Goldsmiths, University of London

Mr. Hussein Farkhani 33425905 BSc Computer Science 18/02/2019

A novel approach to building a refreshable braille reader

Introduction

What I aim to demonstrate is a more efficient design of a refreshable braille reader, current designs of braille readers are plagued by the paradigm of building one cell and then scaling up, which has resulted in excessive financial and engineering cost of braille readers. It is far more efficient instead to take a divide and conquer approach.

Most electronic braille readers range from £800 to over £3000+, which is out of the price range of the majority of blind people, an estimated 36 million people in 2015. There was a great attempt to make an affordable and accessible refreshable braille display see (http://blitab.com/ approx. cost £300 - 400) however the technology is patented, there is no information on how it works and moreover there has been no meaningful updates in last 6 months which indicates as with other braille reader attempts that they had problems securing funding.

Unfortunately, high cost designs require a lot of substantial capital for the product to become a reality and enter the market.

Why do we need refreshable braille readers? the chicken and the egg problem...

A typical braille book needs to be at least 3 times larger than a conventional book, when you read braille you need space to brush your fingers over the braille dots, the spacing between dots is required to distinguish letters, numbers and punctuations (there is even braille music notation)

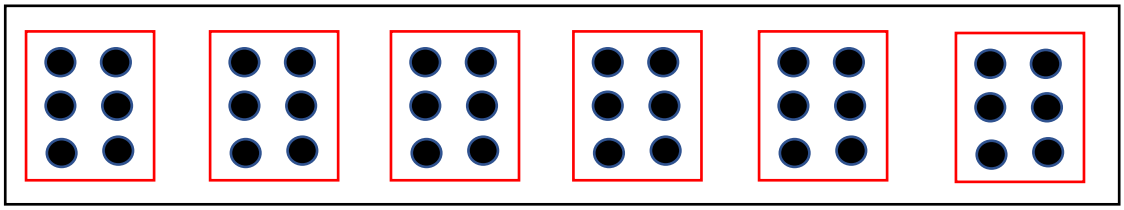
As you can imagine, a normal novel of 300 pages converted into braille would take at least 6 to 10 large volumes and be non-portable (carried in cardboard boxes). Because of the large physical size, braille books are expensive to print and transport, because they are expensive, less blind people can access braille books as well as less blind people can learn braille so therefore there is less books and higher costs - the chicken and the egg problem, a vicious cycle.

With a refreshable braille display, you can read as many books, including books that do not have a braille format, all with just one device which is portable and only needs to be built once.

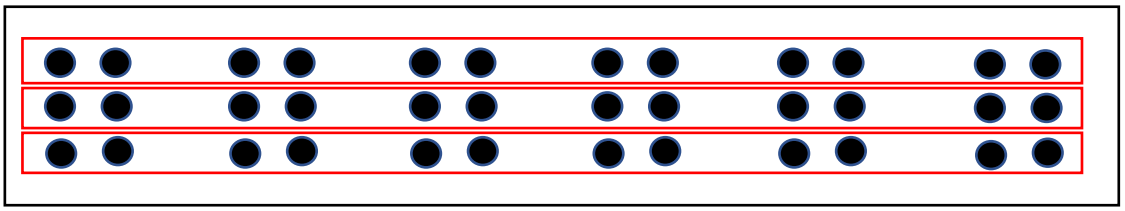
I intend to create numerous realistic designs, Proof of Concepts (PoC) of a new approach to the design of braille readers. Nearly all braille readers that have been made/designed have all focused on building one braille cell and then scaling this up to form a line of cells (mimicking a sentence), most designers seem to follow this pattern. The problem with this is that the cost and the electrical/power requirements increase drastically from one cell to

twelve or more cells. What I intend to demonstrate is that is far more efficient to carry out a divide and conquer strategy.

Normal cell being scaled up, this the current design paradigm of braille readers on the market today.



My PoC dividing a line of braille cells into three parts, a divide and conquer approach.



I am hoping this PoC can influence new and more efficient attempts at making a refreshable braille display, which is desperately needed.

[Update] I have already created the 3D models, what I discovered through building the models, is that a refined finished product is not possible at the end of this project and that it is more effective to build a tool. A tool similar to TinkerCad but with actual physical constraints considered.

Aims and Objectives (Updated)

- What is the minimum required number of resources to build a single line of a braille cell?
- Are there alternatives, especially DIY, into building a braille display with minimum use of microelectronics?
- Create 3D models of all of the PoC with animations showing how they work, this is essential in order to communicate the concepts efficiently. I understand my own drawings, but I think others will not be able to, that is why I need to create 3D models.

- All of the models are scaled to real life proportions, so they are ready for 3D printing or dimension copying.
- Build a small physical PoC based of the 3D models to showcase the designs.

My deliverables will consist of a whole set of animated and non-animated 3D models showcasing DIY efficient braille reader designs which have correct dimensions and a physical proof of concept (PoC) showing the reality of one of the designs. I intend to use the Blender software to model and animate the 3D models. [Update] the deliverables will now consist of a Unity program similar to TinkerCad/MecaBricks.

I have already achieved most of these objectives, I now need to advance the project further. What I have discovered is that it is far more effective in terms of social impact to build a tool rather than then build a refined finished product (which is unlikely due to resource constraints)

What I find with other 3D modelling packages such as TinkerCad, MecaBricks, Blender etc... is that they do not consider the constraints of each part, but they allow modelling a concept to be fast and efficient;

Constraints for physical parts include:

- Weight
- Durability (cardboard is less durable than MDF wood, for example)
- Output power (if electrical)
- Output force (if mechanical)

What I am going to build is something similar to TinkerCad and MecaBricks, a 3D modelling package that is simple to use, but each part has a constraint. A sum of constraints (called cost) for any given solution is displayed as a number. The aim of the user is to build a solution to braille reader with the lowest cost. The user does not need to be concerned with technical aspects of the parts, they focus on finding the most efficient solution.

The user can play around with components such as the Arduino, pieces of cardboard, DC motors just like Lego. Reducing the cost of experimentation – the key reason why Lego was so successful.

By removing technical barriers to starting, more creative and novel solutions can be found to a low-cost braille reader. This project will be built using Unity3D which I have experience with.

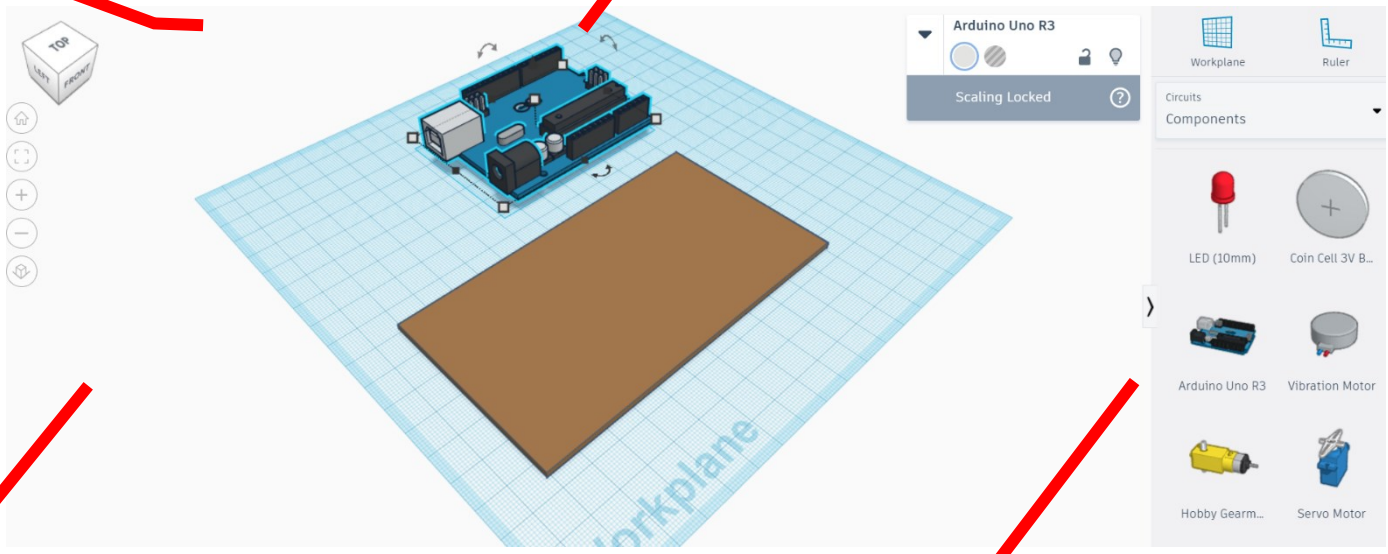
New aims and objectives:

- To build a tool that allows a user with no or low technical knowledge to model a braille reader
- To provide components necessary in the program to build a low-cost braille reader
- A simple UI that allows a user to start quickly

- Each component must have constraints communicated to the user in a non-technical way

Total sum of constraints would appear here, a total score. Aim is to reduce the score to the lowest while finding the solution to the braille reader problem

User can move, rotate and connect components together



Selected component constraint would appear here, each components constraint adds to the to the total score

List of real-life components, each with their constraints

Methods (old)

- Learn 3D modelling with blender (there is currently a new major update to blender, which will speed up workflow)
- I already have drawings of multiple PoCs, I just need to model them.
- Evaluate the models according to the Braille Reader Evaluation Criteria.
- After evaluation stage, animate the models, this is to show how PoC would work, this is the most efficient way of communicating the concept.
- Select one PoC design and build a physical demonstration.
- All designs must follow the braille reader evaluation criteria.

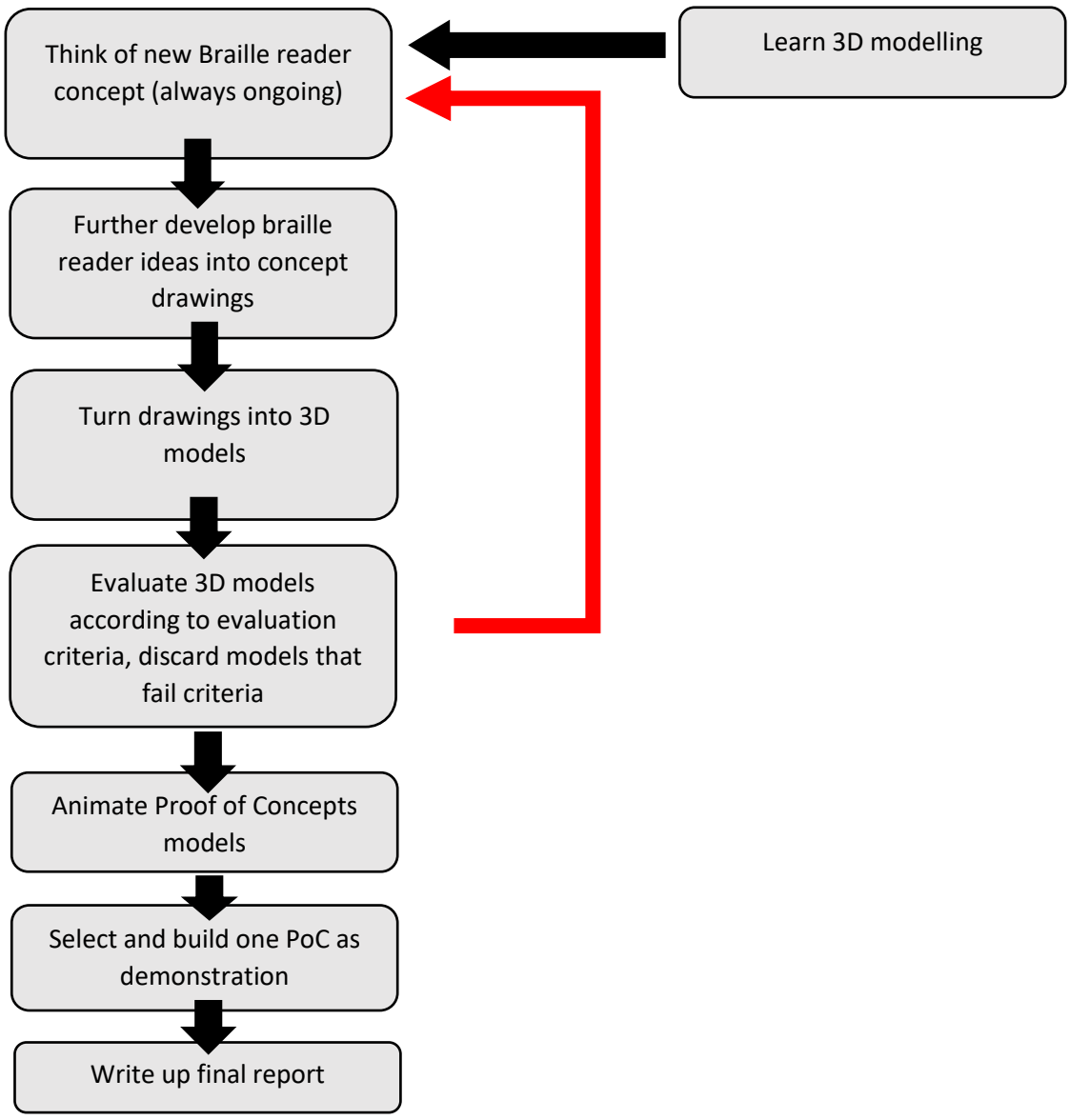
- *Methods (Updated)*

- Start off with simple placeholders (Unity primitive object) in unity, being able to move the placeholder and display information on the constraint of each component.
- Total score of constraints
- Rotate the placeholders
- Detect when placeholders are next to each other, Collison detection, avoid clipping
- Adjust length, width and depth of placeholder. Some placeholder represents sheet material such as carboard or string.
- Replace placeholders with real objects
- Create a menu for the program, ability to save and store designs.
- Ongoing testing and write-up of final report

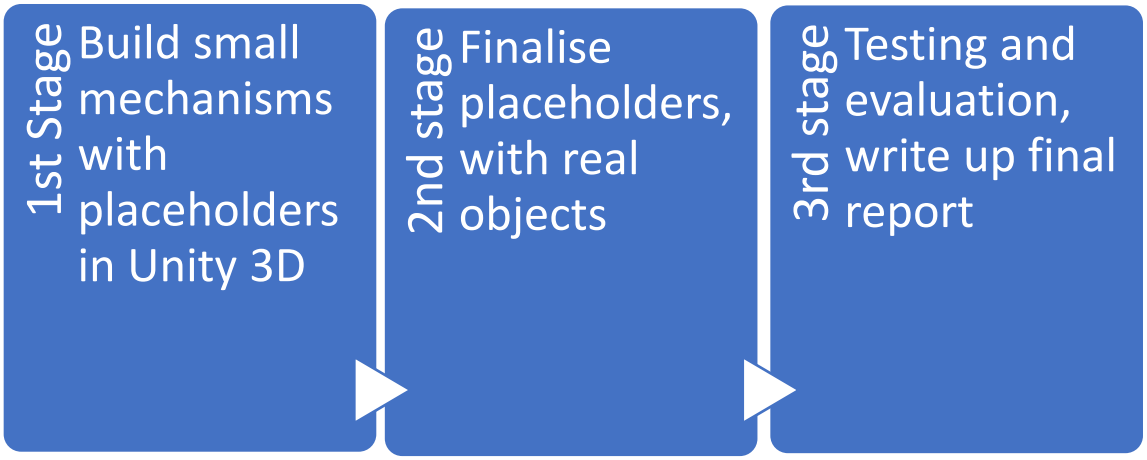
Braille Reader Evaluation Criteria:

- Low cost (under £100) to build.
- Must use physical materials that are available in electronic shops, electronic dumps or salvaged from cheap (under £20) existing electronic products.
- Must only use electronic components that are available in electronic shops, electronic dumps or salvaged from cheap (under £20) existing electronic products
- Must only be built with consumer available tools, not Industrial tools. Therefore, no factories but small 3D printers are allowed.
- Must always consider how a blind person actually reads braille, ideally low cognitive load in using the reader.
- Must be portable (battery powered), relatively lightweight and durable.

(Old) Project plan 12 weeks duration



Updated Project Plan 12 weeks



Progress to date

The Main inspirational Book I read was *Code: The Hidden Language of Computer Hardware and Software*. I have also begun learning how to use the Blender software. I have completed many drawings of PoC and evaluated them using the Braille Reader Evaluation Criteria, they are ready to be made into 3D models.

I initially started off with experiments with ideas that were feasible, I initially had many designs, but I had to narrow down the search space into solutions that are actually low cost and feasible:

First experiment:

I've decided that after experimenting with DIY electromagnets that there is no way to control the repulsion which is the key mechanism that allows a braille dot to be raised or lowered. The problem I discovered was that after a electromagnet becomes magnetised it doesn't become de-magnetised when you switch the electromagnet off.

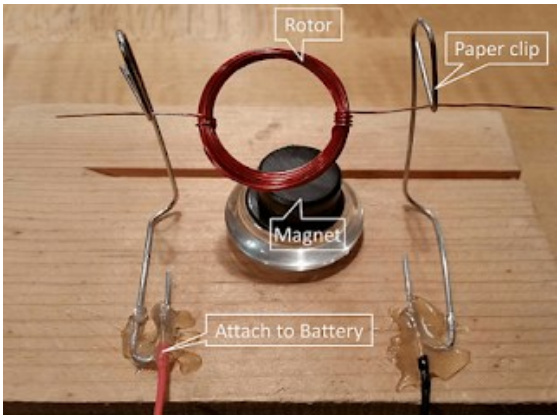
I had to move onto Plan B which was to replace the electromagnet with a simple DC motor. Why DC motor? firstly, they are available everywhere all over the world, there found in many toys and electronics (and electronic landfills), so there accessible in the 3rd world, secondly

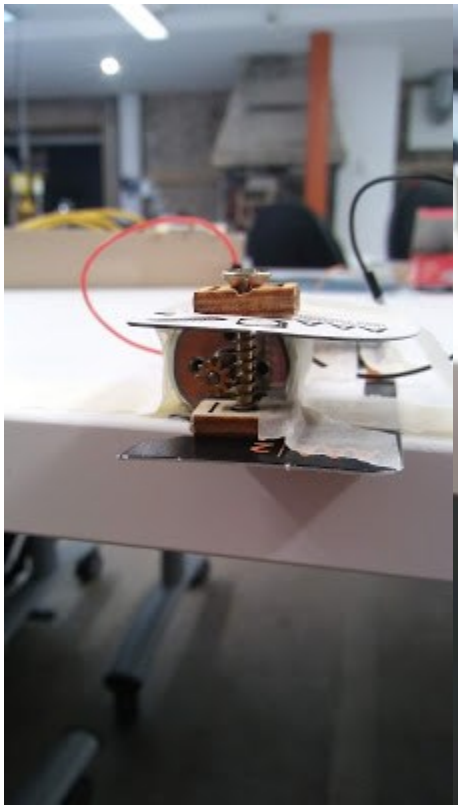
there cheap and lastly, there simple enough to the point where you can make your own just like with the electromagnet.

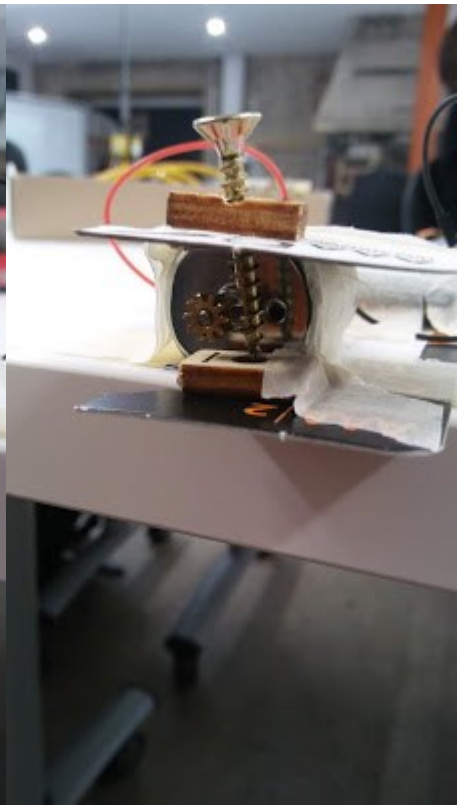
I can still use the metal nails; most dc motors have a gear attached to the rotor. I found that the gear can move upwards and downwards when the DC motor is on and you change which direction of the screw (clockwise/anticlockwise) when you reverse the direction of the battery.

Using the 9V battery was too powerful, I found that the AAA battery was satisfactory and it is much smaller than the 9V battery. For this prototype I used what I could find, the mini wood block is a broken piece I found in the kit and cardboard was from a batter pack.

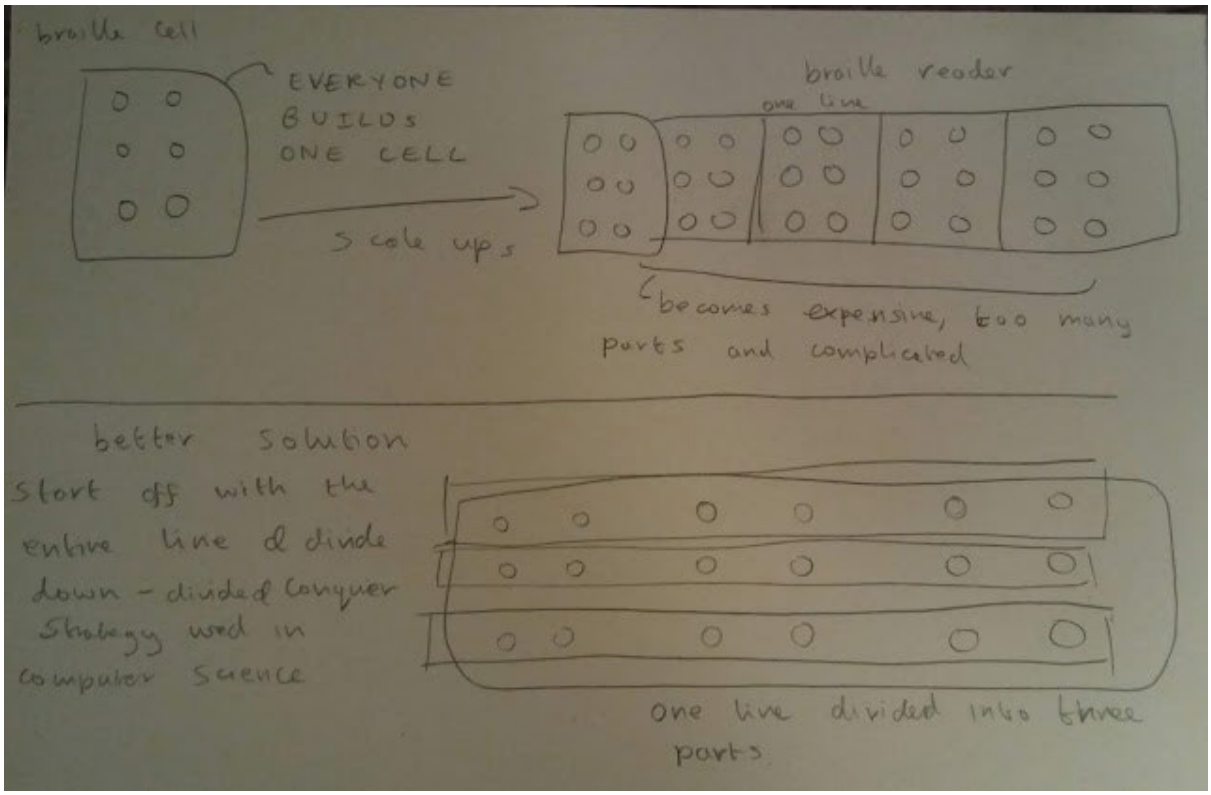
I now need to introduce some control to the DC motor and build a more secure base for the prototype.







Second experiment:

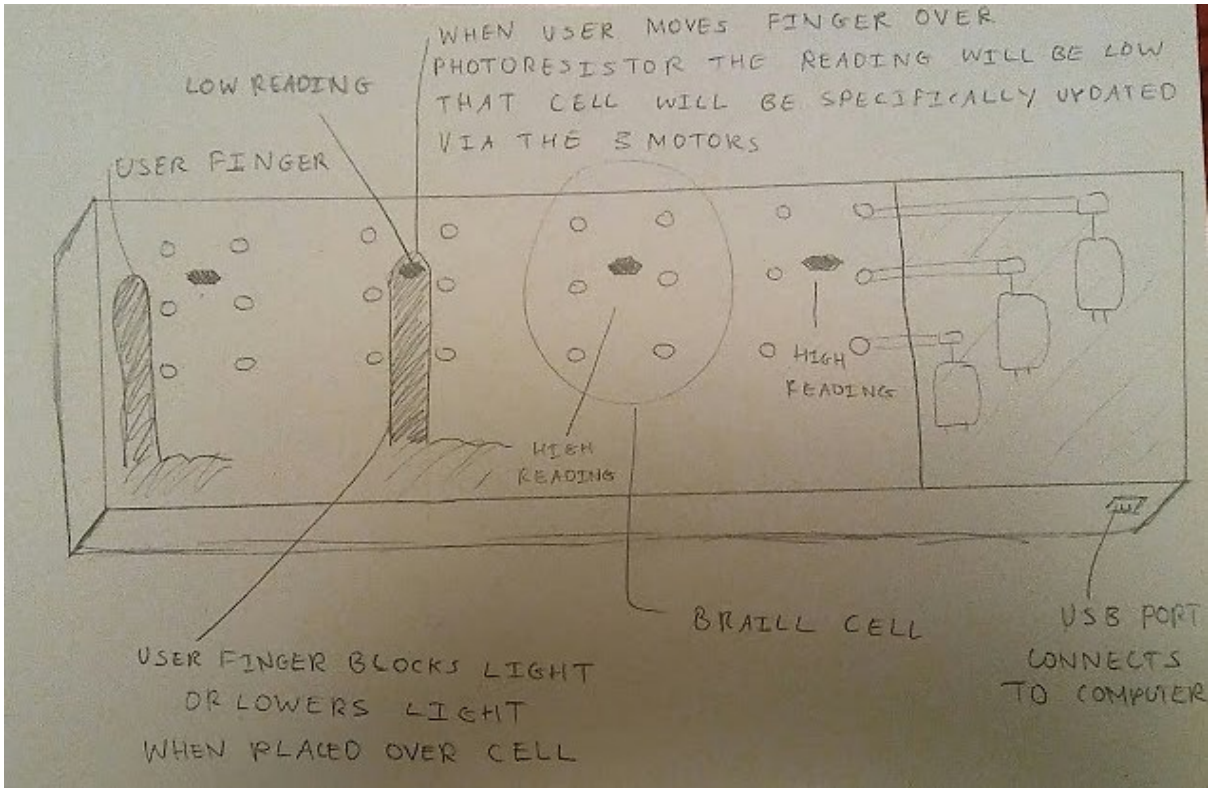


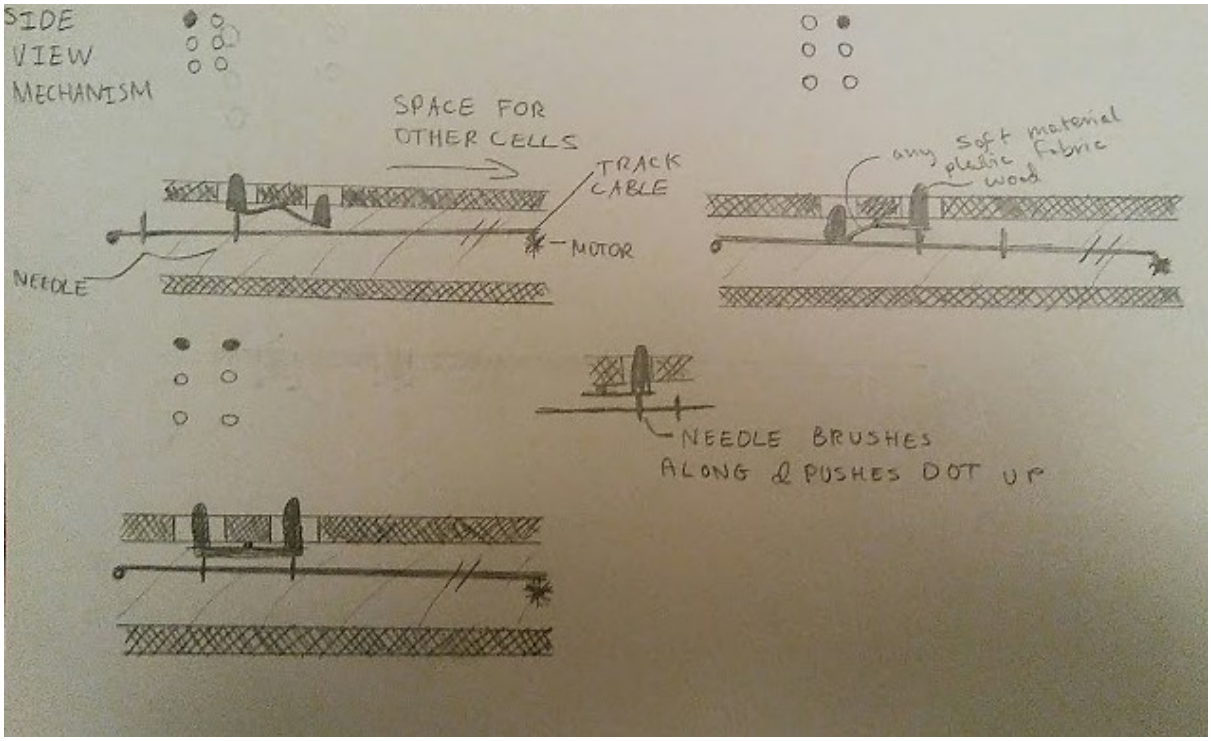
I've thought about the design a lot, really tore it apart in my head and researched other open source attempts, I think I've found a solution that is unique. Instead of building one braille

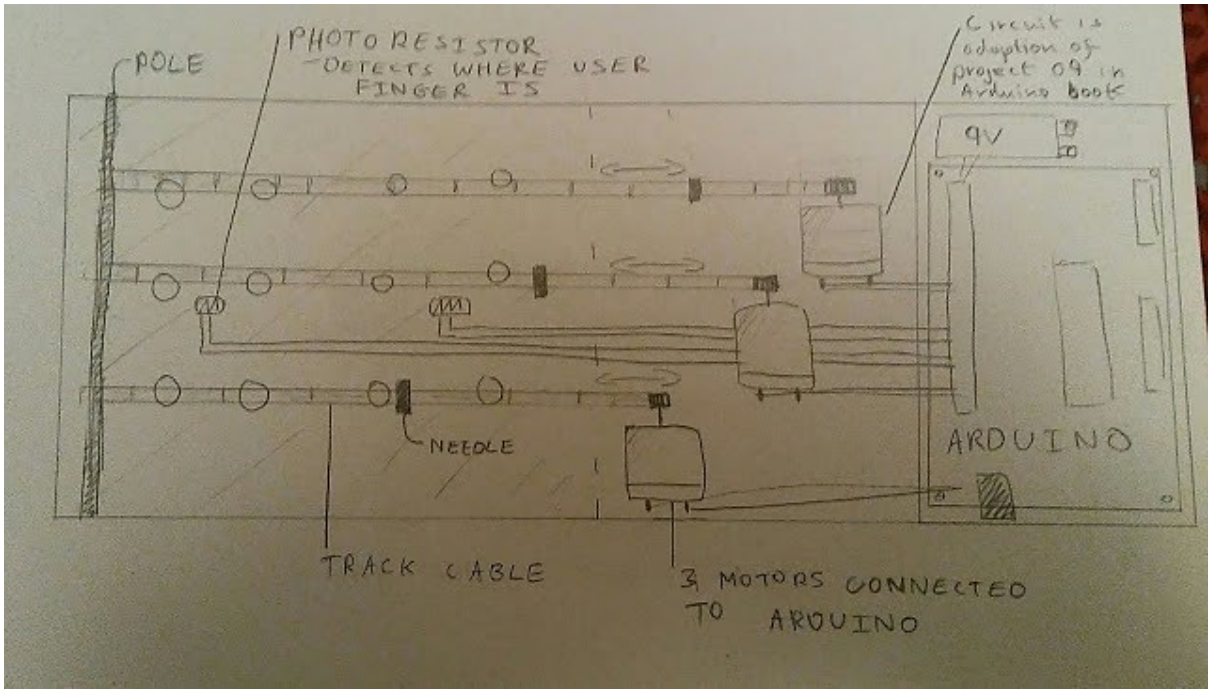
cell and then scaling up, it is more efficient to just divide an entire braille line into three parts.

So, all I had to focus on was simply making one divided part raise a dot or lower a dot. A rail in which a needle was pushed along by a motor was the solution that required less parts.

The only problem is, there is only one needle per divided part how do multiple cells get updated? the solution here is to find a way to detect where the user finger is, which braille cell is there finger currently on. A blind person would read braille by brushing their index finger over the cells. The simplest way to do that is detecting light changes. It's better explained in diagrams than in words.

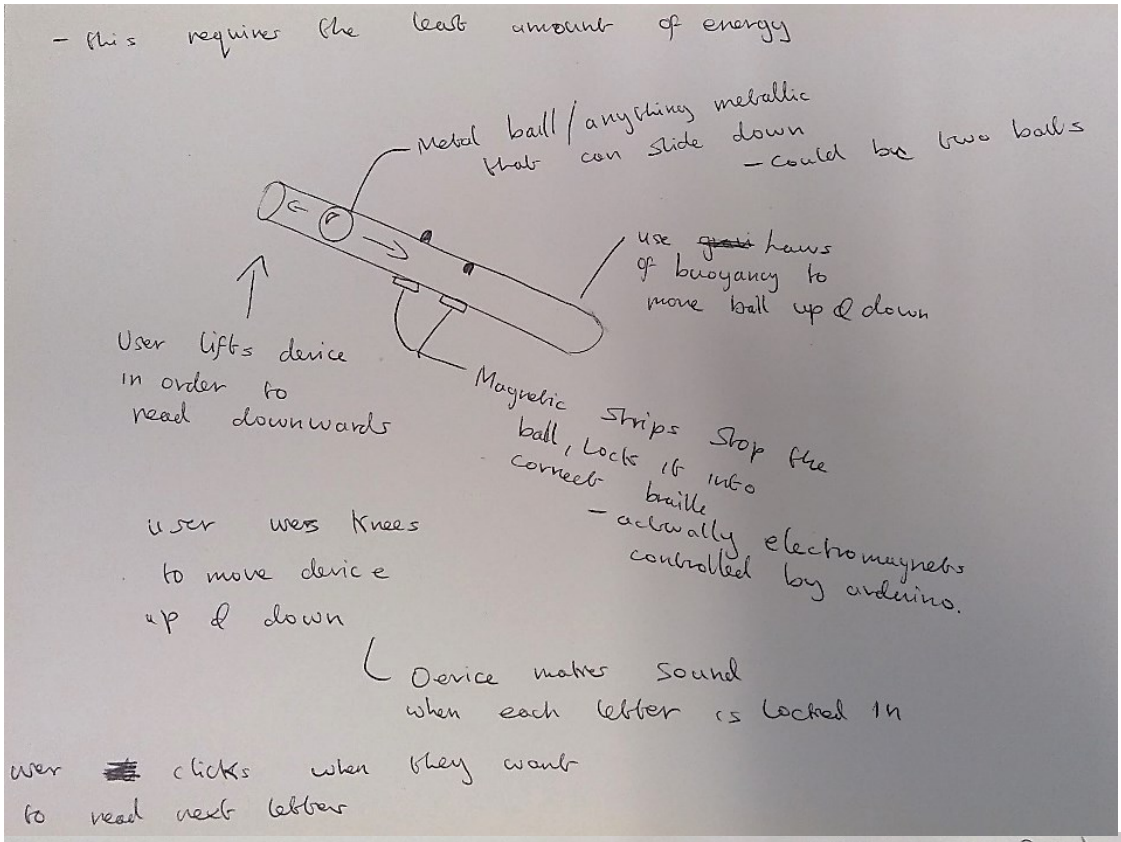


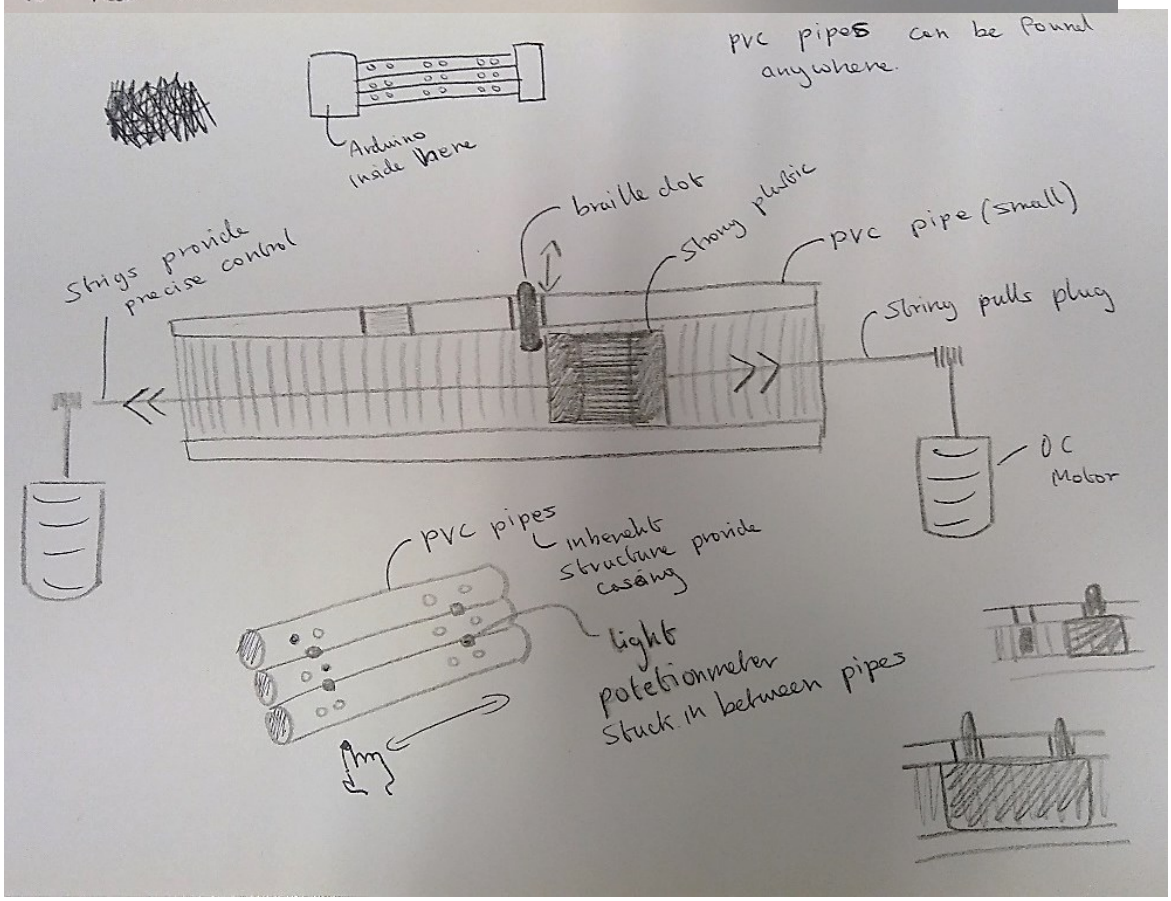


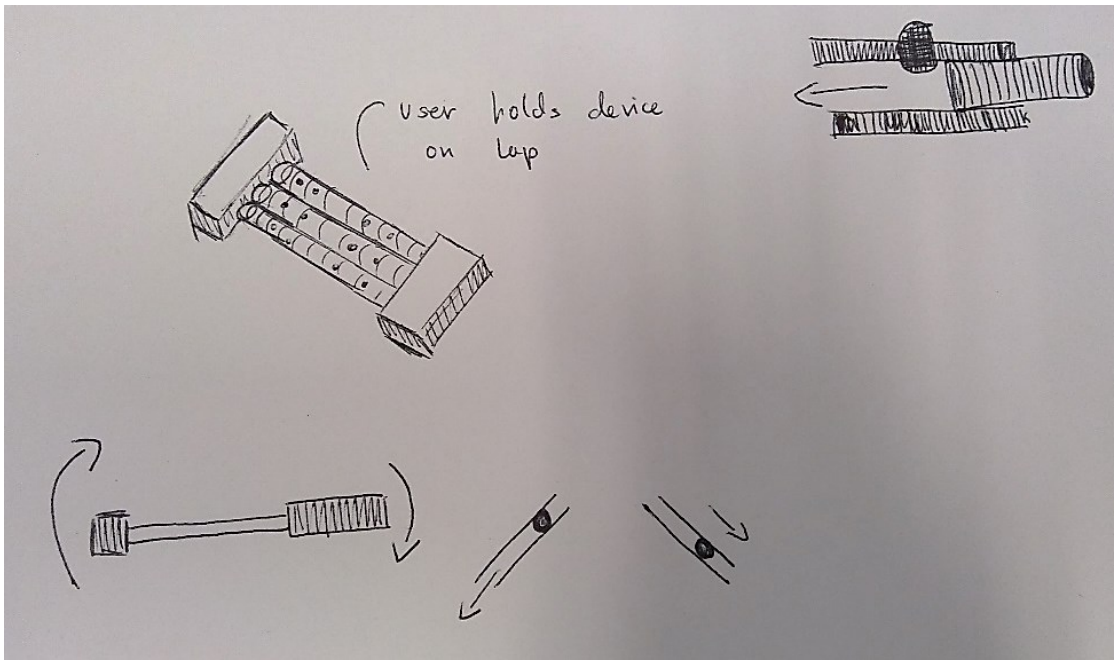


Third experiment:

I further improved the design by using two smaller motors. All of the braille designs I intend to use, all use off the shelf parts, materials that can easily be acquired online, in an electronics store or even from electronic dumps and old electronics products.


- this requires the least amount of energy
Metal ball / anything metallic that can slide down
- could be two balls
use laws of buoyancy to move ball up & down
User lifts device in order to read downwards
Magnetic strips stop the ball, locks it into correct braille
- actually electromagnets controlled by arduino.
user uses knees to move device up & down
Device makes sound when each letter is locked in
user clicks when they want to read next letter


pvc pipes can be found anywhere
Arduino inside here
braille dot
Strong plastic
pvc pipe (small)
Strings provide precise control
String pulls plug
DC Motor
PVC pipes
Inherents structure provide casing
light
potentiometer
Stuck in between pipes

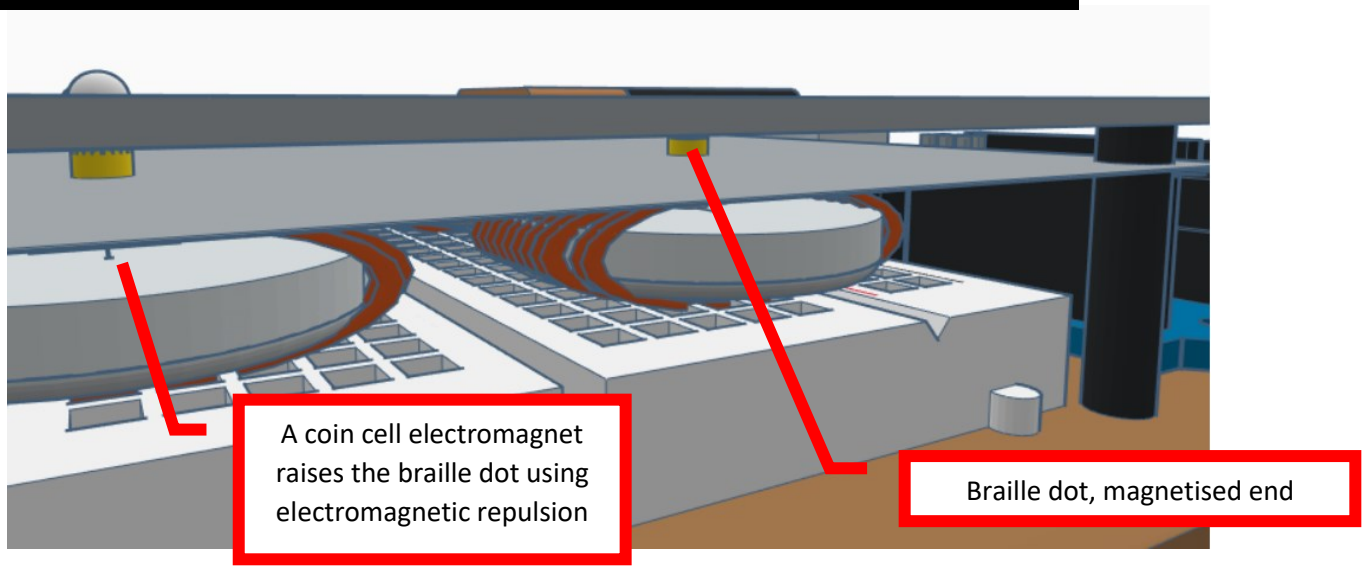


I think the project is complying with the project plan so far. The next stage is to start 3D modelling the designs.

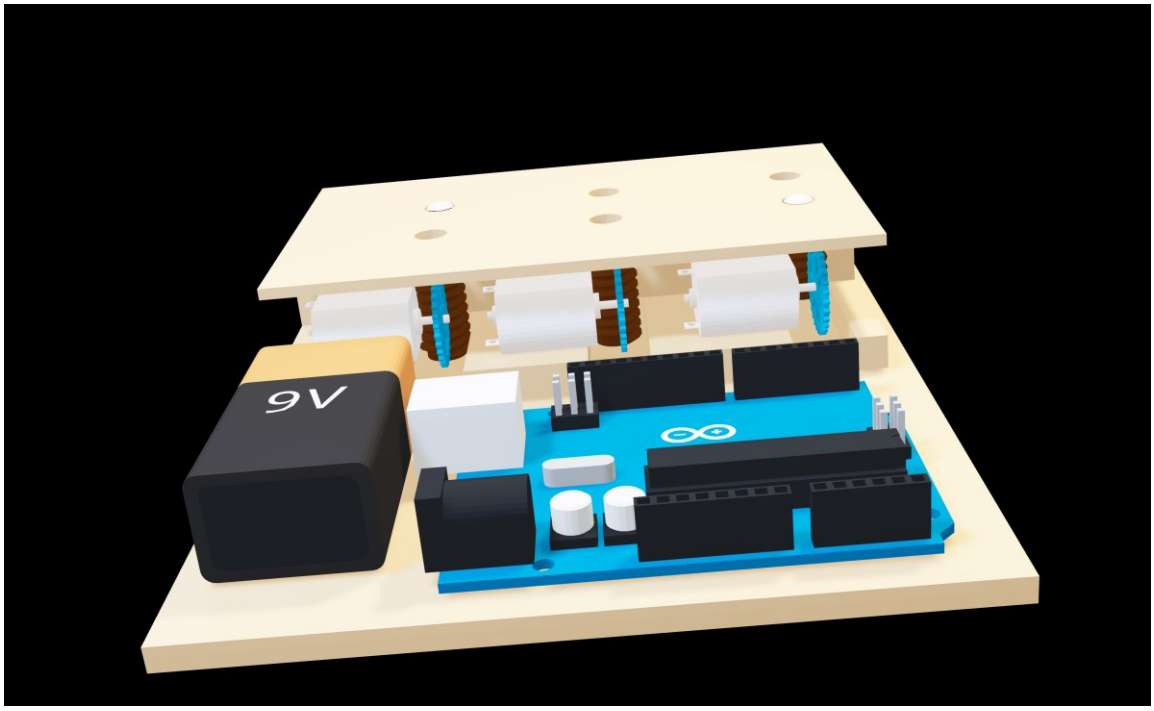
For all the experiments you have seen previously I have now 3D modelled all of them, this is ahead of schedule, so much so that there is more than enough time in the project to build a tool. Throughout my work with this project I realised that it is more effective to build a tool rather than a finished product as sole developer.

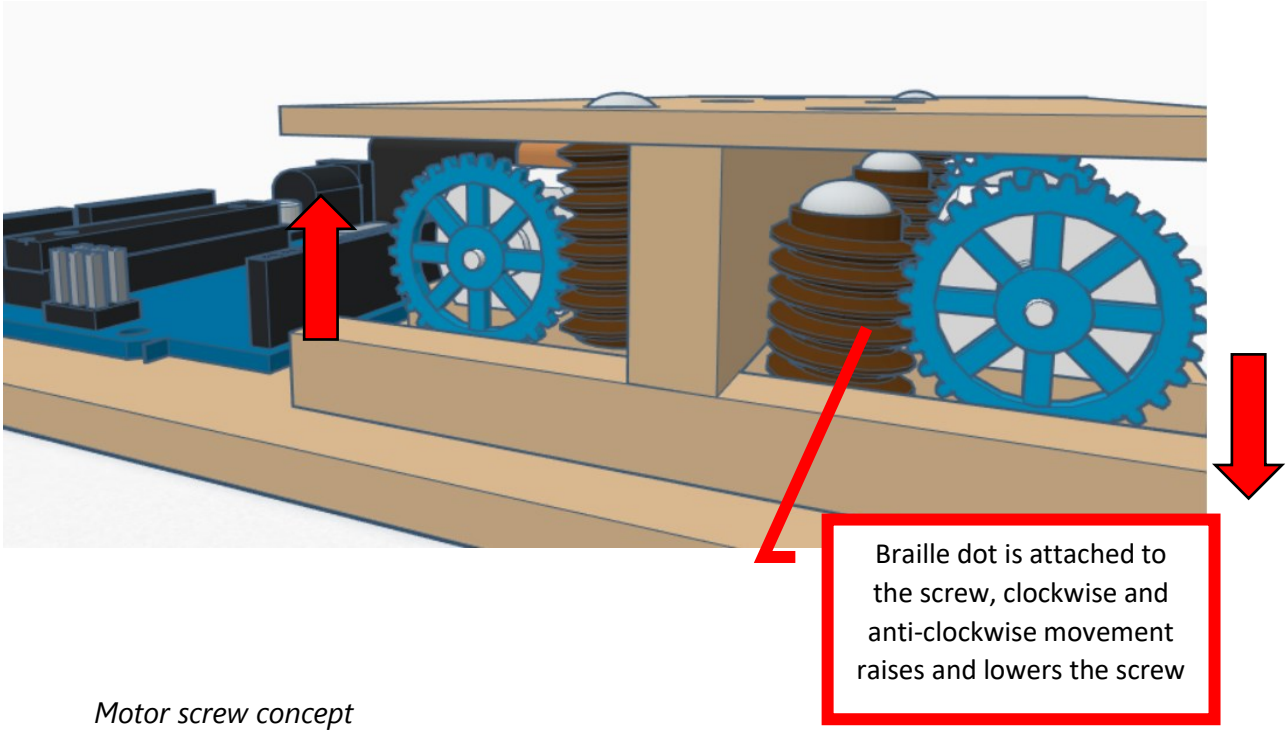
019

the key
ages.

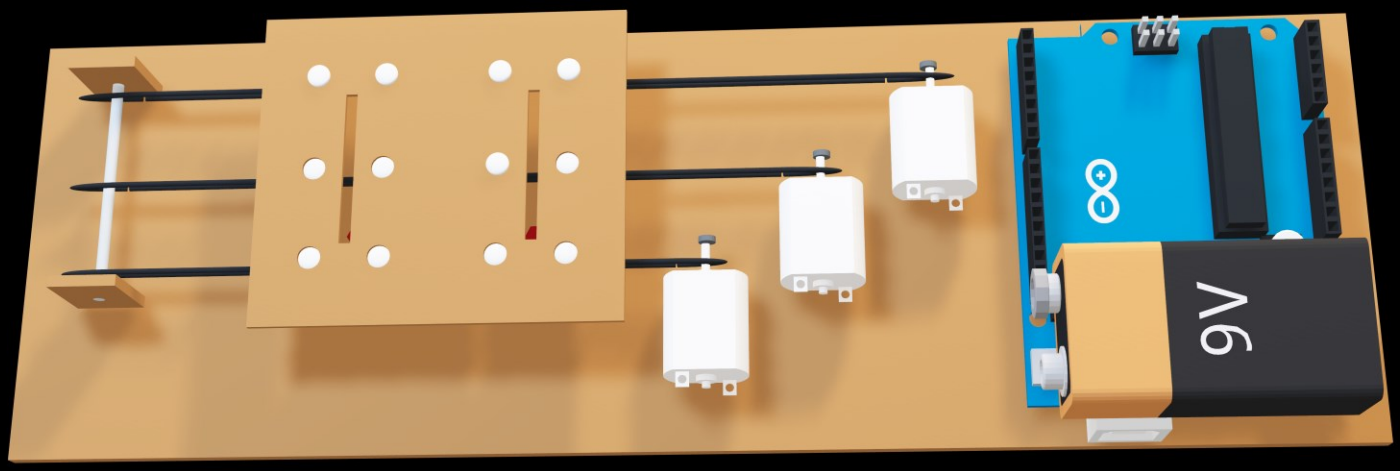


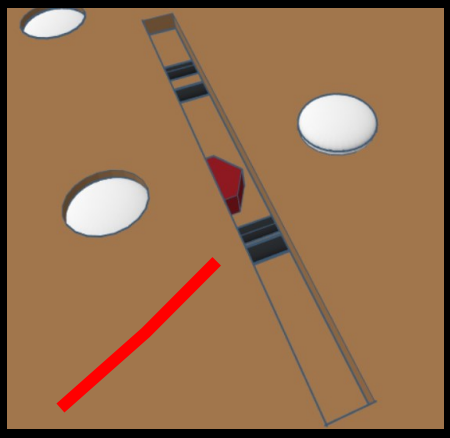
Electromagnet Concept



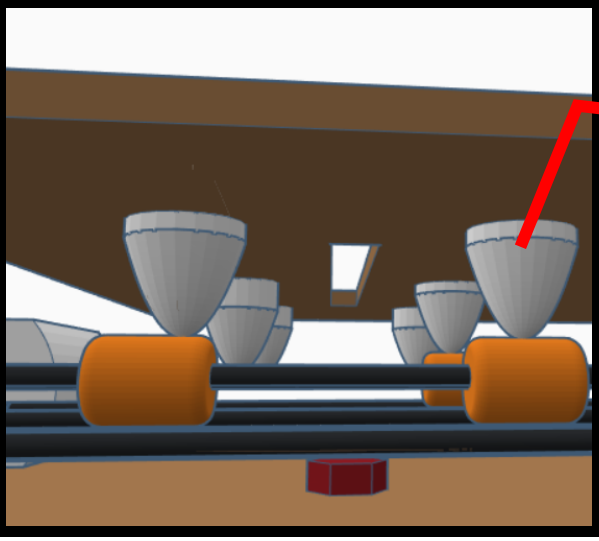


Motor screw concept

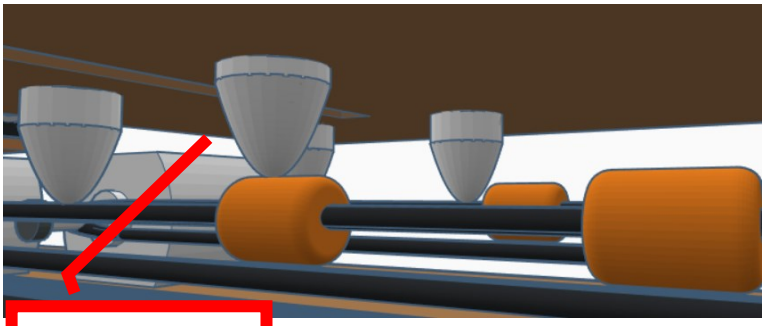




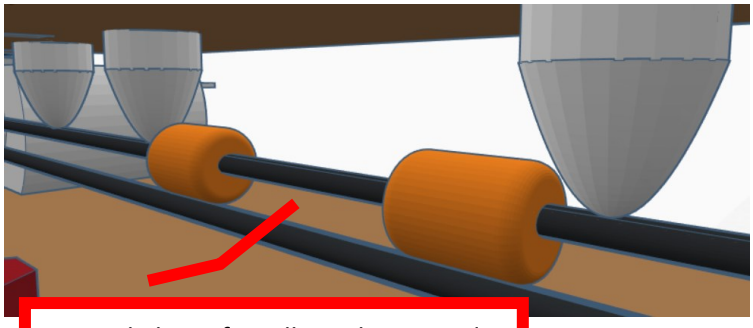
Photometer, the gap lets light in. A low or lower reading means that the user has passed their finger into the vicinity of the braille cell



Both Braille dots are raised

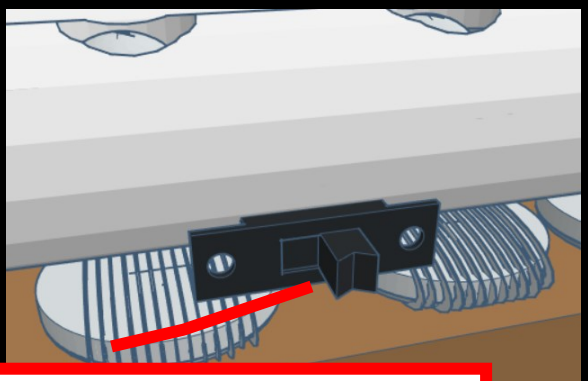


One cell raised

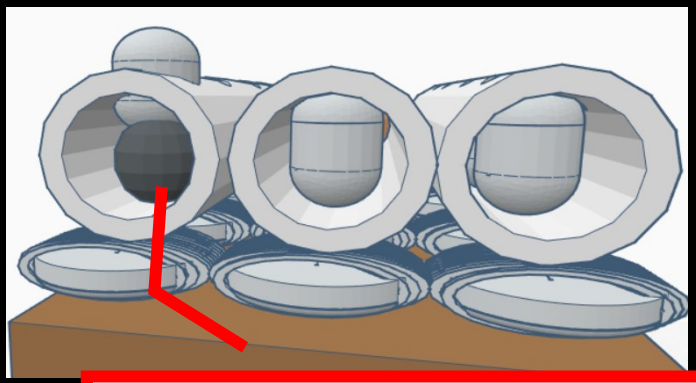


In single line of a cell, no dots raised. Notice, the next cell is not affected

Rail concept revision 1



Switch for user to change locked position of magnet ball.



Magnet ball stopped by the electromagnets. User tilts the reader to move from one braille cell to the next.

Planned work

- Currently all the PoC drawings are complete. [complete]
- Learnt all the required knowledge for making a PoC reality, I am competent with Arduino and electronics. [complete]
- Currently learning 3D modelling with Blender.
- I am going to start 3D modelling the PoC soon after a reach a satisfactory level with Blender. [complete]
- Start building suing placeholders in Unity

References and Bibliography

Number of blind people in the world: https://www.thelancet.com/journals/langlo/article/PIIS2214-109X(17)30293-0/fulltext#seccestitle180 an estimated 36 million people are blind according to the lancet in 2015.

Main book that interested in braille: https://www.amazon.com/Code-Language-Computer-Hardware-Software/dp/0735611319

Make: Encyclopedia of Electronic Components Volume 1: Resistors, Capacitors, Inductors, Switches, Encoders, Relays, Transistors https://www.amazon.co.uk/Make-Encyclopedia-Electronic-Components-Transistors/dp/1449333893

Make: Encyclopedia of Electronic Components Volume 2: LEDs, LCDs, Audio, Thyristors, Digital Logic, and Amplification https://www.amazon.co.uk/Make-Encyclopedia-Electronic-Components-Amplification/dp/1449334180/ref=cm_cr_arp_d_product_top?ie=UTF8

Make: Encyclopedia of Electronic Components Volume 3: Light, Sound, Heat, Motion, Ambient, and Electrical Sensors https://www.amazon.co.uk/Encyclopedia-Electronic-Components-Orientation-Oscillation-ebook/dp/B01DYXGCYW/ref=dp_kinw_strp_1

The Art of Electronics 3rd Edition

Burns Braille Guide: A Quick Reference to Unified English Braille, Second Edition https://www.amazon.co.uk/Burns-Braille-Guide-Reference-Unified-ebook/dp/B015QJ8290/ref=sr_1_2?ie=UTF8&qid=1549637161&sr=8-2&keywords=braille+UEB