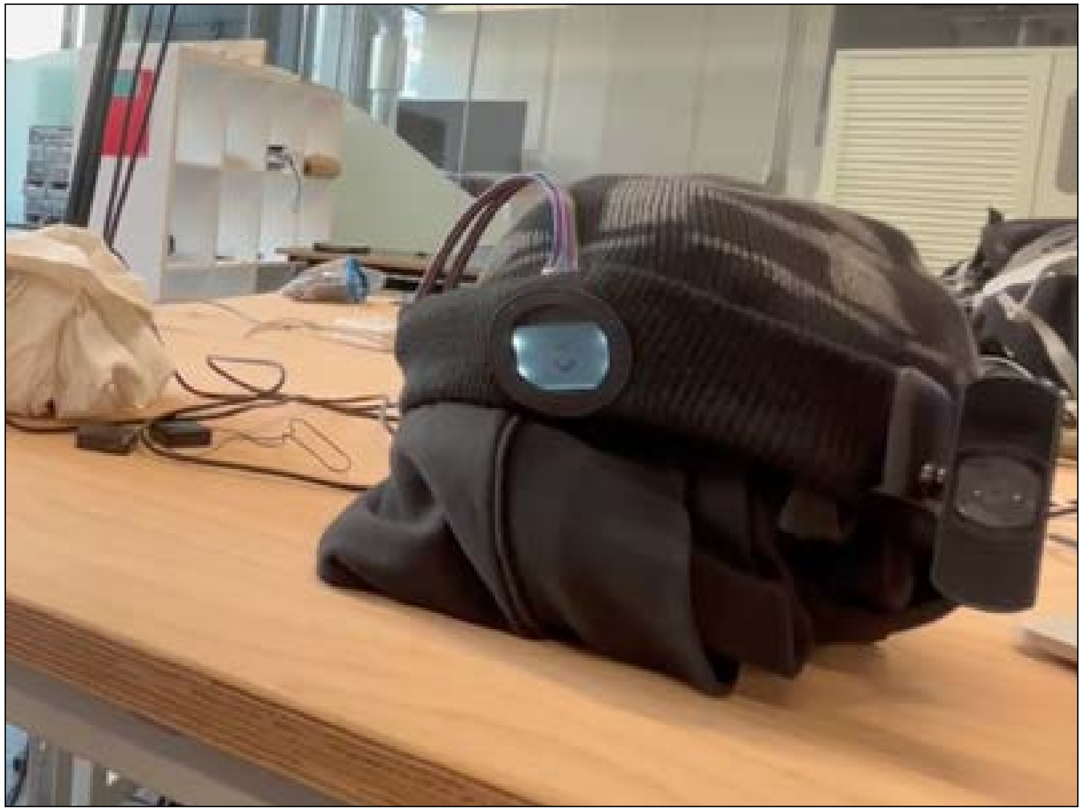


NERF FUN SENTRY TURRET

ML Powered home defense system

KEYSCAPE

Gesture Controlled Music Creation



SMILEY BEANIE
The Beanie that smiles back at you

The final working interactions :)

Watch Here
https://www.youtube.com/watch?v=H6V_2QWeIMw

...and...

You Can Play The Terminal Yourself Here:
Nerf Gun Sentry
https://gsymonds1.github.io/pages/NERFGUNSENTRY/movementDetection/index_V2.html

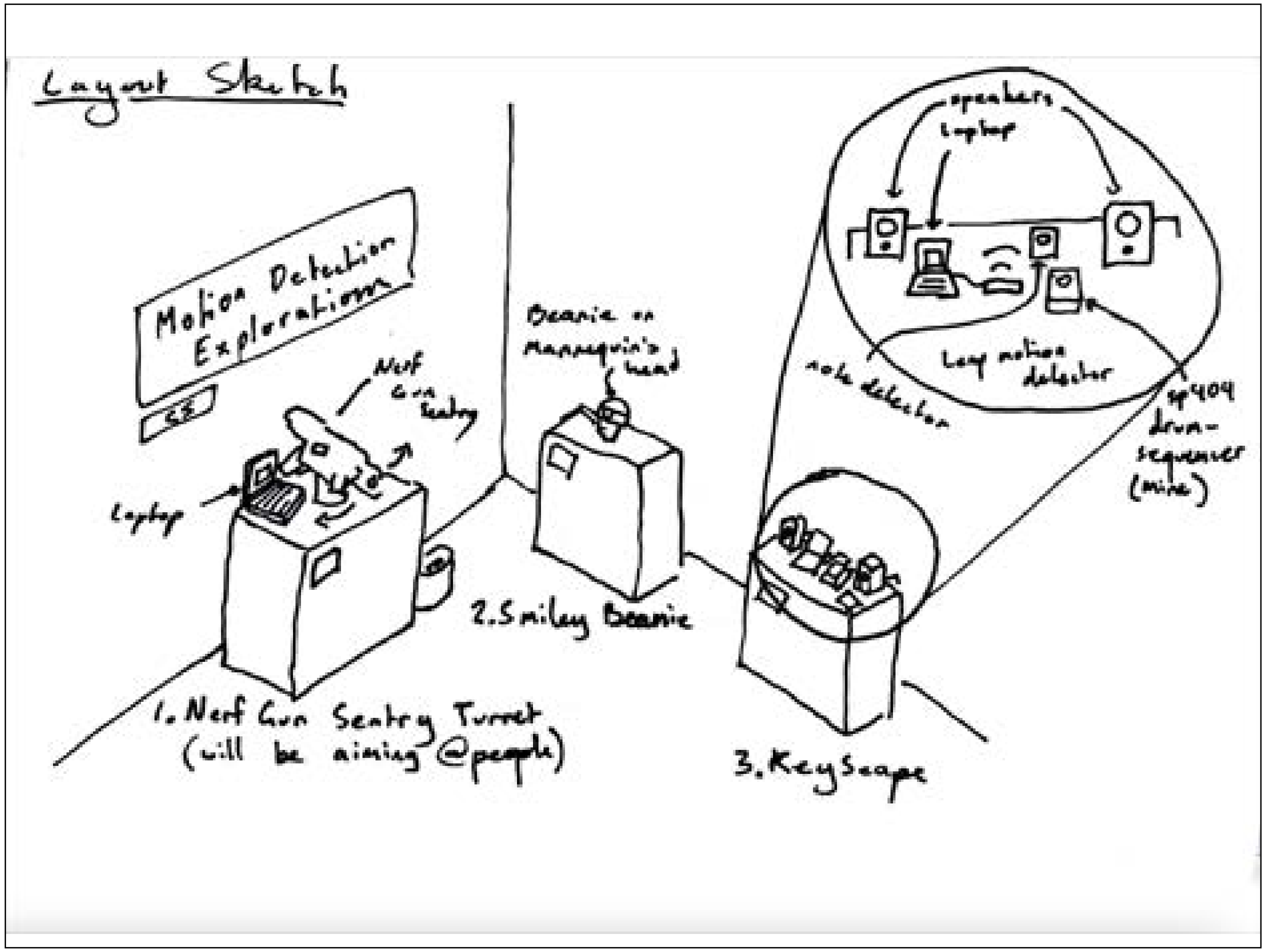
Smiley Beanie
https://gsymonds1.github.io/pages/NERFGUNSENTRY/movementDetection/index_V2.html

In Exhibition Space:

In the exhibition:

In the exhibition the three projects will be presented next to each other showcasing them in action.

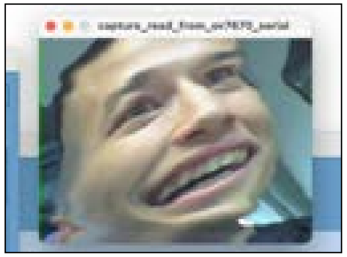
Next to each stand will be a discription of the project and guidebooks that anyone will be able to take home to try out themselves


Layout Sketch
Motion Detection Explorations
Nerf Gun Sentry
Laptop
Beanie on Mannequin's head
speakers
laptop
Leap motion detector
note detector
sp404 drum-sequencer (mine)
1. Nerf Gun Sentry Turret (will be aiming @people)
2. Smiley Beanie
3. KeyScape

Evaluation + Reflection:

The Present:

In this critical reflection, I will explore my physical computing project, examining what I achieved, how I reacted to challenges along the way, and propose ideas for future developments.



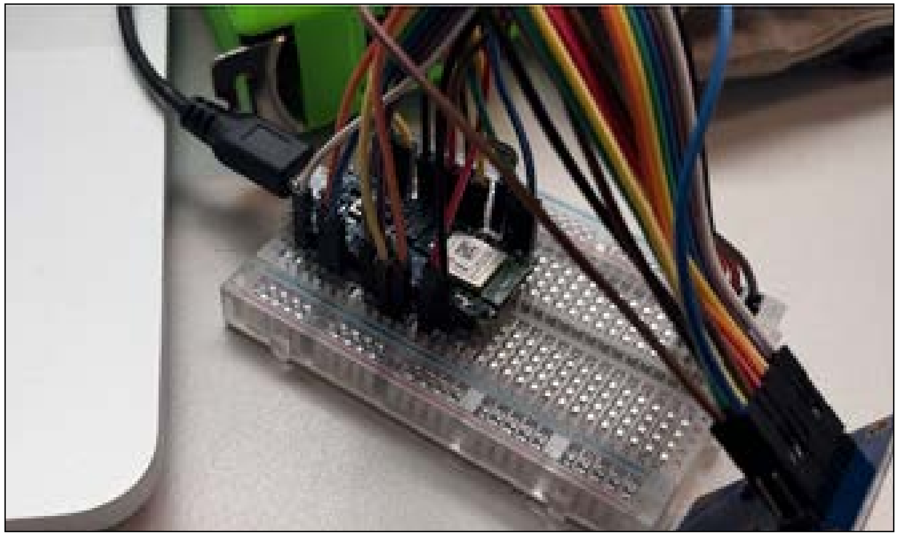
I am happy with what I have accomplished, aswell as achieving almost everything I wanted to do in my aims. This project has allowed me to learn a wide range of new skills through having to create 3 different physical outcomes.

I futhfilled my aims in creating 3 fun projects that would engage someone who would like to get into coding, as proved by feedback I'd get from other students and friends when I tell thema bout what I'm doing

I also feel like the theme of education is my project is positive.

Throughout this project I'd find myself fustrated with getting stuck on certain problems - even problems that I initially thought wouldn't be difficult, such as getting Nerf Gun Sentry to detect multiple faces and only target the closet one. But I found taking a break and step back, drawing out mechanisms on paper with a fresh mind would always help me find the solution.

The Failed Arduino Nano 33 BLE + OV7670 Camera Module:



[2] Challenges Face: The SmileyBeanie

From the start I knew the SmileyBeanie project would be ambitious, but I was keen to learn more about arduino along the way, which I did and I'm happy with that.

I didn't however predict that it would be too much andit's unfortunate that I wasn't able to make a working uncabled version for this hand-in,

However deciding to make a prototype instead with the amount of time i had left was the best decision I made so I could still show proof of concept rather than nothing.

Going Forward

Next Steps and Future Developments:

The next step is the final degree show, for it I aim to further improve and optimise my projectsm making more effective, user friendly, and stable, aswell as further improving the guidebooks

Nerf Gun Sentry Turret

3D brakets to attach the servos to the gun with more grip, so it more reliable pulls the trigger and doesnt slip

KeyScape

Add quantisation, an inbuilt delay loop where every cycle it takes where the hand position is and sends out the coresponding midi notes instead of updating everytime the hand moves (sometimes glitchy and plays 2 notes at the same time)

SmileyBeanie

Get it working uncabled, It looks like it will be possible using the ESP-32 CAM module which I will order and develop.

In Conclusion

I'm happy with my final hand-in for the year, it concludes and contains everything I've learnt in the course and has given a great starty into coding which I will be continuing to develop in my practise next year going back into Graphic Design and future careers.

Bibliography:

Thank you for reading!
This Project was fun :)

George Symonds

https://github.com/plusea

https://www.designoftheworld.com/piano-stairs/