



VRIJE
UNIVERSITEIT
BRUSSEL



Graduation thesis submitted in partial fulfilment of the requirements for the degree of
Master of Science in Applied Sciences and Engineering: Computer Science

PHIXEL: AN INTERACTIVE PHYSICAL USER INTERFACE FOR EXPLORING DATASETS

XUYAO ZHANG
Academic year 2019–2020

Promoter: Prof. Dr. Beat Signer
Advisor: Prof. Dr. Beat Signer
Faculty of Sciences and Bio-Engineering Sciences



VRIJE
UNIVERSITEIT
BRUSSEL



Afstudeer eindwerk ingediend in gedeeltelijke vervulling van de eisen voor het behalen van de graad Master of Science in de Ingenieurswetenschappen: Computerwetenschappen

PHIXEL: AN INTERACTIVE PHYSICAL USER INTERFACE FOR EXPLORING DATASETS

XUYAO ZHANG
Academiejaar 2019–2020

Promoter: Prof. Dr. Beat Signer
Advisor: Prof. Dr. Beat Signer

Faculteit Wetenschappen en Bio-ingenieurswetenschappen

Deze masterproef is (ten dele) tot stand gekomen in de periode dat het hoger onderwijs onderhevig was aan een lockdown en beschermende maatregelen ter voorkoming van de verspreiding van het COVID-19 virus. Het proces van opmaak, de verzameling van gegevens, de onderzoeksmethode en/of andere wetenschappelijke werkzaamheden die ermee gepaard gaan, zijn niet altijd op gebruikelijke wijze kunnen verlopen. De lezer dient met deze context rekening te houden bij het lezen van deze masterproef, en eventueel ook indien sommige conclusies zouden worden overgenomen.

This master's thesis came about (in part) during the period in which higher education was subjected to a lockdown and protective measures to prevent the spread of the COVID-19 virus. The process of formatting, data collection, the research method and/or other scientific work the thesis involved could therefore not always be carried out in the usual manner. The reader should bear this context in mind when reading this Master's thesis, and also in the event that some conclusions are taken on board.

Abstract

Data physicalisation is a promising method to facilitate the communication with and understanding of virtual data. It makes use of human's active perception abilities and allows the data to be touchable. People have long realised the benefits of representing data with physical properties of artefacts, and created a large number of data physicalisation products in a range of fields. Information visualisation is one of the representative areas, which utilises the advantages of data physicalisation to help users reason about data.

Most of the previous work on physical data representation requires users to first fabricate the objects and then perform interactions manually. This static and passive representation is hard to modify and thus does not enable dynamic updates that are supported by screen-based visualisations. Tangible User Interfaces (TUIs) and shape-changing interfaces are both emerging solutions to address this limitation. While the former is mostly used as input, the latter is usually applied for output. This thesis presents Phixel, an interactive and dynamic physical interface for dataset exploration. In the Phixel application, users take tangible objects as input and render a shape-changing display as output. The proposed application is constructed by 40 bars (pixels) which are arranged in a 5×8 bar chart. Similar to an on-screen bar chart, the physical bar chart supports fundamental visualisation tasks such as filtering and organisation.

In order to evaluate the usefulness and usability of Phixel, a user study was conducted with 10 participants. They were asked to use Phixel to explore a dataset on food nutrient content based on some food related questions. Results and participants' feedback have shown that Phixel can greatly enrich users' perception and facilitate their manipulations to gain deeper insights from data.

This thesis also illustrates the potential employment of Phixel in two different scenarios, as well as its further development in terms of a variety of aspects. Besides, the strengths and limitations of using Phixel are discussed. In general, Phixel is an enjoyable data physicalisation application, which enables users to explore datasets in an intuitive way. It is expected that the approach proposed in this thesis can provide some new ideas for developments in the domain of dynamic data physicalisation.

Acknowledgements

First, I would like to express my deepest appreciation to Prof. Dr. Beat Signer for his thorough and patient guidance throughout the duration of this thesis. He played a decisive role in proposing valuable suggestions and effective solutions for the Phixel system. I had great pleasure of working with Professor Signer and his enthusiasm for the area of dynamic data physicalisation has encouraged me a lot on the further development of the Phixel system and other relevant products.

Second, I would like to extend my sincere thanks to members of the WISE lab. I would especially like to thank Maxim Van de Wynckel and Maxime Libert for the great amount of assistance in 3D printing and Arduino hardware. They always facilitated me when I encountered tricky problems and their help and kindness will always be remembered.

Finally, I gratefully acknowledge the incredible effort of my family. They have provided me the most important support throughout my academic career. Special thanks to my mother for her unwavering belief in my abilities and useful advice on the thesis.

Contents

1	Introduction	
1.1	Benefits of Data Physicalisation	2
1.2	Problem Statement	3
1.3	Contributions	5
1.4	Thesis Outline	6
2	Related Work	
2.1	Static Data Physicalisation	7
2.2	Dynamic Data Physicalisation	10
2.3	Other Techniques for Exploring Datasets	14
2.4	Approaches for Exploring Datasets about Food	14
2.5	Thesis Motivation	15
3	Solution	
3.1	Design Idea	18
3.2	Architecture and Functionalities	19
4	Implementation	
4.1	Hardware	21
4.1.1	Hardware Setup	22
4.1.2	Android Device	23
4.1.3	Arduino Microcontrollers	23
4.1.4	Bluetooth Module	25
4.1.5	Port Expander	26
4.1.6	RFID Technology	26
4.1.7	Stepper Motor and Driver	28
4.1.8	LEDs	28
4.1.9	3D Printed Items	29
4.2	Software	32
4.2.1	Android Studio	33
4.2.2	Arduino IDE	39

5	Evaluation	
5.1	Procedure	45
5.1.1	Dataset Exploration	46
5.1.2	Questionnaire and Interview	48
5.2	Results and Feedback	50
5.2.1	Pre-test Result	50
5.2.2	Dataset Exploration and Observation Result	51
5.2.3	Questionnaire Analysis	52
6	Use Cases	
6.1	Food Nutrition Exploration Scenario	57
6.2	Marketing Analysis Scenario	59
7	Future Work	
7.1	Including Personal Information	61
7.2	Including More Data Analysis Tasks	62
7.3	Including Other Modalities	63
8	Discussion and Conclusion	
8.1	Discussion	65
8.2	Conclusion	67
A	Appendix	
A.1	Circuit Diagram Appendix	69
A.2	Evaluation Records Appendix	72

1

Introduction

Representing data in physical form has a long history. As early as 8000 BC, clay tokens were used to represent numerical quantities of goods, which is the earliest physical data visualisation, that happened even before the invention of paper and writing [41]. In the 5th century BC, Greeks voted on the competing claims of heroes Ajax and Odysseus by dropping pebbles in an urn, expressing their opinions by the colour of pebbles¹. In the 19th century, the Torres Strait Islanders still tied sticks to a rope and used those bundles to help them record their prey, and it is believed that their way of representing numbers is much older and could be traced back to when those people lacked a numeral system [14].

All these cases belong to the field of physical data visualisation, also known as data physicalisation. Jansen et al. [22] proposed this term as “*a physical artifact whose geometry or material properties encode data*”. This area is closely connected to several other disciplines such as digital fabrication or tangible user interfaces (TUIs). These domains have been mutually reinforcing and have flourished over the past two decades, offering great potential for gaining insights into virtual data in natural ways.

Creating data physicalisations usually requires a lot of time. However, the advancements of digital fabrication technologies, especially 3D printing

¹<http://blogs.getty.edu/iris/voting-with-the-ancient-greeks> (accessed 26/2/2020)

and laser cutting, have sped up this process and made it much easier to build physical manifestations. With the increasing availability of these technologies, fast production of customised objects has become prevalent, thus boosting the development of data physicalisation.

Another research area tightly related to data physicalisation are TUIs, which aim to bridge the gaps between the virtual on-screen and the real world [19]. A main difference between data physicalisation and TUIs is that TUIs not only use physical artefacts to represent digital information, but also to control it [48]. In this respect, TUIs can be considered as a kind of data physicalisation with dynamic properties.

1.1 Benefits of Data Physicalisation

The advantage of encoding data with physical objects is apparent: people can touch, grasp or move objects to interact with data, rather than just seeing them on the screen. The perceived affordances [33] of physical objects also promote people to manipulate them intuitively. Many studies have been conducted on the benefits of data physicalisation, in order to find out how to make the most of it.

Learning and Education

Data physicalisation can leverage one's cognitive skills and be well applied in the education area [22]. It has been used as an important tool for the preparation of educational content. For example, 3D physical models of molecules provide a straightforward way to help students or researchers learn chemical structures [40, 51]. Similarly, Do-Lenh et al. [9] designed an experiment in which students learn the layout of a warehouse by setting tangible shelves in an augmented tabletop environment. The results suggested that the tangibility of physical objects could improve the efficiency of completing tasks without affecting learning outcomes. In addition, groups working in this environment got more fun and had a better collaboration [1, 42].

Spatial Flexibility and Memorability

Compared to 2D or 3D flat user interfaces (UIs), data physicalisation offers more spatial flexibility and convenience: users can communicate data in the physical world outside the screen, rather than simply pressing a computer mouse or clicking on a virtual button. In terms of spatial memorability, Bevan and Steger [5] did a recall test and found that people recalled physical

objects more often than digital pictures. Furthermore, Stusak et al. [44] investigated the memory decay of information gathered from a static physical bar chart and a visual one. The results suggested that physical representations outperformed visual ones in both immediate recall and delayed recall.

Reasoning and Understanding

In many cases, data physicalisation overlaps with data visualisation, as they share the same target to engage people in the world full of numbers and help them get a better comprehension of abstract data. Both of these approaches transform raw data into an easy-to-understand structure, so that people can quickly grasp the story underlying the data.

However, some experiments have shown that data physicalisation behaves better than data visualisation with respect to the understanding of and reasoning about data. Jansen et al. [21] compared the impact on exploring datasets between physical 3D bar charts and on-screen 3D visualisations, and found that although data visualisation raised the efficiency of processing complex data, rendering the visual structure into the real world enriched users' experience and facilitated performing information retrieval tasks. Similarly, Taher et al. [46] investigated how dynamic physical bar charts can engage people and help them gain deeper insights to large datasets. They also mentioned that data physicalisation provided rich interaction possibilities and made the exploration procedure much more enjoyable.

1.2 Problem Statement

As mentioned above, data physicalisation has strong advantages in many aspects, which have strongly motivated research on the application of data physicalisation. This thesis focuses on data physicalisation for data analysis, especially for large dataset analysis, which unfortunately, has not yet been discussed or implemented by many people.

Most of previous physical data visualisations are either used for illustrating a simple picture without containing much information inside (e.g. Figure 1.1), or for raising one's awareness in certain fields as aesthetic artefacts (e.g. Figure 1.2). Apart from that, the great majority are passive and static, which means that the physical objects will become out of synchronisation once the original data changes. Even worse, the reproduction of updated physical objects is usually time consuming, making the whole process inefficient and annoying.

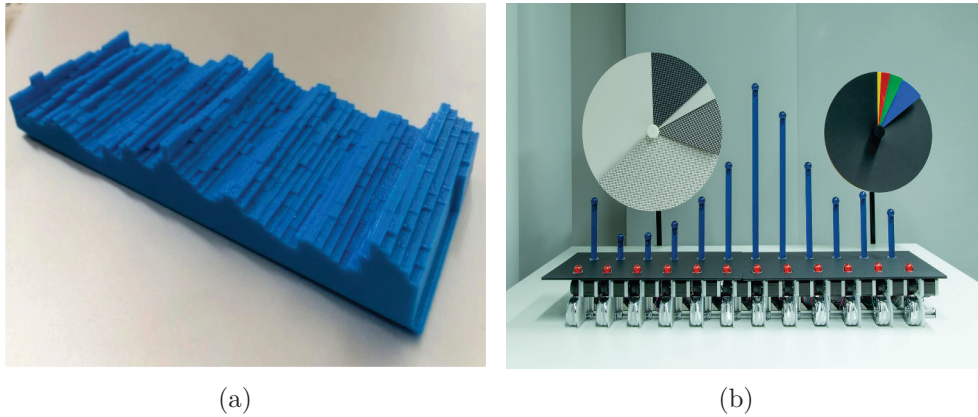


Figure 1.1: Examples of physical visualisations for conveying information: (a) A physical chart of the U.S. unemployment rate between 1948 and 2015, with each column representing a month, designed by Jon Keegan; (b) Physical pie charts showing statistics of people who live or work on Tenison Road in Cambridge, UK, produced by David Sweeney from Microsoft Research.



Figure 1.2: Examples of physical visualisations as artwork: (a) Average daily traffic data of the U.S. highway system encoded in a 2D area map [28]; (b) Electricity generation data encoded in the jagged card boards [7].

The development of shape-changing displays provides a new way to build physical data visualisations. However, this technology is almost never used for data physicalisation with the purpose of interacting with datasets. Some typical applications of shape-changing interfaces in the field of data physicalisation are discussed in the following chapter. As far as the author knows, EMERGE [45] is the only application which utilises shape-changing displays to construct a dynamic and physical bar chart for analysing datasets. Despite this, the application is limited by its large size, and can only help explore a specific dataset.

The application described in this thesis is inspired by the EMERGE project, with more advanced interaction techniques and more powerful data representation approaches. The presented Phixel (Physical Pixel) solution aims to address the problems stated above, and the research questions are the following:

- How can the combination of TUIs and shape-changing displays be used to construct a dynamic physical bar chart to move data visualisation to the real world?
- Is it useful and meaningful to use dynamic data physicalisation to help users explore and understand datasets?
- Does dynamic data physicalisation have a positive influence on performing basic visualisation tasks in the context of both individual and multi-user setups?

1.3 Contributions

The main objective of this thesis is to take advantage of dynamic data physicalisation to benefit communication with and reasoning about datasets. The notable contributions of this thesis include:

- A dynamic physical bar chart combined with tangible input in order to enhance users' interaction with datasets. This application can be used for individual exploration as well as a multi-user setup to facilitate cooperation. Besides, it also supports different datasets so users are free to choose any dataset to be physicalised.
- A user study that investigates how people use this application and how they think about it. A dataset about food was used in the user study, and an observation experiment was conducted to see how people interact with this application.

- A discussion of the potential and limitations of the presented application, and its possible improvements in the future.

1.4 Thesis Outline

This thesis is organised as follows: Chapter 2 gives an overview of related work, as well as an analysis of their approaches, benefits and drawbacks. Specifically, the first section reviews a number of static data physicalisation applications adopted in different areas, after which some representative examples of dynamic data physicalisation and how they solve the problems of static ones are discussed. Then, several alternatives to on-screen data visualisations are analysed, which have similar purpose of helping people comprehend data. At last, existing work on visualising datasets about food is described, and their advantages and disadvantages are discussed.

Chapter 3 is a brief but complete depiction of how Phixel works, including its design ideas, architecture and functionalities. It offers all the key information of this application. Next, Chapter 4 thoroughly describes the implementation of the new dynamic physical bar chart, which consists of its setup, as well as the use of software and hardware. A number of important libraries for coding main functionalities are also explained. Following this, the procedure of the user study (the design of basic data visualisation tasks and a questionnaire investigating user experiences) is reported and detailed results are discussed in Chapter 5. Chapter 6 presents two different scenarios that Phixel can be used in the future. Chapter 7 illustrates several possibilities of further development of Phixel for a variety of aspects. Finally, the highlights and limitations of the Phixel system are discussed and the proposed work is summarised.

2

Related Work

As data physicalisation is a promising approach to enrich people's understanding of digital information, many studies have been conducted to explore the possibilities and potential in this area. There are two main data physicalisation strands: static data physicalisation and dynamic data physicalisation. This chapter is a review of relevant work of both strands, covering their application fields and foremost technologies, followed by a comparison between data physicalisation and other approaches used for exploring datasets. Besides, some interesting applications in the food domain are discussed, as the food dataset will be adopted in a user evaluation to study the performance of Phixel.

2.1 Static Data Physicalisation

Static data physicalisation can be seen as data sculpture which cannot be modified or updated after being created. People have long used data sculptures to support learning and problem solving with data, thanks to their properties that leverage human's natural abilities. The progress of digital fabrication technologies has highly encouraged the use of data sculptures in various fields.

Scientific Visualisation

The first area in which static data physicalisation is commonly used is scientific visualisation. As many scientists frequently work with 3D data, they have a great need for finding better ways than flat-screen displays to understand presented information. A large number of them prefer to print out scientific data via 3D printers, because current 3D printing technologies enable the production of the same high-resolution objects as on the screen [3]. Bathsheba Grossman created a lot of 3D data physicalisation products through subsurface laser engraving¹. The data she used covered multiple areas such as biology, cosmology or physics. Similarly, Paul Bourke presented scientific findings as 3D crystal engravings for better communication about research results at conferences². Alternatively, some researchers printed full-colour 3D models of scientific visualisations through additive manufacturing [4], which is also the main technology for 3D printing. Figure 2.1 shows two examples of physical scientific visualisations using laser engraving and additive manufacturing, respectively.

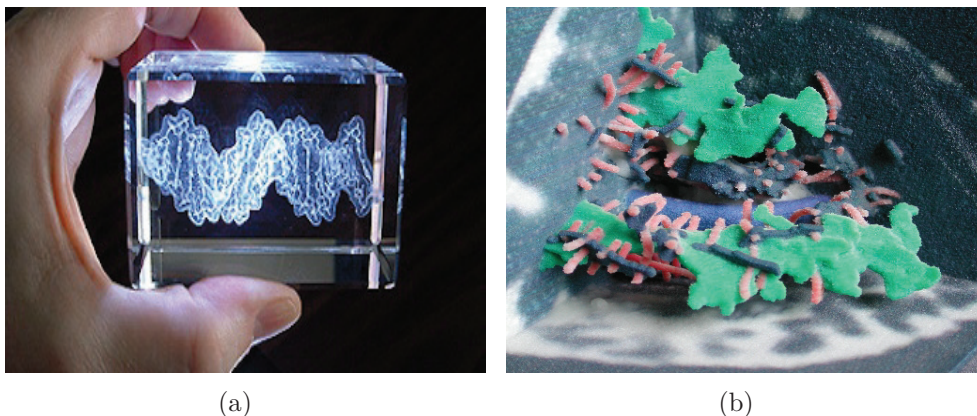


Figure 2.1: Static data physicalisation examples using two different 3D printing technologies: (a) Laser engraving example: A piece of DNA molecule produced by Bathsheba Grossman; (b) Additive manufacturing example: A plaster model of planer water self-diffusion [4].

Aesthetic Work

While scientists take good advantage of static data physicalisation for understanding and communication, artists noticed the beauty of static data

¹<https://bathsheba.com> (accessed 22/3/2020)

²<http://paulbourke.net/miscellaneous/crystalengraving> (accessed 22/3/2020)

physicalisation and have built some enjoyable data physicalisation work for expressing their claims and telling the story behind the data. For example, Meshu³ helps people memorise places that are important to them by turning space information into jewellery or print. Other artists embedded weather or temperature data into common artwork, such as woven sculptures [15], bracelets⁴ and scarves⁵, to raise the awareness of climate change. Moreover, there are designers who encoded air pollution data into clothing [34] and necklaces⁶, of which uncomfortable materials or sharp shapes indicate bad air quality, thus promoting people to take actions to improve the air environment (see Figure 2.2).



Figure 2.2: Air pollution data physicalisation examples: (a) Dresses show the concentrations of 100 chemical contaminants measured in the air of a particular household. The small squares represent different chemicals and the square size represents the chemical concentration; (b) Air quality data measured in the city of Sheffield for one week. Each segment represents 6 hours, and its shape represents the concentration of particulate matter during that period of time, designed by Stefanie Posavec.

³<http://meshu.io> (accessed 23/3/2020)

⁴<http://dataphys.org/list/weather-bracelet-and-measuring-cup> (accessed 25/3/2020)

⁵<http://dataphys.org/list/temperature-afghans-and-scarves> (accessed 25/3/2020)

⁶<http://dataphys.org/list/touching-air-necklace-shows-air-pollution> (accessed 25/3/2020)

Data Analysis

Various static data physicalisations have been applied in the area of data analysis for scientists or regular people to make sense out of data. Numerous and complex numbers make people pay little attention on things that might be very relevant to them, but data physicalisation provides an alternative to virtual information displays to engage their analytical mind [21, 22].

There are mainly two common forms to show digital numbers in the real world: shape or the quantity of an object. The first can simply be achieved by fabricating objects of various heights or sizes. In general, the colour of objects also encodes additional dimension of data. For example, Moher [30] mapped the earthquake data to styrofoam balls hung from classroom ceiling. The height at which a ball hangs, the colour of the ball, as well as its diameter, represent the intensity of an earthquake. Rase [38] created several physical 3D maps to illustrate some statistical data for Germany (such as unemployment rate, average building price or travel distance), with different surfaces representing the data trends.

The second form requires placing or stacking some objects together to represent different data (see Figure 2.3). Using LEGO⁷ bricks to build physical bar charts is also a popular way to show data in physical form. Researches have been stacking and rearranging them to illustrate federal election⁸, the tax system⁹, and social mobility¹⁰. However, since it is just a process of constructing bricks, it can be considered as the same as the second form.

2.2 Dynamic Data Physicalisation

The main drawback of static data physicalisation, as mentioned above, is that it is constructed from passive blocks, and gets disconnected from the data source once built. In contrast, dynamic data physicalisation overcomes this limitation by combining digital fabrication with other technologies.

Augmented Reality

The first way of making data physicalisation dynamic is involving Augmented Reality (AR) in the environment. A typical example are molecular biology

⁷<http://lego.com> (accessed 23/3/2020)

⁸<https://www.youtube.com/watch?v=1aUPeXZlPEg&feature=youtu.be> (accessed 27/3/2020)

⁹<https://www.brookings.edu/blog/up-front/2015/04/09/how-much-does-the-tax-code-reduce-inequality/> (accessed 27/3/2020)

¹⁰https://www.youtube.com/watch?v=t2XFh_tD2RA (accessed 27/3/2020)

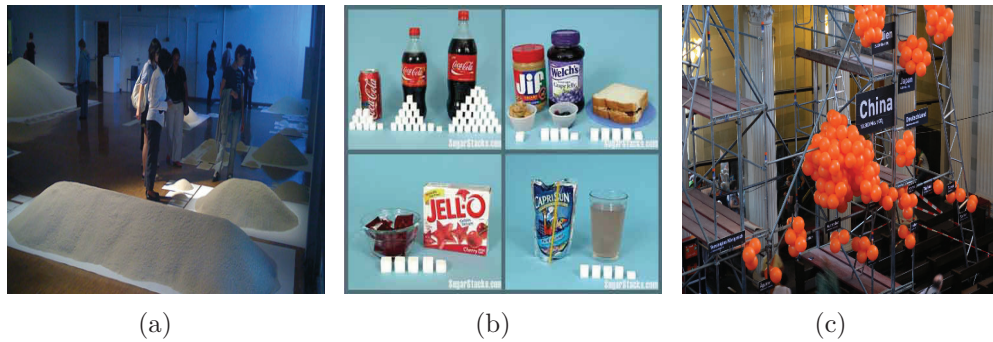


Figure 2.3: Examples of physical visualisations: (a) Grains of rice represent population statistics, created by a group of artists in Stan’s Cafe; (b) Regular sugar cubes represent sugar consumption of different food or drinks, seen in [6]; (c) Balloons show the CO₂ emissions of some countries in 2014, with each balloon representing 100 billion tonnes of carbon dioxide, created by Mario Klemm and José Ernesto Rodriguez.

applications [12, 13], which overlaid augmented virtual 3D representations onto 3D printed molecular models. Users can see different molecular properties by easily changing virtual information. Priestnall et al. [36] projected data visualisations on physical landscape models to display landscape details. This Projection Augmented Relief Model not only served aesthetic purposes, but also conveyed informative and attractive geographic visualisation. Other similar examples include Emoto, a projection augmented sculpture of Twitter data collected during one day of the Olympics, in which the most interesting themes are projected as heat maps (see Figure 2.4(a)), and Season in Review, a physical chart application integrated with an iPad for showing baseball statistics, with each chart corresponding to an option in the app (see Figure 2.4(b)).

Shape-changing User Interfaces

Although combining fabricated objects with AR technology can support dynamic data visualisation, this method cannot be regarded as totally dynamic data physicalisation, because the physical parts are still static and users can only experience the changes visually. A more advanced way is using shape-changing user interfaces. The shape of physical objects can be changed in various ways such as orientation, form or texture [39]. Therefore, shape-changing interfaces have a great potential to be used in dynamic data physicalisation applications.

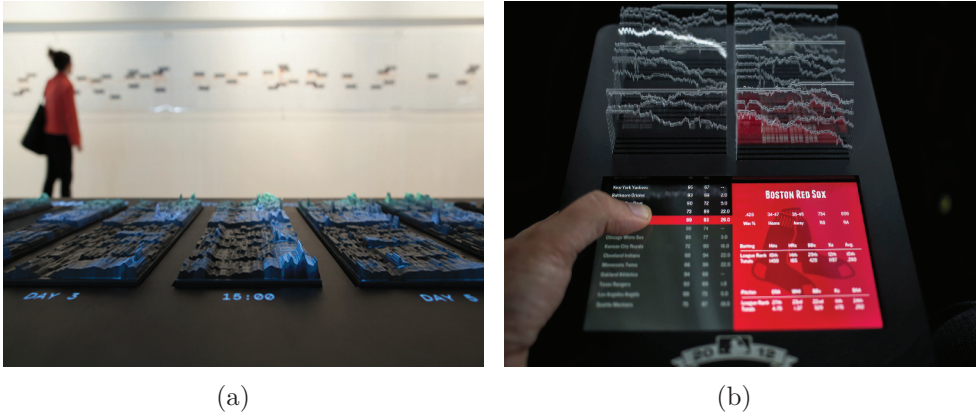


Figure 2.4: Examples of dynamic data physicalisation: (a) Emoto created by Moritz Stefaner, Drew Hemment and Studio NAND; (b) Season in Review created by Teehan+Lax labs.

Changing object shapes can be achieved by using special materials. For instance, Follmer et al. [10] proposed a particle jamming system which can control material stiffness and provide both soft and rigid material states. PneuUI [52] is another enabling technology for shape-changing interface with the use of pneumatically-actuated soft composite materials. This technology allows deformation on curvature, volume and texture, and can be used as both input sensing and dynamic output. Niiyama et al. [32] designed a device that took advantage of injection or removal of liquid metal, in order to transfer mass and realise weight and volume changes. All these examples, however, are restricted by the development of material science.

Making use of an array of actuated rods is another typical method to change shape output dynamically. FEELEX [20] is the first to use motorised pins to construct a spatially continuous surface for users to get haptic feedback when touching an image. Lumen [35] is another early interactive display consisting of an array of movable illuminated rods. When it senses user hand shapes or finger motions, different images or forms can be created by changing colour and height. Relief [26, 27] goes a step further by allowing free-hand gesture input and direct touch input. Later examples such as inFORM [11] and TRANSFORM [18] support other new interactions, including object actuation and remote gestural control. Compared with them, Sublimate [25] combines shape displays with augmented reality and users can switch between visual and physical representations. A more recent application TiltStacks [47] supports not only z-dimensional actuation but also tilting movement in an attempt to create a more powerful display surface.

Linear Actuation

Inspired by the applications of actuated rods, a similar linear actuation approach can be applied to build dynamic physical bar charts. Indeed, physical bar charts are one of the most commonly used visualisation method to transform abstract numbers to informative graphical pictures. The Physical Interaction Lab¹¹ created a dynamic bar chart called Wable (see Figure 2.5(a)) to show web behaviours. Each bar corresponds to one of the web services from RSS feeds, and it will arise when there is a specific web activity. The Digit London company¹² built Poly (see Figure 2.5(b)), a larger self-actuated bar chart encoding online poll data, which can fold or unfold to show a real-time voting result in a straightforward way. Tinker London¹³ made a physical bar graph Centograph (see Figure 2.5(c)) to show keyword popularity. When a user does a keyword query on a desktop computer, the graph displays how often that word was used in news articles over the past century.



Figure 2.5: Examples of dynamic physical bar charts: (a) Wable: Web behaviour shown with a dynamic bar chart; (b) Poly: Online poll data shown with a self-actuated bar chart; (c) Centograph: A physical bar graph shows keyword popularity.

More recently, Taher et al. [45] developed EMERGE, a dynamic physical bar chart to display international export data. The interface has a 10×10 self-actuating rods with RGB LEDs, so as to enable shape and colour output. Besides, a projector is used to show additional information on the table surface. Their work showed that rendering data visualisation in the physical world can provide an informative way for users to carry out interaction tasks, with a higher efficiency and better experience than an on-screen approach.

¹¹<http://www.physicalinteractionlab.com> (accessed 27/3/2020)

¹²<http://littlemachines.digitlondon.com> (accessed 27/3/2020)

¹³<http://tinkerlondon.com> (accessed 27/3/2020)

2.3 Other Techniques for Exploring Datasets

As the main objective of Phixel is to facilitate the exploration to datasets, the author also reviewed some techniques other than on-screen data visualisation for the same purpose.

When exploring large datasets, moving the visualisation from small computer screens to larger displays is a promising way to address the problems of limited pixels. Experiments have suggested that a large display can improve the efficiency of performing tasks [53] and boost sensemaking [2]. This technique is particularly useful in the cosmology domain [31], and its integration with touch-ability and hand gesture recognition is a powerful aid in large datasets comprehension [49].

Another helpful technology is to move the visualisation to an immersive virtual reality (VR) environment. In general, users have to wear a VR headset to simulate a virtual surrounding, which makes them feel like walking through the datasets [8, 29]. Similar but more advanced, Tangible Holograms [43] is used in a mixed reality (MR) environment. It is also worth noting that Tangible Holograms is a dynamic data physicalisation application, as it provides physical feedback (shape, texture or temperature) to augment data changes. Figure 2.6 displays some examples talked above.

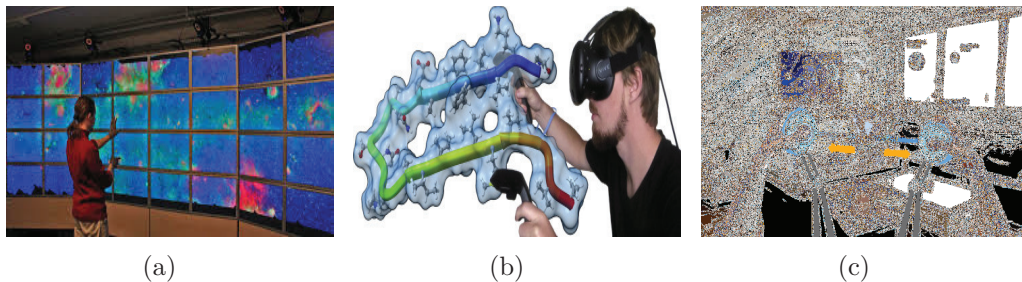


Figure 2.6: Examples of other techniques for exploring datasets: (a) Large high-resolution displays for showing galaxy data [31]; (b) A user explores biological data in a VR environment [29]; (c) Tangible Holograms: virtual objects are augmented by physical variables.

2.4 Approaches for Exploring Datasets about Food

As mentioned in the previous section, a dataset about food is used to investigate the application performance. The reason why this specific dataset is

chosen is because food is closely related to everyone’s life, and a better understanding of food nutrition can greatly help people live in a healthy lifestyle. In addition, experts or researchers who get deeper insights into food datasets can propose better diet suggestions.

So far there is no food-related application in the field of data physicalisation. When talking about food dataset visualisation, the most common or even the only example is the food guide pyramid¹⁴, which only roughly suggests dietary patterns for normal people. However, if people want to learn more about food nutrition, or researchers want to obtain detailed information of some food groups, this visualisation pattern is not enough.

A more pleasant way of telling food nutrition information is to use a wearable or mobile device to help recognise food and calculate nutrition data. Mango [50] is a mobile assistant for providing healthy eating habits and raising food awareness. It works by capturing a picture of food and responding with the degree of intake accordingly. Other mobile applications [24, 37] not only recognise the name of the food, but also estimate its volume for more accurate calculations. More recently, Jiang et al. [23] designed a food visualisation for Google glass¹⁵ to help consumers make nice shopping choices in real scenario (e.g. supermarkets).

2.5 Thesis Motivation

Through the review of representative studies in relevant fields, the work in this thesis is highly motivated in the following perspectives:

- Previous data physicalisation work for dataset analysis is mainly static, which cannot receive as exciting capabilities as data visualisation, because the latter can be updated and modified easily. Phixel should be dynamic and achieve as many functionalities as possible.
- Although there are existing dynamic physical bar charts to facilitate the communication of data or the exploration of datasets, they are limited to a specific data source. Apart from that, these user interfaces are usually fixed somewhere, and users cannot easily move them to other places for personal use. Phixel can be used for physical data visualisation of any kind of datasets. It also outperforms previous applications with respect to mobility and portability.

¹⁴<https://www.fns.usda.gov/mypyramid-graphics> (accessed 29/4/2020)

¹⁵<https://www.google.com/glass/start> (accessed 29/4/2020)

- There are no previous experiences on applying data physicalisation to the food domain. A meaningful use case study is conducted to see if Phixel can improve users' efficiency in retrieving nutrition information.

3

Solution

Nowadays, digital user interfaces (DUIs) are still the mainstream when building interactive applications, as digital objects are easy to create, duplicate, modify and delete. In contrast, dynamically changing physical characteristics of a physical artefact (such as texture, shape and position) cannot be as simple as changing a virtual one. Although Ishii et al. [17] put forward the huge development space for Perfect Red, a programmable matter whose physical properties can be updated according to user input or autonomous sensing, the vision of Radical Atoms cannot be fully realised in the short term because of the limitations on material science. Actually, most of the interactive physical data representations are complemented with TUIs (e.g. Emoto¹ and Virtual Gravity²) or DUIs (e.g. Relief [27] and Season in Review³). The work described in this thesis also combines TUI, DUI with physical data representation (shape-changing display) to develop an interactive and dynamic data physicalisation application.

¹<http://www.nand.io/visualisation/emoto-installation> (accessed 27/3/2020)

²<http://www.silkehilsing.de/?projects=virtual-gravity> (accessed 27/3/2020)

³<http://labs.teehanlax.com/project/season-in-review> (accessed 27/3/2020)

3.1 Design Idea

Phixel serves for users who want to explore a dataset in a natural way. Users first choose a dataset and all the columns in this dataset will be shown on the tablet. Then, they can select the items they would like to see and the digital information is encoded into tangible tags or cards. Finally, they place the tangible objects on specific readers to interact with the physical data representation, which will render a 3D physical bar chart of the corresponding data. If users remove a tangible tag or card from a reader, the relevant bars will go down accordingly. When there are a number of pre-encoded tangible objects, users can skip the first two steps and only interact with the physical interface. However, they can always have a look at the tablet, because detailed values are shown there simultaneously.

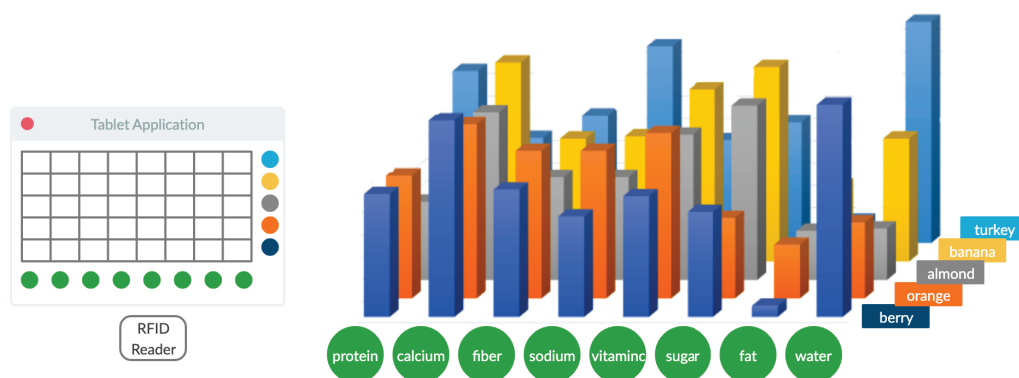


Figure 3.1: Concept sketch of the application

The system consists of three main parts: a tablet (digital user interface), a shape-changing display (physical user interface), and some tangible objects (tangible user interface) act as the bridge between these two interfaces. The physical user interface is made up of 14 RFID readers and a 5×8 array of physical bars (each bar has an LED strip inside). Figure 3.1 is a concept sketch of the application, which takes a food dataset as an example. The physical bar chart on the right displays eight kinds of nutritional information of five foods, and the tablet on the left shows their corresponding values. Each physical bar is illuminated by an LED strip and the position of the LED can be used to indicate additional information (e.g. while the height of a bar represents the maximum value of a food group, its average value can be highlighted by a colourful LED).

Phixel outperforms previous interactive physical interfaces in the following aspects:

- **It is customisable.** The application is not limited to a specific dataset. Instead, it supports customised datasets and users can select any dataset they need to explore.
- **It is flexible.** The application offers a digital display and a physical representation, but users are not forced to interact with both of them. On the contrary, they can choose to use the physical bar chart exclusively. In this case, either the input or the output is physical.
- **It is totally dynamic.** Both the tangible objects and shape-changing display are dynamic. The data encoded in a tag or a card can be modified dynamically. Also, the physical bar chart can be updated in real time when there are tangible objects approaching or leaving the range of the RFID readers.
- **It is portable.** The use of a tablet allows the users to move freely, as long as the connection between the tablet and the physical bar chart is ensured. The installation is not fixed, so users do not need to adjust their positions or operations based on some devices (like a camera or a projector in previous work).

3.2 Architecture and Functionalities

The basic structure of the application is pretty simple. As shown in Figure 3.2, an Android tablet acts as the digital user interface, on which users choose a dataset and some items to be encoded. All the datasets are stored in an SQLite server for accurate and fast querying. The digital user interface communicates with the control firmware (Arduino microcontrollers) through Bluetooth, and then the firmware drives relevant physical bars to present a visualisation.

As discussed in previous sections, Phixel is specially designed for exploring datasets. Therefore, it supports some of the basic data analysis tasks. Heer and Shneiderman [16] defined a taxonomy of visual data analysis tasks and the thesis follows a similar manner to describe the dataset explorations that Phixel can do:

- **Filtering.** A filtering task is a data hiding or refining task based on users' input. In the Phixel system, one data item can be encoded into

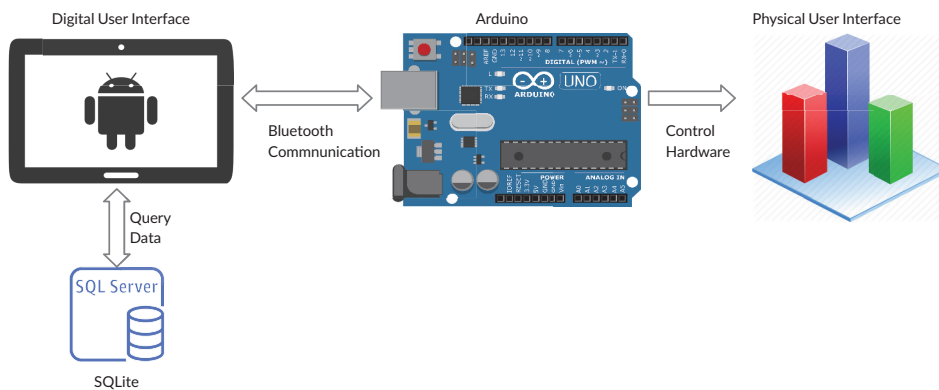


Figure 3.2: Architecture of the application

a tangible artefact, and users just put artefacts on readers to render a dataset on the physical display. Users can remove one or more artefacts to filter the dataset. The target of this task is to hide irrelevant data for a better reading and comparison experience. Phixel provides 40 pixels which are enough for users to easily compare the differences of multiple aspects between items.

- **Organisation.** The physical bar chart is not restricted by 5 rows and 8 columns. A tangible tag or card can be placed either in a row or a column to rearrange the result. Users can also adjust the order of tags to organise a dataset on the physical display.

4

Implementation

Essentially, the Phixel system is a dynamic bar chart for displaying dataset information physically. First, some hardware needs to be set up to support data display and update. Second, a software model has to be built to control the hardware and to manage the communication between them. Finally, a software application should also be implemented to do interactive tasks with users, such as choosing datasets, encoding virtual data into tangible objects, as well as showing some important information about the dataset. The implementation of the Phixel system strictly follows this process, and this chapter elaborates how Phixel works from two aspects: hardware and software.

4.1 Hardware

The hardware of Phixel is made up of two parts: hardware for DUI and for PUI. The former just requires an Android device (either a tablet or a phone), while the latter is relatively complicated, because it needs not only several electronic components, but also some customised 3D printed items. The electronic components include some actuation devices, lighting devices as well as a number of tag readers. To control these hardware, two Arduino¹

¹<https://www.arduino.cc> (accessed 19/5/2020)

microcontrollers are used. They are also responsible for the Bluetooth communication between the DUI and the PUI. Table 4.1 shows the electronic hardware requirements.

UI	Name	Quantity	Description
DUI	MI Pad	1	Android tablet
PUI	Arduino Mega 2560	1	Arduino microcontroller
	Arduino Uno	1	Arduino microcontroller
	HC-05	1	Arduino Bluetooth module
	MCP23017	10	I2C expander
	MFRC522	14	RFID reader/writer
	Mifare Card	some	RFID card
	28BYJ-48	40	Stepper motor
	ULN2003	40	Driver for stepper motor
	WS2812B RGB Strip	40	Full colour LED strip with IC chip (each strip is 12cm long)
	MB-102 Breadboard + Power Module	some	Construction base for prototyping of electronics, the power module provides both 5V and 3.3V supply
Jump Wires	some	Connecting hardware	

Table 4.1: Hardware requirements

4.1.1 Hardware Setup

The physical user interface takes tangible objects as input and a shape changing display as output. All the circuit diagrams of how these hardware are connected can be found in Appendix A.1.

There are 13 RFID readers for reading the data encoded in RFID cards and 1 specific reader/writer for writing data to tangible objects. Regarding the shape changing display, it consists of a 5×8 array of physical pixels, each of which is actuated by a stepper motor and illuminated by a full colour LED strip. The linkage of a motor to a pixel is a set of 3D printed thing, which mainly includes a gear and a rod with serrated edge. When placing an available card on a reader, the reader gets the value and makes the corresponding row or column of motors rotate, thereby driving the rods up. On the contrary, the rods go down if the reader detects nothing. The travel length of a rod is 120mm. Figure 4.4 shows the general hardware setup. While Fig-

Figure 4.1(a) is the side view of the setup of the Phixel system, Figure 4.1(b) is the top view which shows the position of the 13 RFID readers.

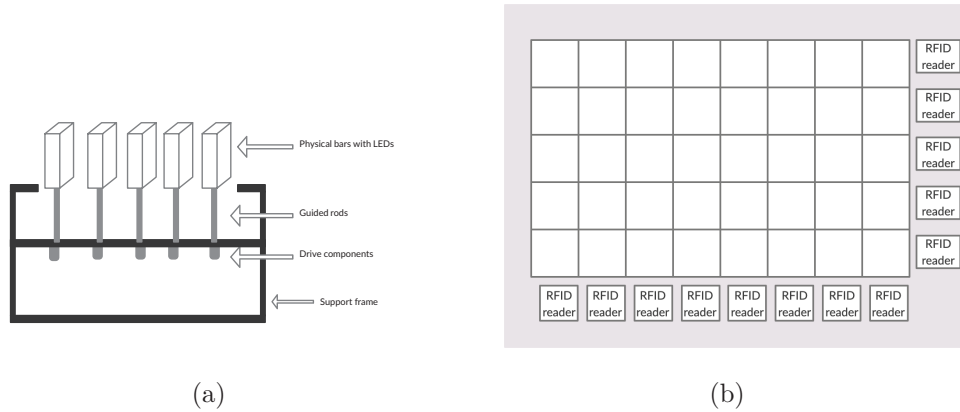


Figure 4.1: The setup of Phixel: (a) Side view; (b) Top view.

4.1.2 Android Device

The digital interface of the Phixel system is an Android device: MI Pad. It is a tablet that features a touchscreen for daily entertainment. Any other Android tablet or mobile phone can be used for Phixel as long as it meets the requirements of the operating system and supports Bluetooth. Figure 4.2 gives other parameters of MI Pad.


Xiaomi MI Pad		Model	A0101
	Operating System	Android v4.4	
	Processor	NVIDIA Tegra K1 Quad Core 2.2GHz ARM Cortex-A15	
	RAM	2GB	
	Internal Memory	16GB	
	Display	7.9"	
	Bluetooth	Bluetooth 4.0	

Figure 4.2: MI Pad parameters

4.1.3 Arduino Microcontrollers

Arduino boards are widely used for building interactive projects, thanks to their low price, extensible hardware, and cross-platform flexibility. They are

capable of conducting reading inputs from sensors, buttons or online data frames, and writing to an output such as making a sound, lighting up an LED, or activating a motor. All this work is done by the microcontroller on the board, which receives instructions from users.

The Phixel system has two Arduino boards: 1 Arduino Uno and 1 Arduino Mega 2560. The Mega 2560 is the main board which controls the Bluetooth module, 32 stepper motors, 32 LED strips and 14 RFID readers/writers, while the Uno manages the remaining 8 stepper motors and 8 LED strips. Although this configuration is not organised well enough that it does not control a series of identical hardware with one board, it is an effective and cost-efficient way to do the implementation.

The Arduino Uno is an entry-level but the most popular board among the Arduino products. It is made up of a microcontroller based on ATmega328P, 14 general-purpose input/output (GPIO) pins, 6 analog pins, and 1 serial port. Although the Uno is sufficient for most of the electronic projects, the number of GPIO pins is small and cannot support the connection of a large number of devices. The Arduino Mega 2560 is an enhanced board with a microcontroller based on ATmega2560, 54 GPIO pins, 16 analog pins, and 4 serial ports. The Mega 2560 is compatible with most shields for the Uno and thus these two boards can work together in harmony. Figure 4.3 shows the configuration of the two different boards.

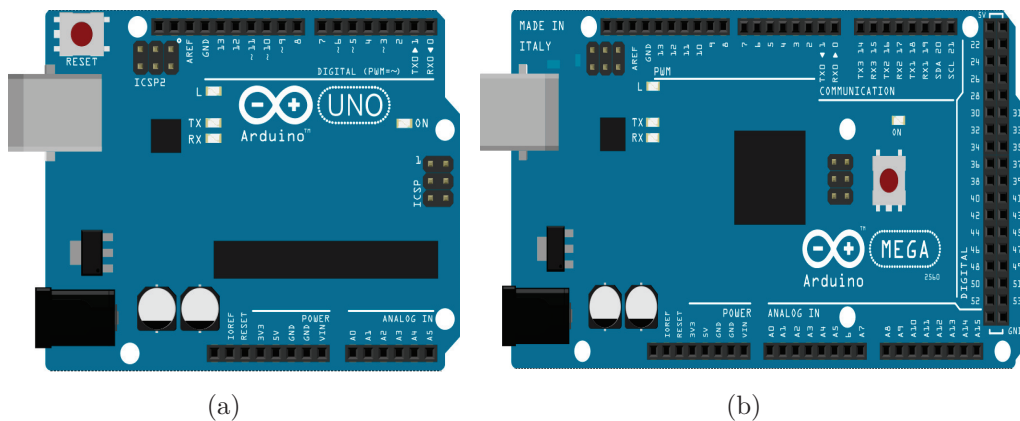


Figure 4.3: Configuration of Arduino boards: (a) Arduino Uno; (b) Arduino Mega 2560.

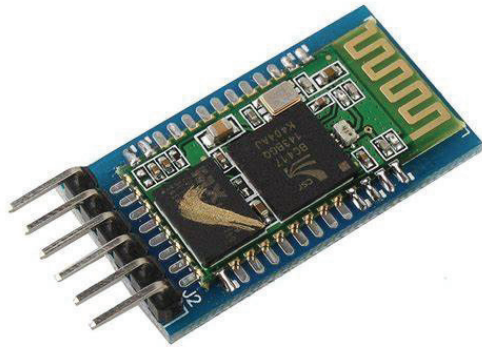
These two Arduino boards communicate with each other through the built-in serial port. As mentioned above, the Uno has only one serial port: pin0 (RX) for receiving data and pin1 (TX) for sending data. They can be connected to any serial port on the Mega 2560 for the transmission of serial

data. Their communication follows the Universal Asynchronous Receiver/-Transmitter (UART) protocol, which means that at any given time, only one of the Arduinos can send bytes to the other. However, this fact does not affect the system's performance because the transmission speed is quite high.

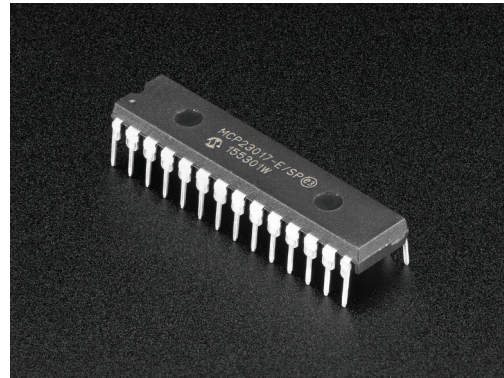
4.1.4 Bluetooth Module

In order to establish a connection between the Android tablet and the Arduino boards, a Bluetooth module called HC-05 (see Figure 4.4(a)) is used. The HC-05 also follows the UART protocol, so the wiring is almost the same. Normally, the HC-05 works at a 9600 baud rate and offers fast speed for short distances (less than 10 meters).

The HC-05 has two modes: command mode and data mode, and a key button is used to toggle between them. If a HC-05 is in command mode, its settings (such as device name and password) can be modified when getting commands at 38400 baud rate. On the other hand, the HC-05 acts as a slave for communicating data with a master board in data mode. In the Phixel system, the HC-05 is wired to the Mega 2560, and is paired with the Android tablet using the default password 1234.



(a)



(b)

Figure 4.4: Some hardware of Phixel: (a) HC-05 Bluetooth module; (b) MCP23017 microchip.

4.1.5 Port Expander

As described before, the Arduino Mega 2560 has 54 GPIO pins, but it still does not meet the requirements of wiring all the hardware needed in the Phixel system. Although using more Arduino boards can solve this problem, it is not a productive solution with respect to its cost. Alternatively, one can adopt some port expanders to extend the Arduino board's GPIO pins. In the Phixel system, the microchip MCP23017 (see Figure 4.4(b)) is used for this purpose.

The MCP23017 is a serial port expander and that can add up to 16 digital pins to one Arduino board. They are connected by the Inter-Integrated Circuit (I²C), which is a two-wire interface with the ability to connect a master to multiple slaves. In this way, the Arduino board only needs to provide two wires (a data wire SDA and a clock wire SCL) to control multiple expanders. The I²C protocol supports addressing so that the Arduino board knows which slave to “talk” to. The MCP23017 has three hardware address pins A0, A1 and A2, and therefore one I²C bus can connect up to 8 MCP23017 expanders, and can add an extra 128 GPIO pins to the Arduino board. Each of the address pins can be wired to either GND or 5V to alter the address (see Table 4.2 for details).

A0	A1	A2	Address
GND	GND	GND	0X20
5V	GND	GND	0X21
GND	5V	GND	0X22
5V	5V	GND	0X23
GND	GND	5V	0X24
5V	GND	5V	0X25
GND	5V	5V	0X26
5V	5V	5V	0X27

Table 4.2: MCP23017 addresses

4.1.6 RFID Technology

The Phixel system uses physical objects to represent digital data, and reading from or writing to those objects constantly happens during the process. This work can easily be done by the Radio-Frequency Identification (RFID) technology.

The RFID technology consists of two main components: a reader and an object attached with a tag which can be identified. Both of them have an

antenna to generate an electromagnetic field for receiving or transmitting a signal. The MFRC522 RFID reader/writer module (see Figure 4.5) is a good choice for most of RFID-based Arduino projects, due to its low price and low power consumption. The Phixel system also uses this module for encoding data onto tangible cards/tags, and for reading information from them.

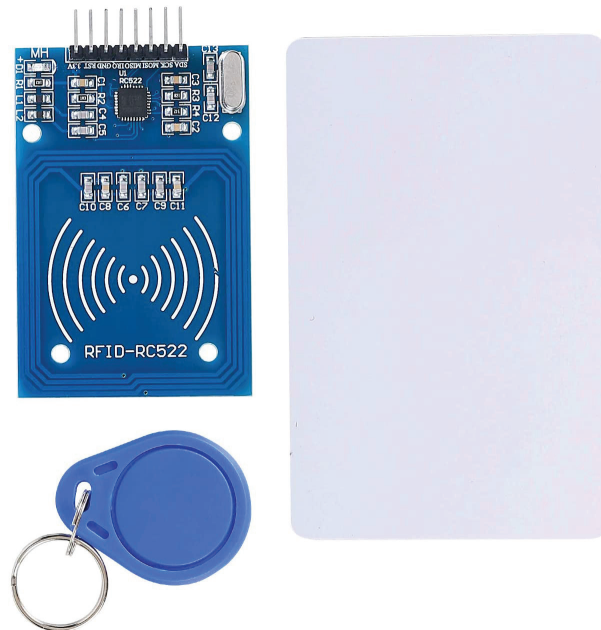


Figure 4.5: MFRC522 RFID module

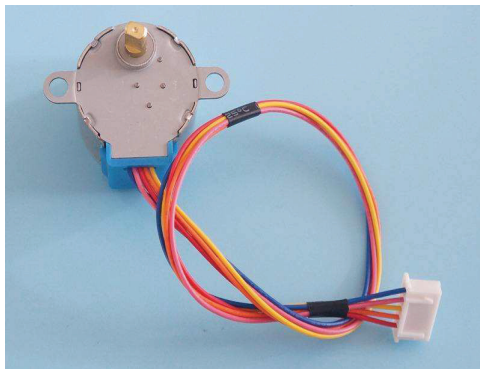
The MFRC522 module communicates with microcontrollers over a Serial Peripheral Interface (SPI). The SPI protocol is slightly more complicated than the previous two protocols (UART and I²C), as it is a four-wire interface and has to follow a master-slave mode. The four wires of SPI are as follows:

- **MOSI**: Master Out Slave In. The data transmits from master to slave.
- **MISO**: Master In Slave Out. The data transmits from slave to master.
- **SCK**: Serial Clock. A clock line defines the transmission speed.
- **SS**: Slave Select. A select line for the master to choose which slave to communicate with.

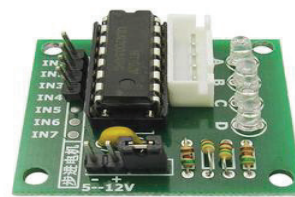
In the Phixel system, all 14 RFID readers are wired to the Mega 2560. They can work together by sharing the same MOSI, MISO, SCK, and using different SS to indicate their address.

4.1.7 Stepper Motor and Driver

Stepper motors are used in a wide variety of Arduino projects such as 3D printers or DVD drives. The biggest advantage of stepper motors is that they can rotate in precise steps. Therefore, if one requires a very accurate movement, a stepper motor is pretty powerful. The Phixel system uses the 28BYJ-48 Unipolar Stepper with the driver ULN2003 (see Figure 4.6) to drive the movement of physical rods.



(a)



(b)

Figure 4.6: Stepper motor and driver module: (a) 28BYJ-48 stepper motor; (b) ULN2003 driver.

The 28BYJ-48 is a five-wire stepper motor that moves 4096 steps to finish one rotation. Normally, it has to be packaged with a ULN2003 driver board. The ULN2003 has four input pins: IN1, IN2, IN3 and IN4. By setting each pin low (0) or high (1), the four coils in the motor get on or off. The stepper motor rotates by switching the coils on or off in a sequence, which creates a magnetic area to rotate the metallic part.

There are 40 stepper motors needed in the Phixel system, which requires 160 GPIO pins for connection. All these pins are wired to MCP23017 expanders to save the pins on the Arduino boards. One MCP23017 can connect four stepper motors, so there are 10 expanders used for these stepper motors, with 8 controlled by the Mega 2560 and 2 controlled by one Uno.

4.1.8 LEDs

As mentioned in the previous section, every pixel is light up by an LED strip in order to offer additional information via the colour. For example, while the height of a pixel represents the maximum value of a set of data,

the brightness and position of an LED can convey its average value. Phixel utilises the WS2812B addressable RGB LED strip to do this work.

As the name suggests, the LED strip is constructed by WS2812B LEDs in series. The WS2812B has a built-in integrated circuit and allows communication through one wire. Only one digital pin is needed to control an LED strip and each LED in the strip can have individual brightness and colour. There are 40 LED strips in total, so 40 GPIO pins are needed to manage the lighting. Phixel uses 8 digital pins in the Uno and 32 digital pins in the Mega 2560 to do this work.

The length of each LED strip is 120mm, because they are put inside some hollow bars and have to fit within them. The density of the strip is 144 LEDs per meter. Figure 4.7 shows some LED strips which are already cut.

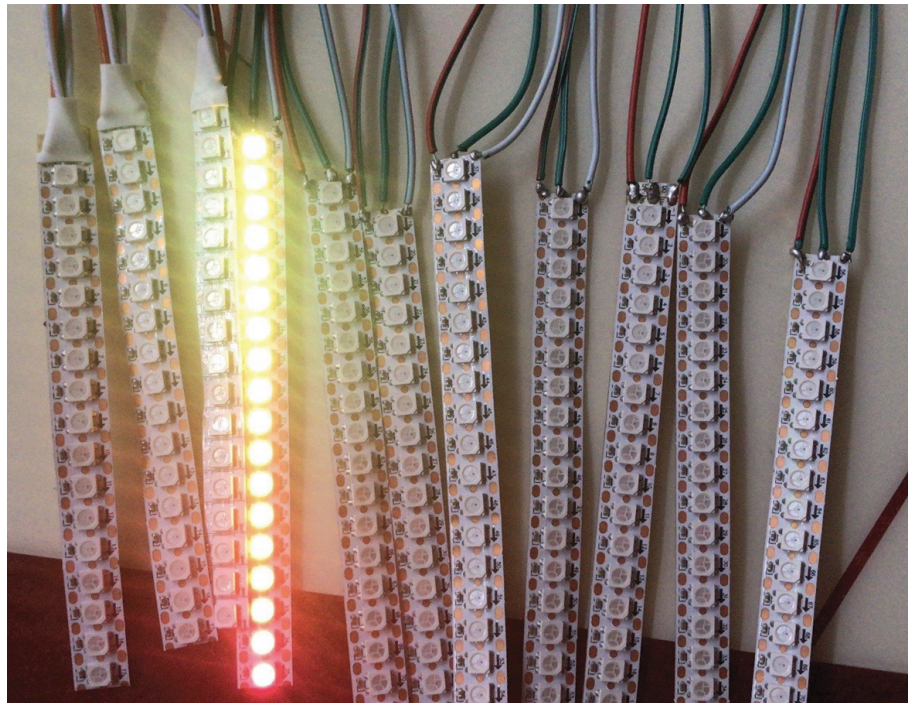


Figure 4.7: RGB LED strips

4.1.9 3D Printed Items

The stepper motor can only rotate clockwise or counterclockwise. However, the pixels have to go up or down to complete the dynamic update of the physical bar chart. Therefore, a linkage has to be built to turn the rotation into a linear motion. Fortunately, 3D printing provides a quick and easy

way to fabricate the customised linkage items. This fabrication is finished by an Ultimaker S5 3D printer in PLA plastic. The Ultimaker S5 has a large build volume ($330\times 240\times 300\text{mm}$) so it can print multiple linkage items at the same time. Besides, its layer resolution reaches down to 20 micron for making an object with very high resolution precision. However, there is a trade-off between resolution and time. For the 3D printing of Phixel, the author therefore did not use the highest resolution in order to reduce the printing time.

The linkage consists of three items: a shell mounted on the stepper motor to hold the other items together, a gear set on the stepper motor, and a rod with serrated edge that matched with the gear. They are assembled well to control the raising or lowering of a pixel. This idea follows a design on Thingiverse², differing from it on the pattern of the guided rod. Specifically, there is a smooth 120mm bar attached to the top of the guided rod, which is designed to host the LED strip. Figure 4.8 shows the comparison between the source file from Thingiverse and the final printed objects.

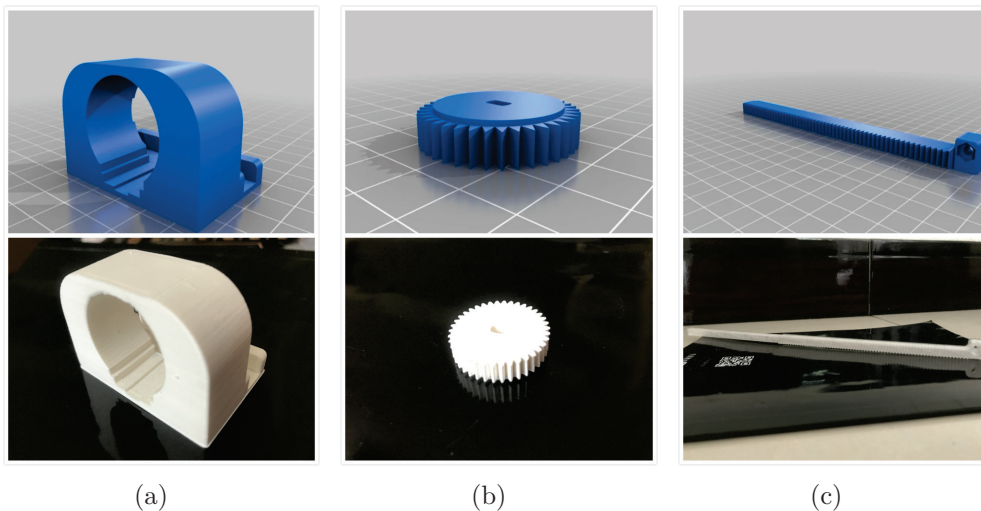


Figure 4.8: The 3D printed linkage objects: (a) A shell for the stepper motor; (b) A gear; (c) A guided rod

Another object that needs to be fabricated by the Ultimaker S5 is the hollow bar, which is the cover of a pixel and made of a kind of semi-transparent material. The size of the bar is $25\times 120\times 25\text{mm}$, and its thickness is 1mm. Figure 4.9 shows some examples of the hollow bars and Figure 4.10 illustrates how a pixel looks like when all 3D printed items are assembled.

²<https://www.thingiverse.com/thing:3593641> (accessed 25/2/2020)

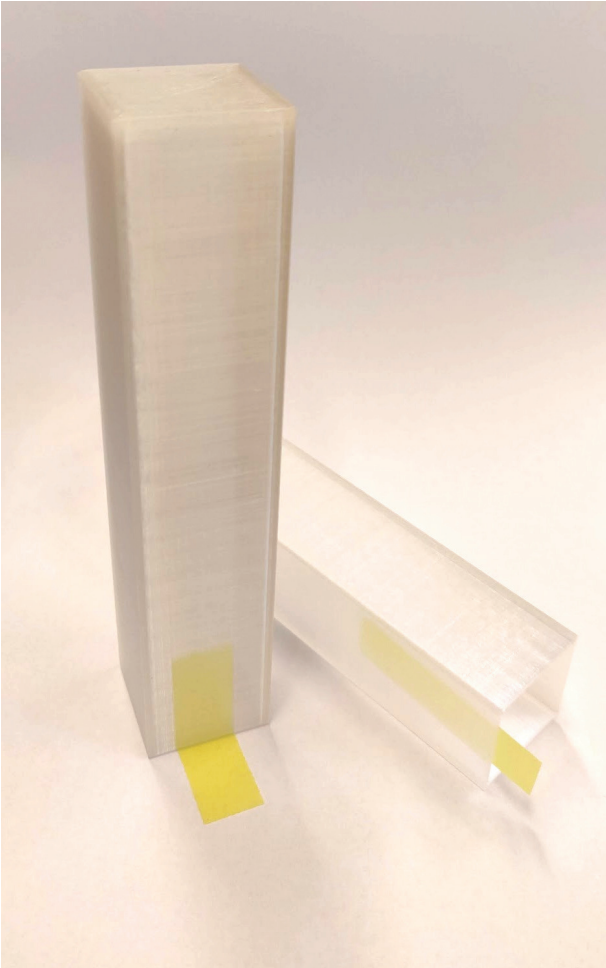


Figure 4.9: 3D printed hollow bar

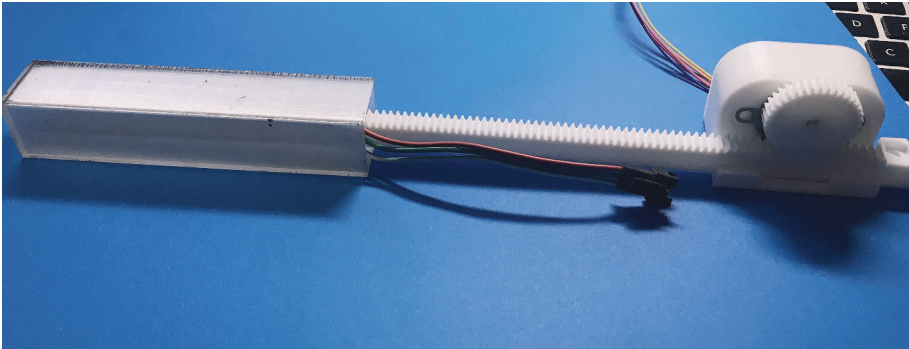


Figure 4.10: Final assembled 3D printed objects

4.2 Software

The software architecture of the Phixel system consists of three parts: the application, the model and the firmware. The Android application is used for conducting a user study, which is written in XML and provides several screens for users to do digital interactive tasks. The model (Phixel API) handles Bluetooth communication between the digital interface and the physical display, and manages queries on datasets. The firmware controls operations on all hardware, including RFID reading and writing, stepper motor rotation, as well as LED lighting. Figure 4.11 is the software stack of the Phixel system, showing its logical layers and their composition. While the first two layers are implemented in Android Studio³, the firmware layer is programmed in Arduino IDE⁴. This section describes how the software components work together, and details all the key libraries used for realising the necessary functionalities.

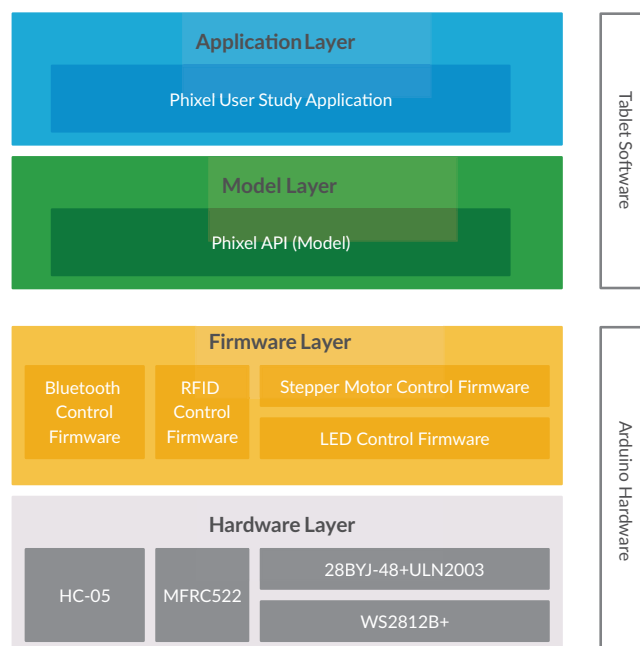


Figure 4.11: Phixel software stack

³<https://developer.android.google.cn/studio> (accessed 19/5/2020)

⁴<https://www.arduino.cc/en/Main/Software> (accessed 19/5/2020)

4.2.1 Android Studio

Android Studio is an official IDE for developing Android applications. It is based on IntelliJ IDEA, so the Java Development Kit (JDK) and the Java Runtime Environment should be installed beforehand. The Android Studio version that the author used is 3.6.2, and the JDK version is 1.8.0.

An Android project mainly constitutes of activities and layouts associated with them. In addition, every Android application has a `Manifest.xml` file to store the content of the application and to provide essential information about it. An activity is a class which represents a single screen of an Android application. Each activity has to be declared in the `Manifest.xml` file. Like the `main()` method in C++ or Java, an activity needs `onCreate()` to initiate and load when the application launches. A layout decides how the user interface looks like by defining the structure of the views (such as buttons, text boxes or widgets), and is associated with an activity through the method `setContentView()`. The activities are written in Java and layout files are written in XML.

Bluetooth Connection

The first functionality of the Android application is to establish a Bluetooth communication between the software and the hardware. The permission to use Bluetooth needs to be declared in the `Manifest.xml`. Then, the Bluetooth operations are written in the `MainActivity.java` class because this activity runs automatically when the application launches. The establishment of a Bluetooth connection is realised by the following steps:

- Step 1: Check the Bluetooth state of the Android device

Android Studio provides a `BluetoothAdapter` class to perform Bluetooth related activities. By using the method `getDefaultAdapter()`, it returns the default Bluetooth adapter of a device. In case the device has no Bluetooth adapter, the method returns `null`, and the application reminds the user to change the device. If a Bluetooth adapter is available, the application checks its state and enables Bluetooth.

- Step 2: Enable Bluetooth

This step requires a communication from the Phixel application to the Bluetooth application of the Android device. Android uses `Intent` to pass information across applications. It is also used for transmission among activities in an application. The application first creates an `BluetoothAdapter.ACTION_REQUEST_ENABLE` intent, and then invokes

`startActivityForResult()` to perform the task which “tells” the device to enable Bluetooth, and also gets its execution result. Once the Bluetooth state is changed, a `BroadcastReceiver` is registered to respond to the broadcast message and to inform the user that Bluetooth is enabled.

- Step 3: Searching for Bluetooth devices

The application first makes itself discoverable to others. Similar to the previous step, by calling `startActivityForResult()` with the `ACTION_REQUEST_DISCOVERABLE` intent, the device’s discoverable mode is enabled. By default, the discoverability lasts for 120 seconds, but one can modify the duration by adding `EXTRA_DISCOVERABLE_DURATION` parameter. The application also registers a `BroadcastReceiver` to notify the user when the device can be discovered by other devices.

Searching for other nearby Bluetooth devices is quite simple. It is done by calling the `startDiscovery()` method of the Bluetooth adapter. Once the discovery finds a Bluetooth device, a `BroadcastReceiver` is registered to add this device to an arraylist. Then, this arraylist is used to initiate a `DeviceListAdapter` instance. The `DeviceListAdapter` is an `ArrayAdapter` class that is created to store the names and MAC addresses of all the discovered Bluetooth devices. Finally, a listview adopts this adapter to show the information of other Bluetooth devices.

- Step 4: Pair with the Arduino

When the discovered Bluetooth devices are listed on the screen, users can click on one item to pair the Android device with it. The click event is handled by the `onItemClick()` function, and the bonding is simply realised by `createBond()` method of the `BluetoothDevice` class. Similarly, the application will inform the user when the device is paired with another one. Only when the Android device is paired with the Arduino, a continue button will appear at the bottom of the screen to allow the next operation. This first screen of the DUI is shown in Figure 4.12.

Dataset Selection and Data Encoding

One of the advantages of Phixel is that it allows users to select a dataset to freely explore. In this section, a dataset about food (see Figure 4.13) is selected to illustrate how this functionality is realised, and how to encode the dataset information into tangible objects. The food dataset used in

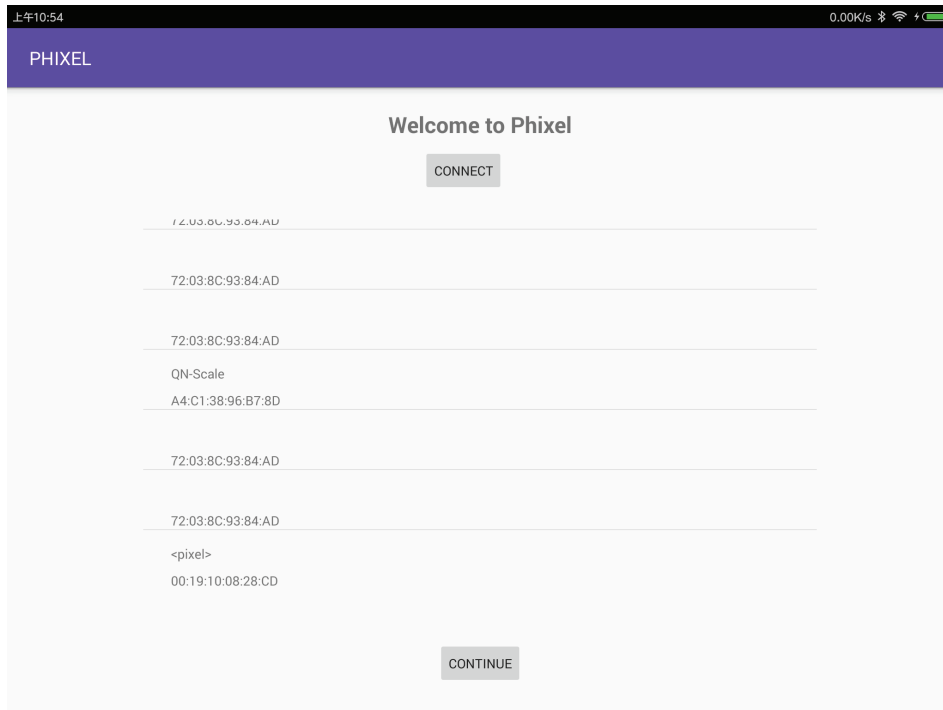


Figure 4.12: The first screen of the DUI showing the Bluetooth connection

the application can be found on USDA⁵. It covers 7637 different foods and separates them into 25 groups. Other than that, it contains 14 nutrients of foods. There is a visualisation of this dataset using parallel coordinates⁶, and this application refers to some operations on the dataset in this visualisation.

Due to the fact that there are many queries to the dataset, the csv file is stored in a database, so as to simplify the code and improve the query efficiency. Android has a built-in SQLite database, which means that it can be accessed without configuring any kind of connections. To create a table to store the content of the food dataset, Android provides `SQLiteDatabase` class and its method `execSQL()` can be used to execute any operation on the database. After the table is created, the application opens the csv file and reads its content line by line, and then inserts them into the table. Once this job is finished, the interface will display all the columns of the dataset for users to do the subsequent operations.

Taking the food dataset as an example, the interface shows its 16 columns (see Figure 4.14) as buttons. If the user clicks on the button of a food name or food group, a new dialog will be opened to show all the food names or

⁵<https://www.usda.gov> (accessed 5/1/2020)

⁶<http://bl.ocks.org/syntagmatic/raw/3150059> (accessed 10/1/2020)

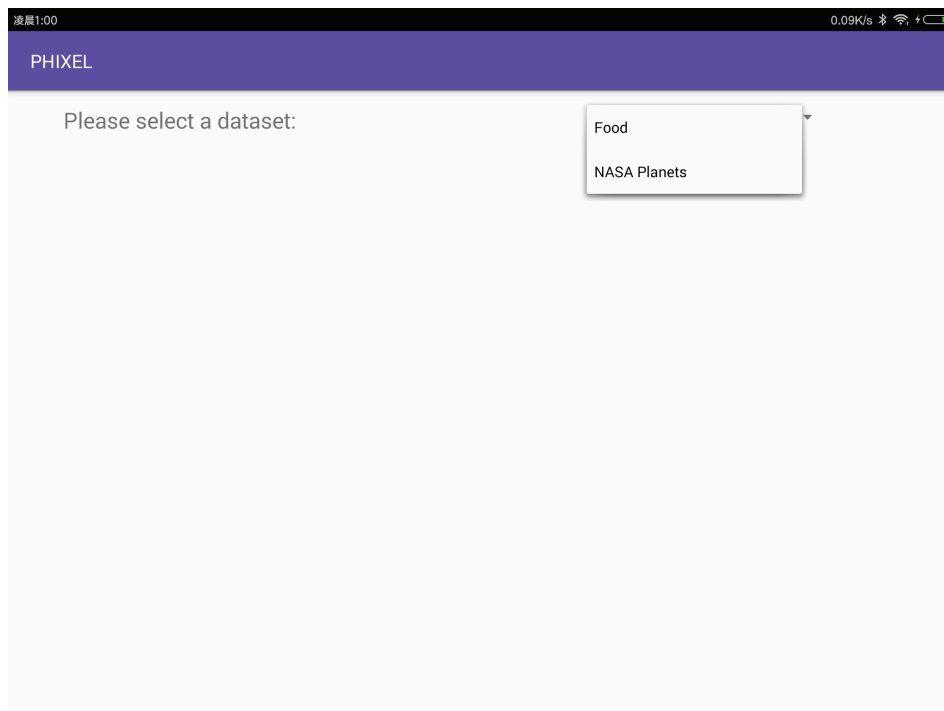


Figure 4.13: The second screen of the DUI showing the dataset selection

food groups, respectively. The dialog has a search box on the top, and users can type some keywords to filter the list and quickly choose what they want. This functionality is pretty useful when there are a large number of items in the list. For example, there are 7637 food names and it is not user-friendly to force users to keep scrolling down to find a specific food. Figure 4.15 shows how the dialog looks like after clicking on the food name button.

When a user selects a food name, a food group or a nutrient, the application sends the corresponding message to the Arduino through Bluetooth. The Arduino receives the message and writes it to an RFID tag, thus the digital data is encoded into a tangible object.

To make the data transmission work through Bluetooth, a channel has to be constructed in advance. First, the application gets the remote Bluetooth device (the Arduino) using its MAC address. Then, it creates a `BluetoothSocket` object to start an outgoing connection to the Arduino Bluetooth device by calling `CreateRfcommSocketToServiceRecord()`. This method requires a Universal Unique Identifier (UUID) as an argument, which is a 128-bit number that uniquely identifies the Bluetooth channel between the two devices. The UUID can be generated randomly, and there are lots of examples online.

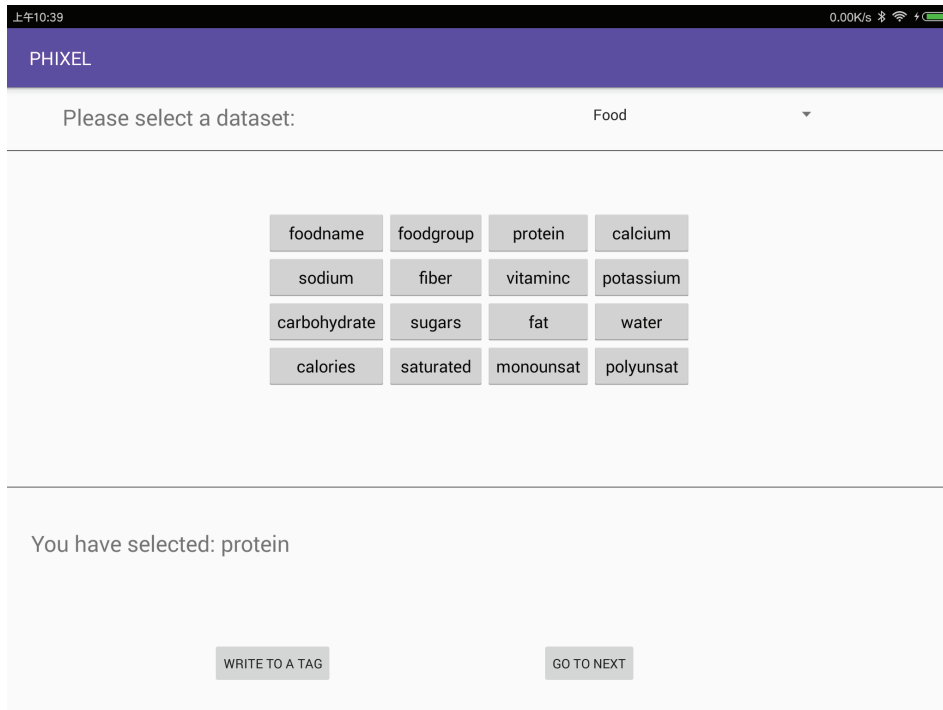


Figure 4.14: The second screen of the DUI showing the dataset columns

Once the Bluetooth communication is established, the application can easily send message by writing data bytes to the `OutputStream` and receive messages by reading data bytes from the `InputStream`. An important point is that when encoding a food name or a food group, the length of the message to be sent is different from the one when encoding a nutrient. Specifically, if one wants to use a tag to represent a nutrient, they click on the nutrient button and the application will send the column index of the nutrient in the dataset (a Hex number from 2 to F), followed by the # symbol. Encoding a food name or a food group is more complicated: The application first queries all the information of the food, including the average and maximum value of the 14 nutrients it contains, and stores these 28 values in a string, separated by the # symbol. Apart from that, this string also contains an index 0 or 1 to indicate the column of food name or food group, and the ID of the encoded item. The ID of a specific food is a five-digit number, while the ID of a food group is 0 to 24 as there are 25 groups in total. In this way, the Android application does not need to query the database when using the physical interface, thus making the PUI work independently and efficiently. The implementation on how to process the message and write it to RFID tags will be elaborated in the Arduino IDE section.

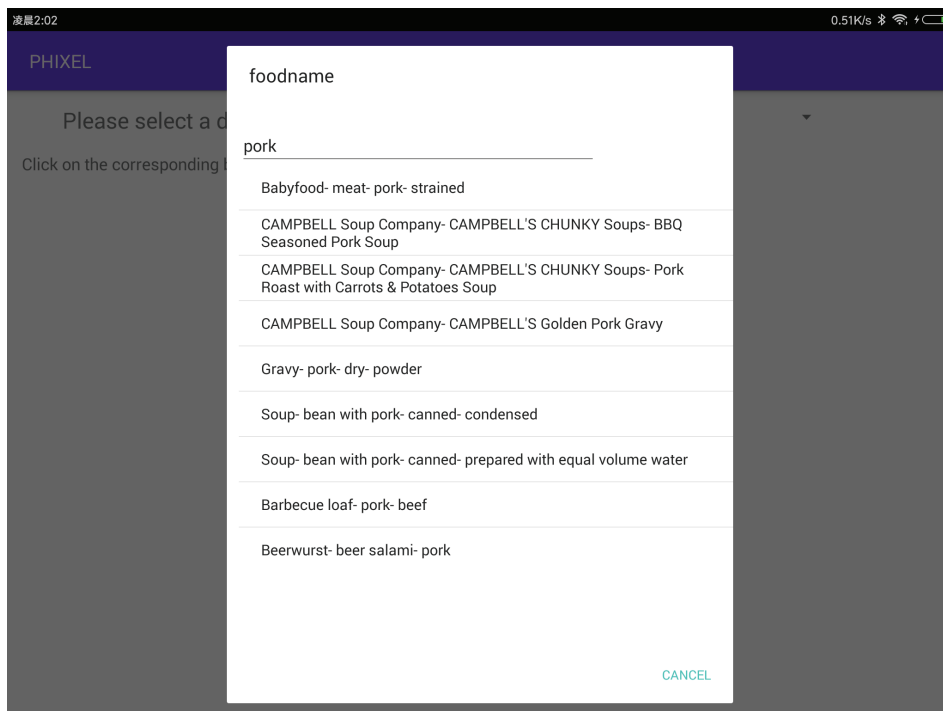


Figure 4.15: The second screen of the DUI showing the dialog

Display PUI state

The last functionality of the Android application is to show the PUI state. This screen consists of a 5×8 table and 13 text views. Actually, this view is just a visual version of the PUI, with the table showing the statistics of the physical bar chart and the text views showing the information encoded in the tag at the corresponding position.

The implementation of this functionality is similar to the data encoding. The Android application receives a string consisting of 4 numbers which are separated by the # symbol through Bluetooth, and then sets the colour and text of a widget accordingly. The received string starts with either 0 or 1, which means that the widget to be modified is the table or the text view, respectively. The second value in the string is an index to indicate a specific table grid or a text view. If it is the case of modifying a grid, the last two numbers in the string are the average and maximum value of an item. On the other hand, if the string is for modifying a text view, the third value represents the column index of the food dataset, and the fourth value is the ID of a food or food group, or 0 when the index points to a nutrient. Figure 4.16 is an example of this screen.

Item	Protein	Calcium	Fibre	Sodium	Vitamins	Sugar	Fat	Water
TURKEY	27.37g	28g	0g	78g	0g	0g	10.4g	62.07g
BANANA	1.09g	5g	2.6g	1g	0.0087g	12.23g	0.33g	74.91g
ALMOND	21.22g	264g	12.2g	1g	0g	3.89g	48.42g	4.7g
ORANGE	0.7g	43g	2.4g	0g	45g	9.14g	0.21g	87.14g
STRAWBERRY	0.58g	21g	5.4g	37g	37g	0g	0.6g	80.66g

Figure 4.16: The third screen of the DUI showing the PUI state

4.2.2 Arduino IDE

The Arduino software (IDE) is an open source platform for writing code and then uploading it to the boards. The program written in the Arduino IDE is called sketch. Each sketch has two main functions: `setup()` and `loop()`. The `setup()` is used to initialise variables and do some work that needs to be done in advance, such as defining the pin mode or start a serial connection. This function is called when a sketch starts, and it only runs once if the user does not reset the board. The `loop()` function runs in every time step, and as the name suggests, the code in this function loops consecutively. The Arduino version that the author used is 1.8.11.

Bluetooth Connection

As mentioned before, the HC-05 Bluetooth module is wired to the serial port of the Arduino board, so the functions for `Serial` are used to do Bluetooth related operations. There are five main functions for the data transmission and reception through Bluetooth:

- `if (Serial)` or `while (!Serial)`: To check whether the serial port is ready. The latter is usually used in the `setup()` function to make the sketch wait until the serial port is connected.
- `Serial.begin()`: To start the serial data transmission. This function is put in the `setup()` function. It requires a speed or baud rate argument to set the data rate. Normally, the speed is 9600 baud. It should be noted that if the HC-05 works in the command mode, the speed has to be set to a 38400 baud rate.
- `Serial.available()`: To get the number of bytes which are already stored in the serial receive buffer. By default, the buffer holds 64 bytes. If the arrived data exceeds this value, the extra bytes will be discarded.
- `Serial.read()`: To read incoming serial data. This function only reads the first byte of incoming data, and removes this byte from the serial receive buffer to release some space. This is the method to receive data through Bluetooth.
- `Serial.write()`: To write binary data to the serial port. A more common usage is `Serial.print`, which writes characters to the serial port instead. This is the method to send data through Bluetooth.

RFID Reading and Writing

The sketch for reading from or writing to RFID tags uses the MFRC522 library⁷ which provides robust methods to do MFRC522 related work.

Before talking about the MFRC522 library, one should know some information about the RFID tag, namely the MIFARE card. The memory of the MIFARE card is 1 KB, and it is divided into 16 sectors, each of which is organised in 4 blocks. Each block can store 16 bytes of data. Figure 4.17 is a 3D representation of this structure. Block 0 is a special block because it contains the Unique Identifier (UID) of a tag and the manufacturer data, so it is better not to overwrite this block. Apart from that, the last block of each sector is used to store access bits which grants the read and write access to this sector. Therefore, only the first three blocks of each sector (excluding the first sector) can actually be used to store data. It is known that the message sent from the Android device has a maximum of 30 values (1 for the column index, 1 for the ID, 14 for the maximum and 14 for the average), so it takes 30 blocks to store these values at most. Hence, the available memory of a MIFARE card is enough for this purpose.

⁷<https://github.com/miguelbalboa/rfid> (accessed 8/3/2020)

In order to read from or write to a tag, a MFRC522 reader object has to be first initialised. The object takes two arguments: the slave select pin and the reset pin. As the reader is wired to the SPI bus, `SPI.begin()` should be called, and then the method `PCD_Init()` of the reader object can be invoked to initialise the MIFARE card.

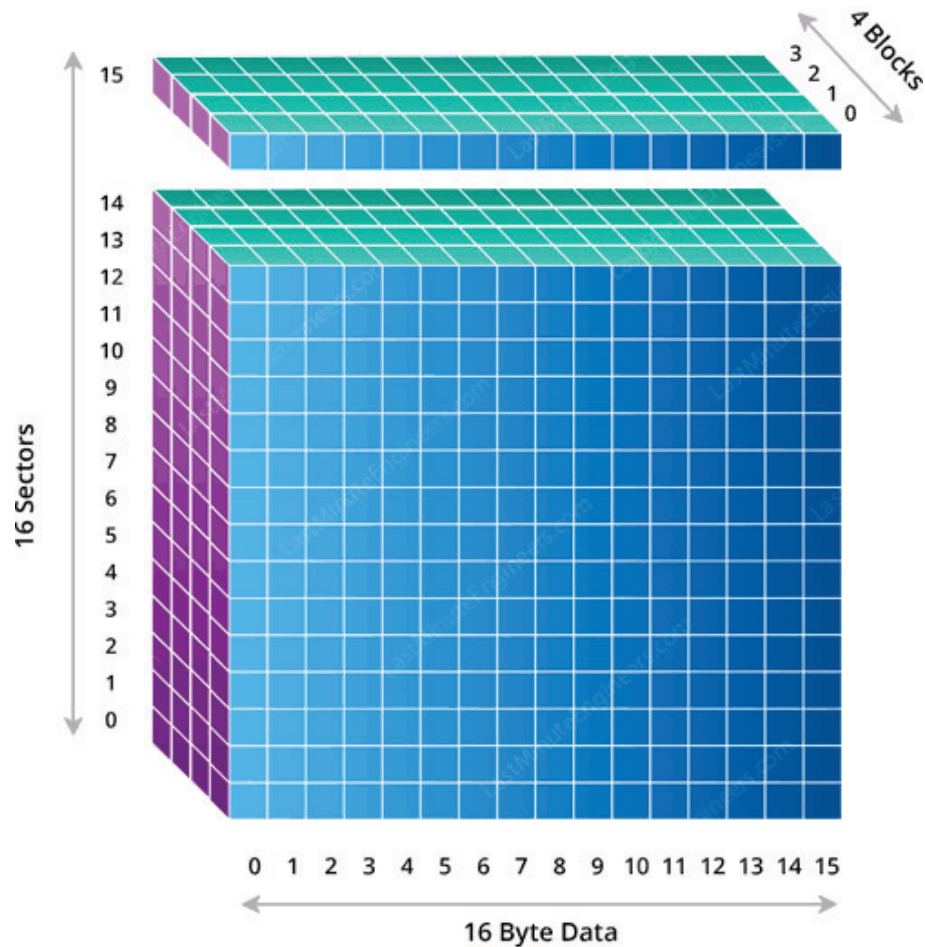


Figure 4.17: 3D representation of the MIFARE card memory (from Last Minute Engineers)

There are two very important methods provided by the MFRC522 library: `PICC_IsNewCardPresent()` and `PICC_ReadCardSerial()`. The former is to check if there is a card coming, while the latter verifies if the card is available for reading and writing. If both of them return a true value, the sketch can continue to the subsequent methods.

Writing data to a block uses the function `MIFARE_Write()`. This function has 3 arguments: the block number to be written, a byte array which stores the bytes of data, and the number of overwritten bits (max 16). Similarly, reading data from a block uses the function `MIFARE_Read()`, which also takes 3 arguments: the block number, a byte array for storing the block content, and a pointer to the read buffer size. It should be noted that the size of the byte array is at least 18 bytes to hold the block's 16 bytes, because this is the requirement of the `MIFARE_Read()` method. After every reading or writing, the RFID tag should be halted by calling `PICC_HaltA()` and the encryption should be stopped by calling `PCD_StopCrypto1()`, otherwise the reader will not recognise other tags.

Stepper Motor Driving

As the stepper motors are wired to the MCP23017 expander, a library named `Adafruit_MCP23017` is used. This library provides a `Adafruit_MCP23017` class, and each MCP23017 expander connected to the Arduino board can be initialised as a `Adafruit_MCP23017` object. Then, the expander can start the communication with the board by invoking `begin()` method. The method takes the address of the expander as an argument.

To rotate a stepper motor, one can set the four input pins of the ULN2003 in output mode by calling `pinMode()`, and then use `digitalWrite()` to set them to high (1) or low (0) in a certain sequence (see Table 4.3).

Direction	Pins	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7	Step 8
Clockwise	IN1	0	0	0	0	0	0	0	0
	IN2	0	0	0	1	1	1	0	0
	IN3	0	1	1	1	0	0	0	0
	IN4	1	1	0	0	0	0	0	1
Counterclockwise	IN1	1	1	1	0	0	0	0	0
	IN2	0	0	1	1	1	0	0	0
	IN3	0	0	0	0	1	1	1	0
	IN4	1	0	0	0	0	0	1	1

Table 4.3: Switching sequence of the stepper motor

The number of steps and the direction that a stepper motor rotates is determined by the RFID tags of the row and column where it is located. For example, if the first row reader is occupied by tag A which encodes all the nutrients of an apple, and tag B which represents the nutrient sugar is placed on the first column reader, the system will read the data from the block that stores the value of sugar in tag A, and drive the stepper motor to rotate the corresponding number of steps clockwise. On the contrary, if one of the tags is removed, the stepper motor will rotate counterclockwise

to the origin. An experiment has been done to find that it takes 625 steps to move a pixel to its highest point (corresponds to 1000g or 1000kcal in the food dataset). Equation 4.1 shows the relation between the step that a motor rotates and a numerical value in the food dataset, where s means steps, n means a numerical value and max means the maximum value of a column. Taking the food dataset as an example, the maximum value of each column equals to 1000, namely $max = 1000$. Therefore, the number of steps that a stepper motor needs to rotate is calculated according to Equation 4.2.

$$s = \lfloor \frac{625n}{max} \rfloor \quad (4.1)$$

$$s = \lfloor \frac{n}{1.6} \rfloor \quad (4.2)$$

LEDs

The implementation of the LEDs uses the `FastLED` library⁸. At the beginning of the sketch, the data pin and the number of LEDs should be declared. Then in the setup function, the methods `FastLED.addLeds()` and `FastLED.setBrightness()` are invoked to initialise an LED strip and set its brightness. Each LED strip is a `CRGB` array, with a size of the number of LEDs.

The library also provides several pre-configured palettes (also known as colour lookup table) to set the colour and brightness of a single LED. Normally, a palette has 256 entries, each of which contains a specific 24-bit RGB colour. One can simply retrieve a colour through a 8-bit index (0-255). The palette used in the Phixel system is `RainbowColors_p`, and the blending method for generating the final colour is `LINEARBLEND`. The colour of one LED is set by calling `ColorFromPalette()` method, which takes palette name, palette index, brightness and blending method as arguments.

In the Phixel system, an LED strip has 17 LEDs, each of which can represent a range of values in the food dataset. Specifically, the LED index i represents the range $[60i, 60(i + 1) - 1]$.

When a pixel is used to show a nutrient of a food group, the brightest LED in the pixel indicates the average. The position of the LED depends not only on the average value, but also on the maximum value (the height of the pixel). Equation 4.3 and 4.4 show how the position i_1 is calculated, where avg is the average value of a food group, and max is the maximum

⁸<https://github.com/FastLED/FastLED> (accessed 8/3/2020)

value of it.

$$i_0 = \begin{cases} 0, & \text{if } avg \in [0, 59] \\ 1, & \text{if } avg \in [60, 119] \\ \vdots & \\ 16, & \text{if } avg \in [960, 1020] \end{cases} \quad (4.3)$$

$$i_1 = i_0 + \lfloor \frac{1000 - max}{60} \rfloor \quad (4.4)$$

By following all the implementations in this chapter, a dynamic physical bar chart was generated to display the visualisation of a subset of the food dataset (see Figure 4.18). It illustrates eight nutrients (protein, calcium, fibre, sodium, vitamin C, sugar, fat and water) of turkey and banana. The Android tablet displays their specific values in the corresponding grids.

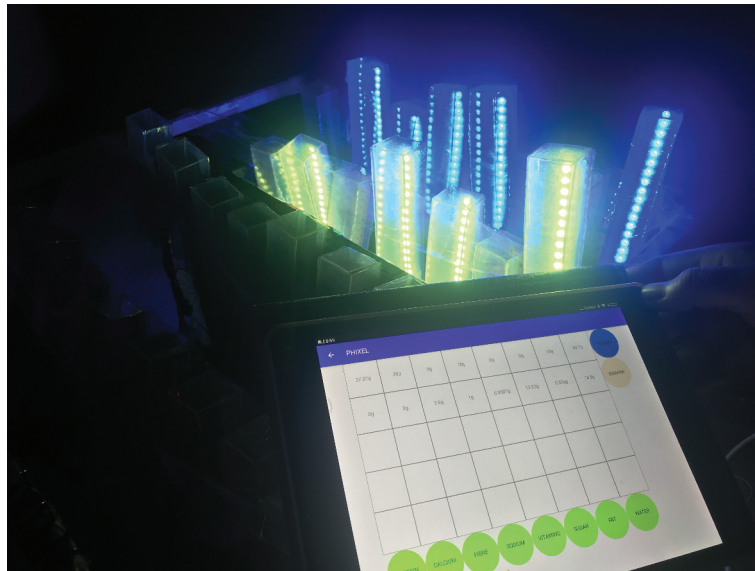


Figure 4.18: Phixel prototype

5

Evaluation

In order to find out how people interact with and what they think about Phixel, a user evaluation was conducted. It was carried out with 10 participants (5 females and 5 males) with a mean age of 28 years. None of them had previous experience in using dynamic data physicalisation applications. This chapter details the procedure of the user evaluation and analyses participants' performance in the evaluation and their feelings about the Phixel application.

5.1 Procedure

To finish the evaluation, participants had to go through three phases. First, they were individually introduced to the Phixel system through a short video, which provided a step by step tutorial on the usage of both the Android tablet and the physical display. A short pre-test (see Appendix A.2) was conducted afterwards to see how participants are familiar with the bar chart that represents a dataset. The target of this phase was to reduce the impact of participants' experience on the evaluation results. Then, participants were shown a food dataset, which was the same dataset as described in the implementation section. They were also asked to carry out a number of dataset exploration tasks pertaining to filtering and organisation using Phixel, and to answer some pre-defined questions regarding the dataset. The Bluetooth

connection had already been set up and all necessary tangible cards or tags had been prepared in advance, so that participants only needed to focus on the physical interface to do the exploration. During this process, participants' interaction behaviours were recorded for later analysis. Finally, participants completed a questionnaire about the usefulness and usability of Phixel, they were also encouraged to provide their experiences and opinions about the system.

5.1.1 Dataset Exploration

As described earlier, Phixel supports two dataset exploration tasks: filtering and organisation. To investigate whether Phixel is useful to help users explore and understand datasets, participants performed these tasks and finished a small test in relation to them. The test accuracy highly decided the usefulness of the Phixel system. Each task and its relevant questions are present in the following.

Filtering

When users wish to hide irrelevant data and only convey the information of interest, they perform a filtering task, which is a powerful aid in reading dataset values correctly [54]. Two subsets of the food dataset were used for this task. The first subset was eight nutritional information (protein, calcium, sodium, fibre, potassium, carbohydrate, fat and water) of cheese fondue, turkey, almond, bologna pork and raw banana. The second one was the same eight nutrients of five food groups, including baby foods, poultry products, breakfast cereals, pork products and beverages.

Participants took corresponding tags and positioned them on readers to render a subset of the food dataset. For the first subset, they were asked to find answers to the following questions:

- **Which of the five foods has the most protein?**

Participants observed the physical bar chart and found the column of protein. Then, they cleared the display of irrelevant bars by removing tags corresponding to columns which were deemed unneeded. When there were only pixels representing the protein of the five foods, participants found the highest pixel, whose corresponding row was the food that contains most protein.

- **Compared with turkey, does bologna pork contain more fat?**

This question required participants to compare the fat of turkey and that of bologna pork. Similar to the previous question, participants removed three tags except turkey and bologna pork in the row, and seven tags except fat in the column. The filtered result is only two pixels, and participants could easily check if the pixel of bologna pork is higher than the one of turkey.

The second subset showed nutrition of five food groups, with the height of each pixel representing the maximum value of each nutrient in one food group, and the position of an LED representing its average value. It was expected that filtering was more useful for the second subset than the first one, because the second physical bar chart displayed an additional dimension and was more complicated. Clearing unnecessary information could make the reading or comparison easier. Participants did the following work to answer questions:

- **Which food group has the lowest average carbohydrate?**

Participants removed seven tags except carbohydrate to hide unneeded columns. They compared the position of LED between the remaining five bars to find the lowest one.

- **Among baby foods, breakfast cereals and pork products, which food group has the lowest average fibre?**

In order to filter the fibre of the three food groups, participants removed two tags which represent poultry products and beverages respectively, and seven tags which represent nutrients other than fibre. After irrelevant pixels were hidden, participants found the pixel with the lowest LED position, whose corresponding row was the answer to the question.

Figure 5.1 shows the four filtering tasks performed by participants. It is obvious that the filtered results make the answers to the questions clear at a glance.

Organisation

Organisation makes the physical bar chart more flexible as it allows users to rearrange a dataset in the physical display. For instance, users might want to swap rows and columns according to their preferences, or they might want to move the last row to the first row that is closer to them. The dataset used for this task was also a subset of the food dataset, which contained two nutrients

(calories, protein) of turkey, bologna pork and beef. Tags representing foods were placed on row readers and those representing nutrients were placed on column readers. Participants were asked to perform two organisation tasks based on this display:

- **Move the rows closer to you**

The original physical bar chart was a 3×2 array of pixels. Participants took away the food tags on the column readers one by one and moved them to a new position that was closer to them. In this way, pixels were moved from the last three rows to the first ones, thus making the physical display closer to participants.

- **Sort the dataset by calories in descending order**

When working on a digital dataset, users can sort the dataset by selecting a column and the system does this job automatically. Phixel, however, requires users to manually operate on the physical interface to complete a sorting task. To organise the dataset in a new order, participants found the tag which corresponded to the highest pixel, and moved it to the first row. The other two tags followed the same manner. As seen in Figure 5.2(b), only the order of rows was changed, with tags representing nutrients remaining the same.

5.1.2 Questionnaire and Interview

After participants finished all their data analysis tasks, they were asked to fill in a questionnaire to gather insights on how people think about Phixel. The User Experience Questionnaire (UEQ)¹ was used for this purpose, which consists of 6 scales with 26 items. Each item is rated from 1 to 7, with a randomised order to avoid tendencies in the answer. The detailed structure of this questionnaire is as follows:

- **Attractiveness.** This scale has 6 items, they assess users' overall impression of the product.
- **Perspiciuity.** This scale has 4 items, they assess the learnability and usability of the product.
- **Efficiency.** This scale has 4 items, they assess if the product requires users to make unnecessary effort.

¹<https://www.ueq-online.org> (accessed 26/5/2020)

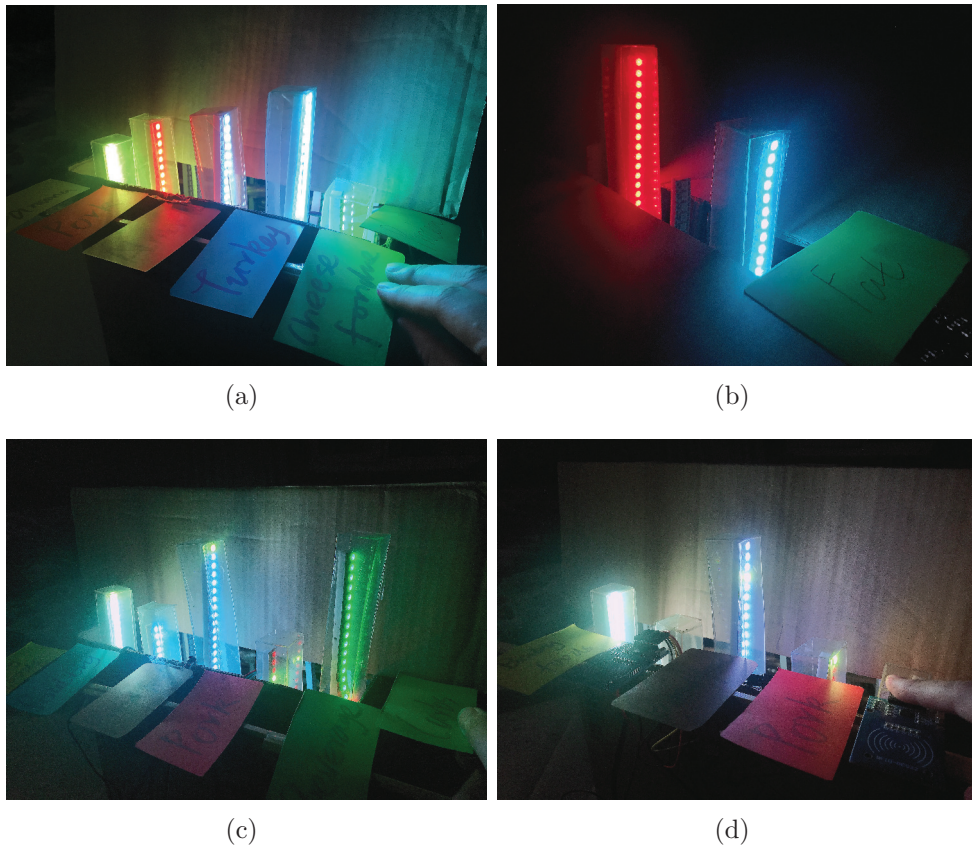


Figure 5.1: A participant performing some filtering tasks

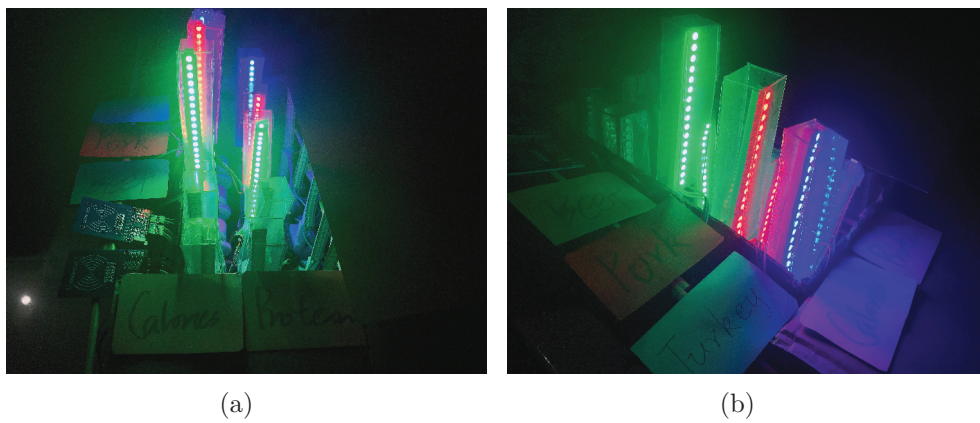


Figure 5.2: A participant performing some organisation tasks: (a) Original bar chart; (b) The result after performing two organisation tasks

- **Dependability.** This scale has 4 items, they assess the controllability of the product.
- **Stimulation.** This scale has 4 items, they assess if the product can motivate users to use it.
- **Novelty.** This scale has 4 items, they assess the creativity and innovativeness of the product.

Participants were also encouraged to take an interview to provide their suggestions or opinions on the Phixel system. Their feedback is discussed in the last chapter.

5.2 Results and Feedback

This section analyses the results of different phases of the user study. First, participants' experiences with bar charts are summarised. All of them have used bar charts before and they clearly know how to utilise this data visualisation technique to help understand a dataset. Second, some interesting observations on how participants interact with the Phixel application and their test accuracy are illustrated. Finally, the results of the questionnaire are quantitatively measured and their meanings are analysed and reported.

5.2.1 Pre-test Result

Figure 5.3 displays the results of each pre-test question. As seen in Figure 5.3(a), all participants have used bar charts previously. Among them, four participants use bar charts at least once a day, because they work in accounting or a financial related area, and have to deal with a large number of datasets daily. One of them mentioned that "*I really like to use a bar charts to visualise the complex numbers, it provides a clear way to compare the differences between several sets of data.*" Two participants use bar charts for presenting the performance of employees in monthly meetings in their companies. Another four participants just use bar charts sometimes. This result is consistent with participants' understanding of bar charts, that six participants who use bar charts often know a lot about this technique, and the rest just know the basic operations. Despite this difference, all participants understand how a bar chart works and how to make use of a bar chart to find answers to specific questions. Therefore, their performances in the following dataset exploration phase are not affected by their knowledge about bar charts.

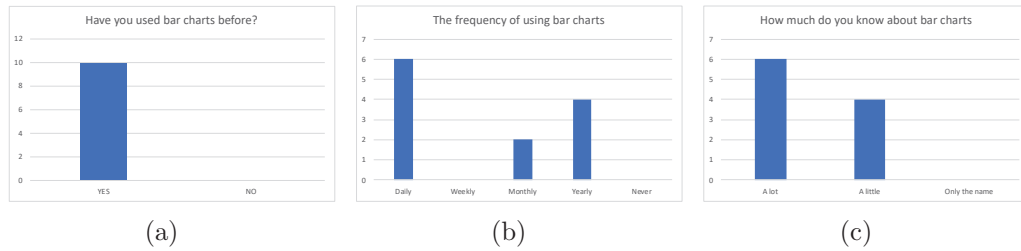


Figure 5.3: The results of pre-test: (a) Participants’ experiences with bar charts; (b) How often participants use bar charts; (c) Participants’ knowledge about bar charts

5.2.2 Dataset Exploration and Observation Result

All participants were able to answer the four questions in the filtering task and complete two organisation tasks correctly. This means that Phixel is useful in helping users understand datasets and perform data analysis tasks. Besides, some interesting observations on how participants interact with Phixel are described here to inspire possible improvements of Phixel.

Most participants showed great interest in this new form of data visualisation. They seemed excited when a pixel is going up or down. *“I was excited to make a pixel move just by placing two tags on the readers. I felt like I was totally controlling the bar chart by using Phixel instead of clicking the mouse multiple times to make the computer control my data. This procedure made me have a feeling of achievement.”* Another observation is that participants moved around the table to see some pixels that may be blocked by others. Sometimes they also touched or grasped a pixel to help confirm an answer.

The author also noticed some negative points of Phixel during the exploration. The first one is that it took a while to look for RFID cards needed from a bunch of cards, especially those representing food groups. Participants had to read the labels on the cards one by one to find what they wanted. *“I did not want to spend much time on finding cards, because the focus should be on the physical output. I suggest using more representative and attractive objects as an alternative to those flat cards.”* Indeed, the author also considered that some 3D printed customised objects (such as food models) will be used as input in the future.

In addition, two participants had different opinions when performing the first organisation task, that required them to move the rows closer. One of them said *“I understand the powerful ability that the physical display can be reorganised flexibly. But for this particular task, rotating the whole display is apparently a much more convenient way to reach the target.”* As the par-

participant said, Phixel is a portable product which can be moved or rotated to provide a better view for users, but the aim of this task is to show how the reorganisation of the physical bar chart is performed. Another participant thought Phixel was unfortunately unable to move the three rows as a whole, just like one can do in a screen-based visualisation. This valuable feedback is well adopted for further development of Phixel.

Moreover, a few participants showed impatience when they were waiting for the update of the physical bar chart because of its low speed. This limitation is discussed in the last chapter and a solution is proposed there.

5.2.3 Questionnaire Analysis

The original questionnaire results can be found in Appendix A.2. To measure the user's feedback quantitatively, a data analysis tool was used which was also provided by the UEQ. Figure 5.4 is the raw data of participants' feedback and Figure 5.5 shows the distribution of their answers per item. In general, most of participants had a positive opinion on each item in the questionnaire.

Items																										
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
7	6	5	1	1	6	6	6	6	2	3	6	1	5	6	7	6	1	2	1	7	2	7	1	1	2	7
5	4	2	3	3	4	5	7	3	1	5	3	4	5	6	5	4	4	4	6	1	5	1	3	3	3	6
6	5	3	3	2	6	6	6	3	2	6	2	6	6	5	5	3	3	3	5	3	5	3	3	3	3	5
5	4	1	3	2	5	5	5	3	1	6	1	2	5	6	6	1	2	3	6	3	6	1	2	2	2	6
7	5	1	2	4	6	6	6	2	1	6	1	6	6	4	6	1	2	1	6	1	6	1	1	1	1	6
7	7	1	2	1	6	7	5	4	2	6	2	4	5	6	6	2	3	3	5	3	5	1	2	1	1	6
6	6	1	2	1	6	5	7	5	2	7	1	5	6	6	6	2	1	2	4	1	6	2	2	2	2	6
4	7	5	3	4	5	5	5	2	4	4	1	5	5	4	5	2	4	4	7	2	6	1	3	3	3	3
5	4	3	4	3	4	5	5	4	3	5	3	4	5	6	5	3	4	4	5	1	6	3	3	3	3	5
5	4	2	4	4	5	5	3	4	3	4	2	3	6	5	3	2	2	2	6	3	4	2	2	2	2	5

Figure 5.4: Raw data of participants' feedback

To further calculate the mean value of participants' opinions on each aspect, the scale of each item was transformed to a range from -3 to +3. Thus, -3 represents the most negative answer, 0 a neutral answer, and +3 the most positive answer. The transformed result was further grouped to six aspects and the mean value was calculated, as shown in Figure 5.6. Only participant 8 gave a negative answer for the novelty of the Phixel product (-0.5), but the value was not strongly negative.

Figure 5.7 is the overall impression of Phixel by all the participants. Although the range of the scales is between -3 and +3, a real application cannot reach values above +2 or below -2, due to the reason that it is impossible to get extreme values after the calculation of means over a range of different

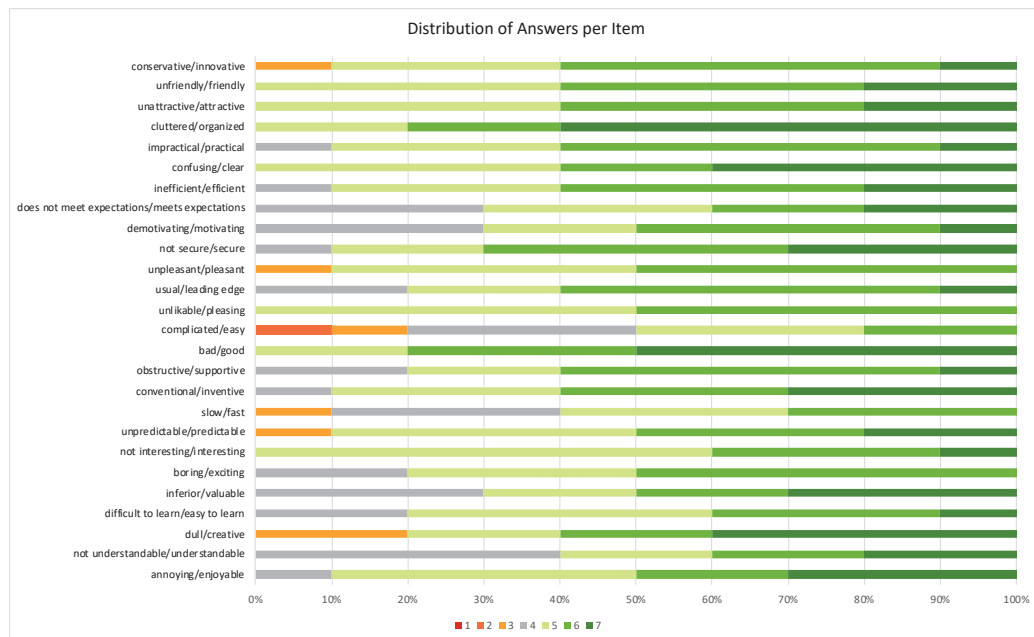


Figure 5.5: The distribution of participants' answers per item. Green colour indicates a positive opinion while red colour indicates a negative one. Grey colour means a neutral feedback.

Scale means per person						
Attractiveness	Perspicuity	Efficiency	Dependability	Stimulation	Novelty	
2,50	2,00	2,75	2,50	2,25	1,50	
1,00	1,00	1,75	1,00	0,50	2,25	
1,50	1,25	1,00	1,50	1,75	1,25	
1,83	0,00	2,00	1,75	1,50	2,50	
2,67	2,00	2,25	2,50	1,50	2,00	
2,17	1,50	1,25	1,50	2,25	2,25	
2,17	2,00	0,75	2,50	2,25	2,25	
1,17	1,75	2,50	0,75	0,50	-0,50	
1,00	0,75	1,00	0,75	0,50	1,25	
1,33	0,00	1,00	0,75	1,00	1,25	

Figure 5.6: The mean value of each scale per person

persons with different answer tendencies. Therefore, it can be stated that all scales of Phixel show an extremely positive evaluation.

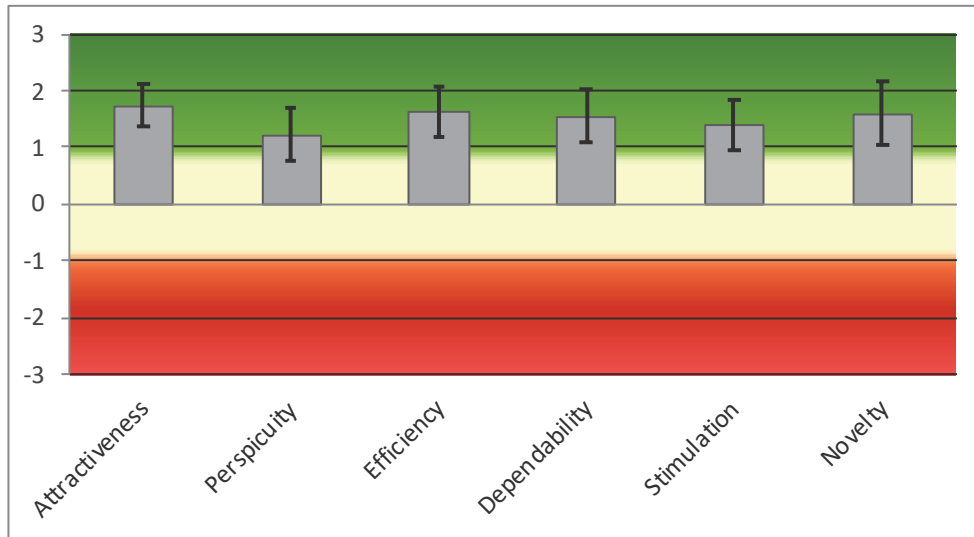


Figure 5.7: Means of participants' opinions

The separate result of how Phixel fulfils the general expectations concerning user experience is not enough. It is also necessary to compare the user experience of Phixel to the results of other established products. Fortunately, UEQ provides a benchmark dataset containing 452 different products. The result of the user experience quality of Phixel is shown in Figure 5.8. Obviously, the five scales (attractiveness, efficiency, dependability, stimulation, novelty) of Phixel outperform other existing products, while the perspicuity of Phixel is below average.

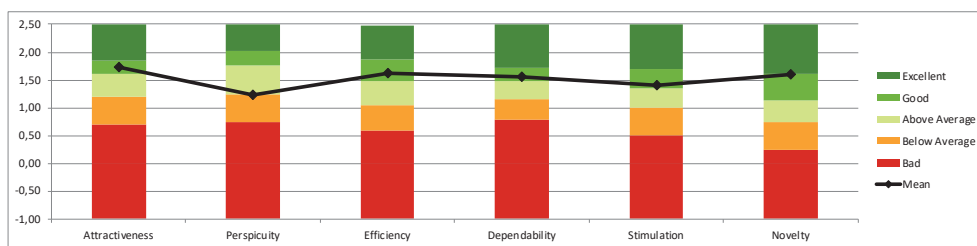


Figure 5.8: The benchmark graph for Phixel

To conclude, Phixel left a good impression on participants. They found the interactive physical interface was attractive and innovative. Participants just interacted with physical objects to get information in a digital dataset,

which made them feel like they are in absolute control of data. Although a few participants cost some time to learn and understand how Phixel works, all of them were motivated to use this product for dataset exploration in the future.

6

Use Cases

Phixel can be used for a variety of scenarios. This chapter presents two different use cases that illustrate the potential use of the Phixel application in supporting the exploration or understanding of datasets. In the first use case, Phixel is used to assist a researcher in filtering a complex dataset and preparing an engaging visualisation. In the second use case, Phixel serves an automobile company for the purpose of marketing analysis and discussion in the context of a group meeting. This scenario involves three datasets, and Phixel provides a convenient way to make an analysis across multiple datasets. It is assumed that the datasets required in both use cases have been uploaded in the Phixel system, so users just need to select the corresponding one to finish their work. These scenarios suggest that Phixel is a user-friendly interface for regular people who are not familiar with dataset cleaning or processing, and a powerful tool for performing basic data analysis tasks.

6.1 Food Nutrition Exploration Scenario

An expert in food nutrition is invited by their university to do a speech about healthy diet in front of some bachelor students. He is glad to accept the invitation and wishes to explore a food dataset to prepare the content of the speech in advance. The food dataset contains 14 nutrients for 25 food groups, covering 7637 foods in total. Considering the large amount and

multi-dimension of the dataset, it would cost much time to process it and find the information he needs, which is absolutely not what the expert wants. Therefore, he chooses Phixel to help him finish the exploration work.

According to the topic of the speech, the expert would like to talk about two food related subjects. The first one is about the food pyramid, which represents the optimal proportion of five food groups (grains, vegetables, fruits, dairy and egg products and meat) to be eaten each day. The expert uses five foods which are representative of each food group: a bread for grains, a lettuce for vegetables, an orange for fruits, a glass of milk for dairy and egg products, and a piece of pork for meat. Each food has an RFID tag attached for further reading and writing. He also prepares five RFID tags to represent the nutrient content of foods, including calories, protein, fat, carbohydrate and sodium, which are the five most common nutrients on a food package. Regarding to the second subject, the expert decides to discuss the impact of different cooking methods on health. He takes apples as an example, and prepares three foods which are made of apples: a raw apple, a box of apple juice and an apple pie. Similarly, each of the foods has a RFID card attached.

After all foods and tags are prepared, the expert opens the Phixel application on his Android phone and pairs his Bluetooth with the physical interface. Then, he selects the food dataset to get all columns in it. The expert clicks the button of the food group to open a new dialog which shows a list of food groups. He is too impatient to scroll up and down to find the grains item. Instead, he types a keyword in the search box at the top of the screen and easily finds what he wants. After he selects grains, the dialog is closed and a message is displayed at the bottom of the screen to tell him what he has selected. Afterwards, he places the bread on the RFID reader, and a button called "Write to tag" appears. He clicks the button to encode nutritional information of grains on the bread. A prompt message will be displayed on the screen to remind him the bread can be removed when the encoding step is done. The encoding of the other four foods which represent food groups and three apple related foods follows the exact same procedure. To encode a nutrient to a tag, the expert just puts the tag on the reader and selects the corresponding button of nutrients to complete the work.

On the day of the speech, the expert brings a case which carries the Phixel solution. He first introduces the five components of the food pyramid, and explains the meaning of each real food. In order to illustrate why the food pyramid is structured like that, the expert renders nutritional information of each food group in the physical bar chart by placing five nutrient tags on the column readers and five foods on the row readers. The generated

visualisation is a colourful physical bar chart showing the differences of five food groups with respect to their nutrients. Based on this bar chart, the expert can elaborate the correct number of servings that a person should eat to stay healthy. For instance, people should take grains as the major dietary source because they provide the most calories needed by the body. A moderate amount of meat and dairy products is also required as they contain enough protein to build the muscles. The expert sometimes moves or rotates Phixel to show a better and clearer picture for the audience.

To present the impact of different cooking methods on food nutrition, the expert needs a new physical visualisation. He removes all five foods from the row readers and puts three apple related foods on them. The updated bar chart shows that although these foods are all made of apple, the apple pie has much higher fat and calories than raw apples, and apple juice contains more sugar than raw apples. Besides, the other nutrients of apple pie and apple juice are lower than raw apples. According to the result, the expert recommends people to choose low-fat and low-sugar cooking methods to keep the good nutrients in food as much as possible. To strengthen the audience's awareness on healthy cooking methods, the expert rearranges the order of foods to make the most suggested cooking method display at the front and the worst one at the back.

6.2 Marketing Analysis Scenario

An automobile company decides to arrange a mid-year meeting to analyse the sales performance in the previous six months, in order to deploy a work plan for the next half of the year. The meeting involves the manager of the company, five sales representatives from different regions, and an accountant who manages the financial accounts of the company. They are responsible for monthly sales, sales of four models, and income statement, respectively.

A staff from the IT department is asked to prepare all the materials needed for the meeting. The staff opens the Phixel application on an Android tablet, and connects the software application with the physical interface. He selects the first dataset about monthly sales from a list of three datasets, and then eight buttons appear on the screen. The first one is a region button, and the other buttons represent the months from January to June, as well as the first half of the year. The staff puts an RFID card on the reader, clicks on the region button, and selects a specific region in the newly opened dialog to write this information to the tag. He does this process five times to encode all regions to five RFID tags, each of which has a distinct colour. Then, he clicks on the other buttons one by one to encode the information of

time on other RFID tags. After finishing the preparation of the first dataset, the staff selects the second dataset, which contains the sales of four models in five regions. The staff only needs to encode the model information as the encoding of regions has already been done. He prepares four vehicle models (sedan, wagon, van, coupe) to encode the data, each of which has an RFID tag inside. Finally, the staff moves to the last dataset. This dataset records the sales income and financial expense for each region. The staff writes them to a green tag and a red tag respectively.

The Phixel product is placed in the middle of the conference table, and participants sit around so that everyone can see the visualisation clearly. First, the manager puts the cards of regions on the row readers and those of time on the column readers, generating a 5×7 physical bar chart. The manager observes if the bar chart of each row follows the same trend to analyse the sales performance of each region accordingly. In addition, everyone can clearly see the region which performs best each month and the semi-annual sales champion from the physical graph.

After that, the five sales representatives are asked to introduce the sales results of the four models in their regions one by one. The first person removes all cards from column readers and puts four vehicle models on them. The person also removes the other four region cards except their own one, resulting in a 1×4 bar chart, based on which the sales differences of each model are compared, and the reasons are analysed. The other four sales representatives follow the same pattern to describe their sales performance. Finally, all region cards are put on row readers for a comparison of the overall accomplishment of the five regions.

The last task of the meeting is to analyse the sales income and marketing expenses for each region. The accountant clears all objects on column readers and replaces them with a green tag and a red tag. The generated bar chart consists of a 5×2 array of bars, with the first column coloured green and the second column coloured red. To find if there is a direct relation between sales performance and advertising expenses, the accountant then puts a card which represents the first half of the year on the third column reader. Then the participants can analyse the relation between the two items by observing the difference in height between the second and third column of each row.

7

Future Work

The Phixel system shows how one can take advantage of 3D printing and existing hardware to construct a dynamic data physicalisation application, which provides a good choice for users to explore a dataset in a natural way. Future Phixel system should allow users to upload their own datasets for more flexibility. Beside, this application has plenty of potential for further development of various aspects. This chapter suggests different possible improvements of the Phixel system in the future.

7.1 Including Personal Information

Previous chapters have elaborated the usage of Phixel for a food dataset. A first possibility of future development of the Phixel system might be a new design with respect to this dataset. For now, the Phixel system offers a dynamic physical bar chart for users to understand the nutrients of foods. In the future, it could be associated with personal information to provide a healthy diet suggestion. People can connect their own mobile devices to Phixel and get a customised healthy eating plate based on their profiles. When a user provides their age, height and weight, a row of pixels will go up to show the amount of each nutrient that one should eat to get the right balance. A camera and a pressure sensor can be used to recognise a real food and obtain its weight, so that Phixel can calculate the nutrients that actually

taken by the user. These values are displayed by another row of pixels for the user to compare with the standard row to see if they have ingested enough nutrients.

Apart from that, when a user mentions their disease or health situation (e.g. obesity, diabetes or pregnancy), the Phixel system will remind them when they take a “bad” food. For example, if a diabetic eats a cake which contains excessive sugar and fat, Phixel will make a warning beep to tell the user this is a “bad” food and do not take it. This development is useful for helping people develop healthy eating habits.

7.2 Including More Data Analysis Tasks

The main advantage of the Phixel system is that it can render a data visualisation in the physical world for conducting data analysis tasks. According to the taxonomy of visual data analysis tasks proposed by Heer and Shneiderman [16], the existing implementation only allows filtering and organisation tasks. Therefore, a second possibility of further developing the Phixel system is to make it support more tasks, namely annotation tasks and navigation tasks.

Annotation allows users to select or mark an individual data point in a dataset. In the future Phixel system, a single data point can be marked by emphasising the corresponding pixel. This work can be done by adding a contour light and a small button on the top of each pixel. When users press the button, a unique colour outlines the pixel and tells users the data point is selected. This functionality is pretty useful when users want to highlight a subsets of data or focus on a certain data point.

A navigation task is a view controlling task that is especially necessary when working with large datasets. If the user works on a digital interface, they can simply scroll up and down to switch pages for more information. The Phixel application, however, has only one view and supports 5 rows \times 8 columns exclusively. Certainly, one can add more rows and columns to render a larger data visualisation, but this solution requires more space and it is impossible to increase the size with no limit. A better alternative is to add two sliders to act as the vertical and horizontal scroll bar, so the user can control multiple views by operating both scroll bars.

Apart from these two tasks, the filtering functionality can be further developed to support more advanced data hiding. For example, if a user wants to see all foods whose calories are lower than a certain value, they can push or pull a pixel to make it reach the value and return the result to the Android device. In the meanwhile, an LCD screen is attached on the pixel

for a real-time display of its value. In addition to being used for the filtering task, the LCD screen can also replace the LED strip to show the average of a set of data.

7.3 Including Other Modalities

Currently, Phixel enables seeing, touching and grasping data and the user evaluation has provided convincing evidence that the combination of TUIs and shape-changing displays supports an interesting user experience. It would be more powerful if Pixel can represent a dataset through other modalities such as smelling, hearing or feeling. The first two can be realised by adding a smell diffuser or a speaker, while the last needs to change the physical variables [22] of an object (e.g. hardness or smoothness).

Including such modalities can serve more use cases. For instance, one would like to learn the nutrients of fruit, so they select this food group, and then the smell diffuser will emit a scent of fruit. Being surrounded by this smell, they will always know what they are exploring without having to see a screen. While the user consciously focuses on the physical interface in the foreground, they still catch some information from the background environment [19]. A speaker can also play the same role as the smell diffuser: when the user selects the food group of fruit, playing a piece of music related to fruit is also an ambient media to provide additional information for the user.

Regarding to feeling the data, it requires relatively advanced technologies. As mentioned before, Signer and Curtin [43] proposed the use of robotic arms and some end effectors to change the shape, temperature and texture of a sphere. A similar solution could be applied to Phixel by developing a number of plug-and-play hardware. In this way, a user might get relevant information of a food by touching a sphere and perceiving a change of its temperature, volume or surface.

Although it is unclear whether the simultaneous use of several modalities can benefit exploring datasets, it provides a potential design for dynamic data physicalisation applications. Moreover, it is meaningful for visually impaired people, and an investigation on how they perform data analysis tasks using different modalities can be conducted in the future to see the efficiency of this proposal.

8

Discussion and Conclusion

The results of the user evaluation provide promising evidence that Phixel is a powerful aid in enriching users' interaction with and the reasoning about datasets. However, the Phixel application is still a work in progress and there are a number of aspects that can be further improved. This chapter highlights the benefits and technological challenges of the Phixel application based on the author's analysis and users' feedback, and concludes the work presented in this thesis.

8.1 Discussion

Phixel is an interactive physical interface which provides an enjoyable and intuitive way for users to communicate and understand datasets. As Phixel is not restricted by a specific dataset, it can be used in various contexts. The chapter of use cases has talked about the great potential of Phixel to be applied in a range of scenarios. Additionally, the employment of a portable Android tablet and the simple installation of the physical display enables a flexible and free movement of users. People do not have to change their position to get a better view of the physical bar chart.

Despite the highlights and benefits of the Phixel system, the presented work has several drawbacks. As mentioned at the beginning of the thesis, the work was completed during the COVID-19 period. Therefore, the devel-

opment of Phixel encountered a number of limitations due to the unexpected condition.

The first one is the 3D printing problem. As the WISE lab was not open to master students, the author could not print enough hollow bars which act as the shell of the pixels. Instead, some transparent plastic boards were cut and glued together to replace a 3D printed one thanks to its simple shape. Figure 8.1 shows the comparison between these two objects. Although the handmade shell looks rather casual and crude, its transparency is good enough to transmit light clearly. Another limitation is that the investigation of how Phixel is used in context of multi-user setup and how Phixel affects users' performance in collaborative tasks could not be conducted due to the regulation of social distancing.

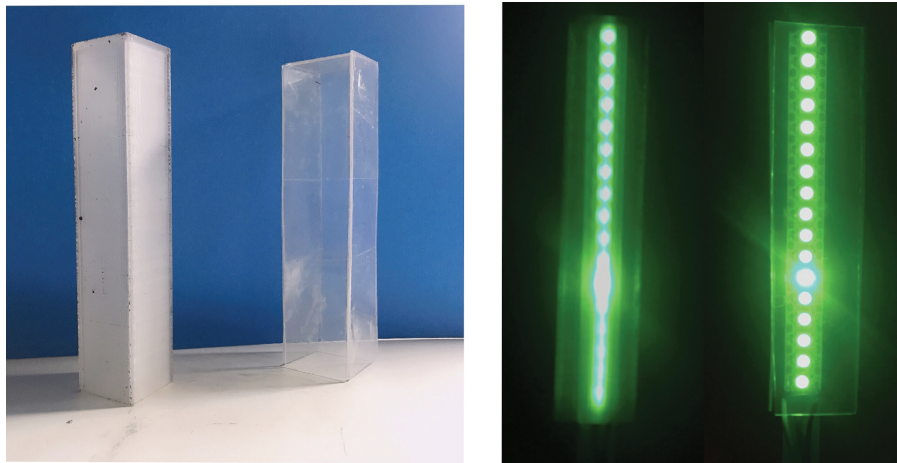


Figure 8.1: Comparison of a 3D printed hollow bar (left) and a handmade one (right)

Apart from the limitations caused by COVID-19, there are several technical challenges of Phixel that need to be addressed in the future. First, the view that Phixel can present is a little bit small. Although the 5×8 array of pixels is enough to offer essential information of a dataset, people may want to explore a large dataset with more dimensions, which exceeds the size of Phixel. The solution to this problem has been discussed in the chapter about future work, that suggests a further development on Phixel to enable the navigation of different views. Alternatively, a participant proposed that *“Instead of always using the 5×8 array of pixels, I would like to have some independent pixels prepared in advance, so that I can customise my own visualisation as I want.”* Second, the motion speed of pixels is fixed to a relatively low value. This is due to the use of the stepper motors which

are used to actuate the pixels are controlled through MCP23017, so their rotation speed are limited by the clock frequency of this I²C chip. It might be possible to find other devices to replace the MCP23017 to provide a higher operation speed. Third, some participants noticed that there was a short pause when the pixels go up or down. The reason for this phenomenon is that Phixel needs some time to read the states of all RFID readers and then drive pixels accordingly. Because there are 14 readers in the Phixel system, this process takes a while to get their status one by one. Finally, the method of using the position of an LED to represent the average value of a set of data does not have a high precision. As described earlier, each LED strip has only 17 LEDs. Therefore, each LED can only indicate an approximate range instead of an accurate value. To address this problem, an LCD screen could be attached to display the average.

8.2 Conclusion

The key objective of this thesis was to develop a dynamic data physicalisation application to promote people's communication with and reasoning about datasets. It is achieved by Phixel, a combination of TUIs and a shape-changing display that renders a physical bar chart in the real world. The proposed Phixel system allows users to physicalise different datasets according to their requirements, and the generated result can be flexibly updated or reorganised.

Phixel mainly consists of a digital software application and a physical display. The former is done on an Android device for providing screens for users to do interactions such as dataset selection, data encoding and hardware state checking, while the latter is composed of a 5×8 array of physical bars (pixels) to structure a bar chart. A pixel is controlled by a stepper motor, and illuminated by an LED strip. Some tangible cards are also used to encode dataset information and then serve as input for the physical interface. The reading and writing operations on these cards are realised by RFID technology. In addition to that, a number of customised 3D printed items are applied as the linkage components between the actuation hardware and the pixels.

In order to investigate the usefulness and usability of Phixel, 10 people were invited to participate in a user study where they explored a dataset about food nutrient and performed several data analysis tasks by using Phixel. In the user study, all needed tangible cards had been prepared in advance so that participants only interacted with the physical display. They were asked to present different physical bar charts, in which the answers to

some pre-defined questions can be founded. The accuracy of participants' answers determined whether Phixel is beneficial to the understanding of datasets. A questionnaire was also involved to quantitatively measure users' experiences with Phixel. Although the user study was not a comparative experiment to compare the performance between a physical visualisation and an on-screen visualisation, the results have shown that: (1) Phixel is useful for helping users quickly find correct answers to data analysis problems, (2) Phixel behaves well for a range of aspects such as ease of learning and ease of use, (3) Phixel is a creative product that catches the interest of users.

In general, the proposed Phixel application provides great help in facilitating users' to understand and explore datasets. Therefore, it can be used for a number of purposes that related to analysing datasets. The thesis presents two possible future usage scenarios. The first use case is to help a food nutrition expert prepare speech materials and share some information about a food dataset with students. The second one is to serve a group in an automobile company for marketing analysis, which involves several relevant datasets. These two scenarios show the great potential of Phixel that can be applied for both individual research and cooperative discussion.

The progress of Phixel is still in an early phase. There are many aspects that can be improved in the future. For instance, the application can be associated with private information for the purpose of customised personal use. It is also possible to involve other modalities in the application, such that users can not only touch and see the data, but also smell or taste the data. Last but foremost, Phixel can be further improved to provide some of the powerful functionalities found in a screen-based visualisation. As such, users can perform all operations of data visualisations in a natural way.

In summary, the thesis presents a combination of TUIs and shape-changing interfaces to develop a dynamic and physical data representation for dataset exploration. Phixel outperforms previous applications with respect to flexibility and portability. It is further expected that the proposed approach can inspire the future development of dynamic data physicalisation.

A

Appendix

A.1 Circuit Diagram Appendix

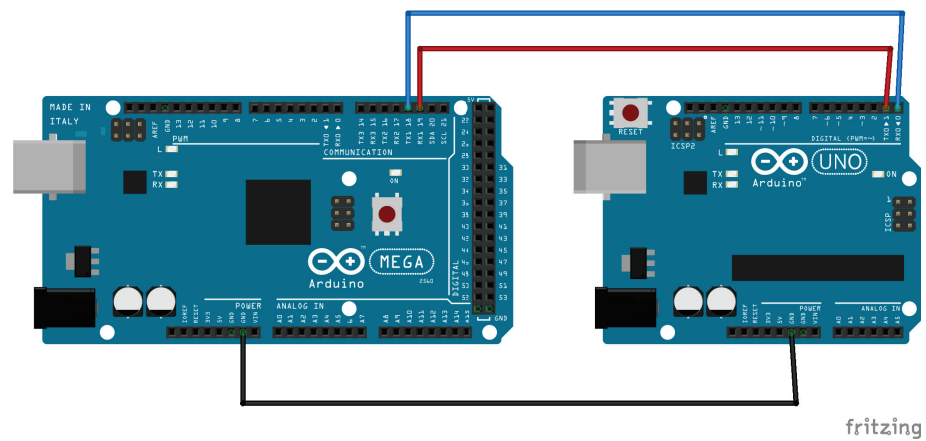


Figure A.1: The serial communication between two boards

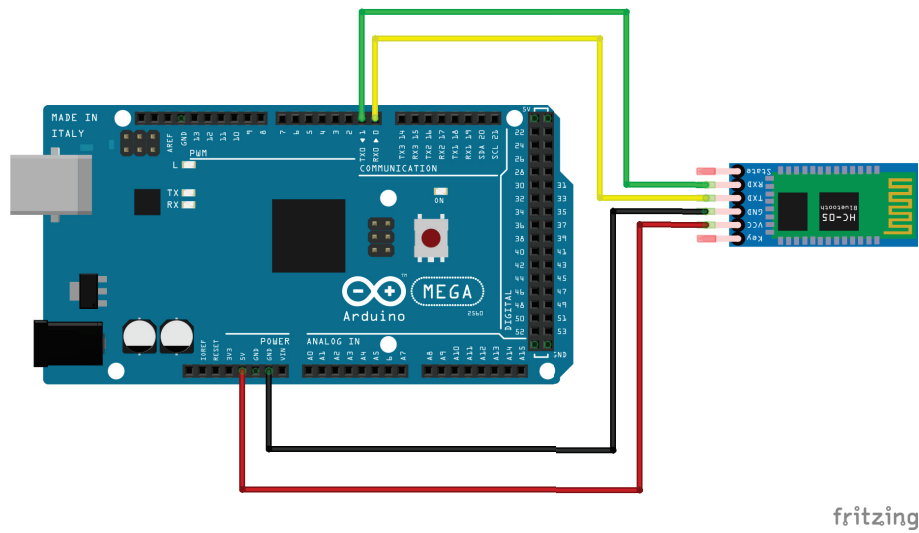


Figure A.2: The HC-05 circuit diagram

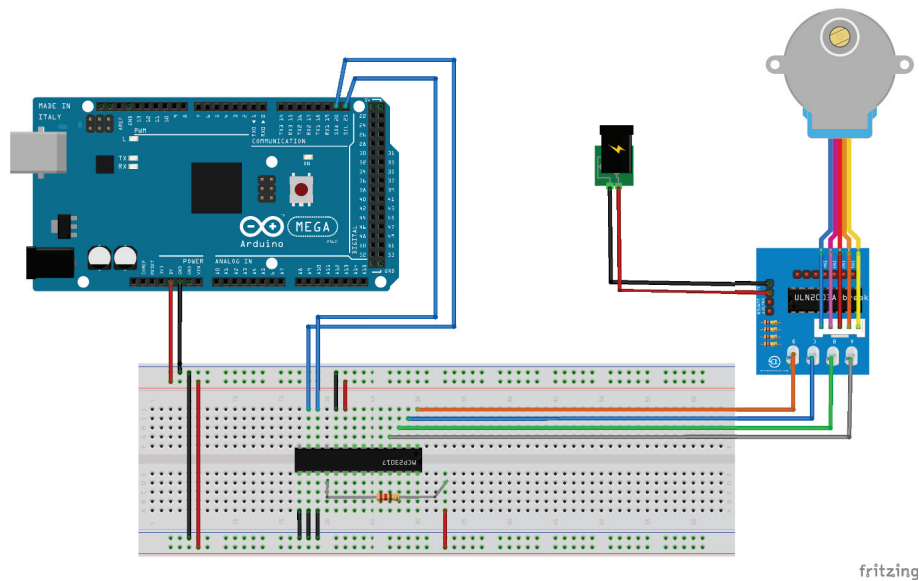


Figure A.3: The circuit diagram to show how a MCP23017 I2C device (address: 0x00) and a stepper motor are wired. If more MCP23017 are needed, just change three address pins and keep other pins unchanged.

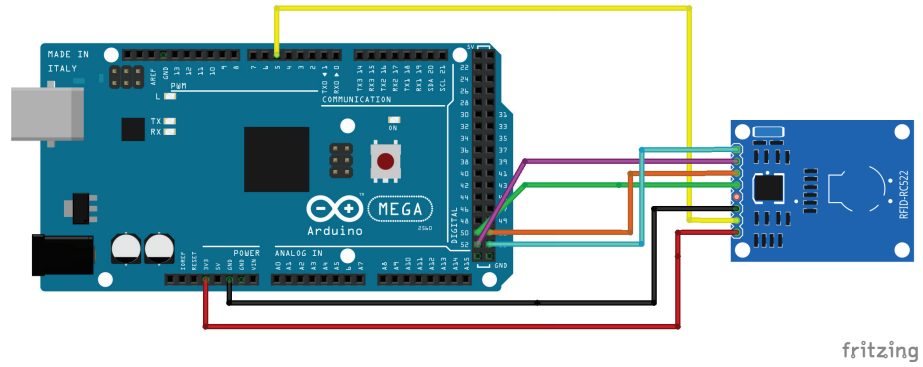


Figure A.4: The MFRC522 RFID reader circuit. If more MRFC522 are required, just change the SS pin and keep other pins unchanged.

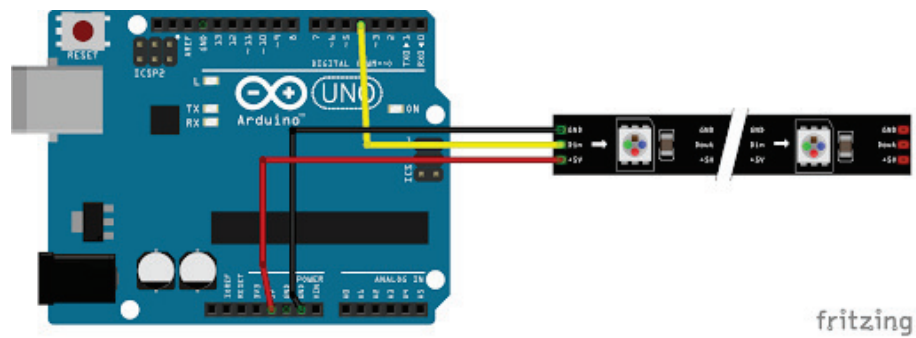


Figure A.5: The WS2812B addressable LED strip circuit

A.2 Evaluation Records Appendix

Pre-test questionnaire

1. Have you ever used bar chart or histogram before?
 - Yes
 - No

2. How often do you use bar chart or histogram?
 - Daily
 - Once a week
 - Once a month
 - Once a year
 - Never

3. How much do you know about the data visualisation technique of bar chart?
 - A lot. I am an expert.
 - A little. I only know basic operations.
 - Only the name and nothing else.

Please assess the product now by ticking one circle per line.

	1	2	3	4	5	6	7		
annoying	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	enjoyable	1
not understandable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	understandable	2
creative	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	dull	3
easy to learn	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	difficult to learn	4
valuable	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	inferior	5
boring	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	exciting	6
not interesting	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	interesting	7
unpredictable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	predictable	8
fast	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	slow	9
inventive	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	conventional	10
obstructive	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	supportive	11
good	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	bad	12
complicated	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	easy	13
unlikable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	pleasing	14
usual	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	leading edge	15
unpleasant	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	pleasant	16
secure	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	not secure	17
motivating	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	demotivating	18
meets expectations	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	does not meet expectations	19
inefficient	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	efficient	20
clear	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	confusing	21
impractical	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	practical	22
organized	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	cluttered	23
attractive	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	unattractive	24
friendly	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	unfriendly	25
conservative	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	innovative	26

Please assess the product now by ticking one circle per line.

	1	2	3	4	5	6	7		
annoying	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	enjoyable	1
not understandable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	understandable	2
creative	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	dull	3
easy to learn	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	difficult to learn	4
valuable	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	inferior	5
boring	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	exciting	6
not interesting	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	interesting	7
unpredictable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	predictable	8
fast	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	slow	9
inventive	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	conventional	10
obstructive	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	supportive	11
good	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	bad	12
complicated	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	easy	13
unlikable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	pleasing	14
usual	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	leading edge	15
unpleasant	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	pleasant	16
secure	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	not secure	17
motivating	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	demotivating	18
meets expectations	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	does not meet expectations	19
inefficient	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	efficient	20
clear	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	confusing	21
impractical	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	practical	22
organized	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	cluttered	23
attractive	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	unattractive	24
friendly	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	unfriendly	25
conservative	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	innovative	26

Please assess the product now by ticking one circle per line.

	1	2	3	4	5	6	7		
annoying	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	enjoyable	1
not understandable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	understandable	2
creative	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	dull	3
easy to learn	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	difficult to learn	4
valuable	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	inferior	5
boring	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	exciting	6
not interesting	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	interesting	7
unpredictable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	predictable	8
fast	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	slow	9
inventive	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	conventional	10
obstructive	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	supportive	11
good	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	bad	12
complicated	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	easy	13
unlikable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	pleasing	14
usual	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	leading edge	15
unpleasant	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	pleasant	16
secure	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	not secure	17
motivating	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	demotivating	18
meets expectations	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	does not meet expectations	19
inefficient	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	efficient	20
clear	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	confusing	21
impractical	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	practical	22
organized	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	cluttered	23
attractive	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	unattractive	24
friendly	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	unfriendly	25
conservative	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	innovative	26

Please assess the product now by ticking one circle per line.

	1	2	3	4	5	6	7		
annoying	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	enjoyable	1
not understandable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	understandable	2
creative	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	dull	3
easy to learn	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	difficult to learn	4
valuable	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	inferior	5
boring	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	exciting	6
not interesting	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	interesting	7
unpredictable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	predictable	8
fast	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	slow	9
inventive	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	conventional	10
obstructive	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	supportive	11
good	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	bad	12
complicated	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	easy	13
unlikable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	pleasing	14
usual	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	leading edge	15
unpleasant	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	pleasant	16
secure	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	not secure	17
motivating	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	demotivating	18
meets expectations	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	does not meet expectations	19
inefficient	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	efficient	20
clear	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	confusing	21
impractical	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	practical	22
organized	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	cluttered	23
attractive	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	unattractive	24
friendly	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	unfriendly	25
conservative	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	innovative	26

Please assess the product now by ticking one circle per line.

	1	2	3	4	5	6	7		
annoying	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	enjoyable	1
not understandable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	understandable	2
creative	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	dull	3
easy to learn	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	difficult to learn	4
valuable	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	inferior	5
boring	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	exciting	6
not interesting	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	interesting	7
unpredictable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	predictable	8
fast	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	slow	9
inventive	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	conventional	10
obstructive	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	supportive	11
good	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	bad	12
complicated	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	easy	13
unlikable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	pleasing	14
usual	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	leading edge	15
unpleasant	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	pleasant	16
secure	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	not secure	17
motivating	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	demotivating	18
meets expectations	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	does not meet expectations	19
inefficient	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	efficient	20
clear	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	confusing	21
impractical	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	practical	22
organized	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	cluttered	23
attractive	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	unattractive	24
friendly	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	unfriendly	25
conservative	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	innovative	26

Please assess the product now by ticking one circle per line.

	1	2	3	4	5	6	7		
annoying	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	enjoyable	1
not understandable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	understandable	2
creative	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	dull	3
easy to learn	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	difficult to learn	4
valuable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	inferior	5
boring	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	exciting	6
not interesting	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	interesting	7
unpredictable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	predictable	8
fast	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	slow	9
inventive	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	conventional	10
obstructive	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	supportive	11
good	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	bad	12
complicated	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	easy	13
unlikable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	pleasing	14
usual	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	leading edge	15
unpleasant	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	pleasant	16
secure	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	not secure	17
motivating	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	demotivating	18
meets expectations	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	does not meet expectations	19
inefficient	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	efficient	20
clear	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	confusing	21
impractical	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	practical	22
organized	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	cluttered	23
attractive	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	unattractive	24
friendly	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	unfriendly	25
conservative	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	innovative	26

Please assess the product now by ticking one circle per line.

	1	2	3	4	5	6	7		
annoying	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	enjoyable	1
not understandable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	understandable	2
creative	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	dull	3
easy to learn	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	difficult to learn	4
valuable	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	inferior	5
boring	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	exciting	6
not interesting	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	interesting	7
unpredictable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	predictable	8
fast	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	slow	9
inventive	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	conventional	10
obstructive	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	supportive	11
good	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	bad	12
complicated	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	easy	13
unlikable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	pleasing	14
usual	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	leading edge	15
unpleasant	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	pleasant	16
secure	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	not secure	17
motivating	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	demotivating	18
meets expectations	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	does not meet expectations	19
inefficient	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	efficient	20
clear	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	confusing	21
impractical	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	practical	22
organized	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	cluttered	23
attractive	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	unattractive	24
friendly	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	unfriendly	25
conservative	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	innovative	26

Please assess the product now by ticking one circle per line.

	1	2	3	4	5	6	7		
annoying	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	enjoyable	1
not understandable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	understandable	2
creative	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	dull	3
easy to learn	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	difficult to learn	4
valuable	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	inferior	5
boring	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	exciting	6
not interesting	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	interesting	7
unpredictable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	predictable	8
fast	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	slow	9
inventive	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	conventional	10
obstructive	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	supportive	11
good	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	bad	12
complicated	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	easy	13
unlikable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	pleasing	14
usual	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	leading edge	15
unpleasant	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	pleasant	16
secure	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	not secure	17
motivating	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	demotivating	18
meets expectations	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	does not meet expectations	19
inefficient	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	efficient	20
clear	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	confusing	21
impractical	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	practical	22
organized	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	cluttered	23
attractive	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	unattractive	24
friendly	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	unfriendly	25
conservative	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	innovative	26

Please assess the product now by ticking one circle per line.

	1	2	3	4	5	6	7		
annoying	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	enjoyable	1
not understandable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	understandable	2
creative	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	dull	3
easy to learn	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	difficult to learn	4
valuable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	inferior	5
boring	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	exciting	6
not interesting	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	interesting	7
unpredictable	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	predictable	8
fast	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	slow	9
inventive	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	conventional	10
obstructive	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	supportive	11
good	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	bad	12
complicated	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	easy	13
unlikable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	pleasing	14
usual	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	leading edge	15
unpleasant	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	pleasant	16
secure	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	not secure	17
motivating	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	demotivating	18
meets expectations	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	does not meet expectations	19
inefficient	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	efficient	20
clear	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	confusing	21
impractical	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	practical	22
organized	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	cluttered	23
attractive	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	unattractive	24
friendly	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	unfriendly	25
conservative	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	innovative	26

Please assess the product now by ticking one circle per line.

	1	2	3	4	5	6	7		
annoying	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	enjoyable	1
not understandable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	understandable	2
creative	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	dull	3
easy to learn	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	difficult to learn	4
valuable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	inferior	5
boring	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	exciting	6
not interesting	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	interesting	7
unpredictable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	predictable	8
fast	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	slow	9
inventive	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	conventional	10
obstructive	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	supportive	11
good	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	bad	12
complicated	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	easy	13
unlikable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	pleasing	14
usual	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	leading edge	15
unpleasant	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	pleasant	16
secure	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	not secure	17
motivating	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	demotivating	18
meets expectations	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	does not meet expectations	19
inefficient	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	efficient	20
clear	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	confusing	21
impractical	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	practical	22
organized	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	cluttered	23
attractive	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	unattractive	24
friendly	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	unfriendly	25
conservative	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	innovative	26

Bibliography

- [1] Shiroq Al-Megren and Roy A Ruddle. Comparing Tangible and Multi-touch Interaction for Interactive Data Visualization Tasks. In *Proceedings of the 10th International Conference on Tangible, Embedded, and Embodied Interaction*, pages 279–286, 2016.
- [2] Christopher Andrews, Alex Endert, and Chris North. Space to Think: Large High-resolution Displays for Sensemaking. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 55–64, 2010.
- [3] Christoph Bader, Dominik Kolb, James C Weaver, Sunanda Sharma, Ahmed Hosny, João Costa, and Neri Oxman. Making Data Matter: Voxel Printing for the Digital Fabrication of Data Across Scales and Domains. *Science Advances*, 4(5):8652, 2018.
- [4] Mike Bailey. Layered Manufacturing for Scientific Visualization. *Communications of the ACM*, 48(6):42–48, 2005.
- [5] William Bevan and Joseph A Steger. Free Recall and Abstractness of Stimuli. *Science*, 172(3983):597–599, 1971.
- [6] Fanny Chevalier, Romain Vuillemot, and Guia Gali. Using Concrete Scales: A Practical Framework for Effective Visual Depiction of Complex Measures. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2426–2435, 2013.
- [7] Alice Cliff and Jenny Rinkinen. Visualising Electricity Demand. *Science Museum Group Journal*, 9(09), 2018.
- [8] Maxime Cordeil, Andrew Cunningham, Tim Dwyer, Bruce H Thomas, and Kim Marriott. ImAxes: Immersive Axes as Embodied Affordances for Interactive Multivariate Data Visualisation. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, pages 71–83, 2017.

- [9] Son Do-Lenh, Patrick Jermann, Sébastien Cuendet, Guillaume Zufferey, and Pierre Dillenbourg. Task Performance vs. Learning Outcomes: A Study of a Tangible User Interface in the Classroom. In *European Conference on Technology Enhanced Learning*, pages 78–92. Springer, 2010.
- [10] Sean Follmer, Daniel Leithinger, Alex Olwal, Nadia Cheng, and Hiroshi Ishii. Jamming User Interfaces: Programmable Particle Stiffness and Sensing for Malleable and Shape-changing Devices. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, pages 519–528, 2012.
- [11] Sean Follmer, Daniel Leithinger, Alex Olwal, Akimitsu Hogge, and Hiroshi Ishii. inFORM: Dynamic Physical Affordances and Constraints Through Shape and Object Actuation. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, pages 417–426, 2013.
- [12] Alexandre Gillet, Michel Sanner, Daniel Stoffler, David Goodsell, and Arthur Olson. Augmented Reality with Tangible Auto-fabricated Models for Molecular Biology Applications. In *IEEE Visualization 2004*, pages 235–241. IEEE, 2004.
- [13] Alexandre Gillet, Michel Sanner, Daniel Stoffler, and Arthur Olson. Tangible Augmented Interfaces for Structural Molecular Biology. *IEEE Computer Graphics and Applications*, 25(2):13–17, 2005.
- [14] Alfred Cort Haddon. *Reports of the Cambridge Anthropological Expedition to Torres Straits*, volume 4. The University Press, 1912.
- [15] Raven Hanna. Weaving Weather. *Proceedings of the National Academy of Sciences of the United States of America*, 110(9):3205, 2013.
- [16] Jeffrey Heer and Ben Shneiderman. Interactive Dynamics for Visual Analysis. *Queue*, 10(2):30–55, 2012.
- [17] Hiroshi Ishii, Dávid Lakatos, Leonardo Bonanni, and Jean-Baptiste Labrune. Radical Atoms: Beyond Tangible Bits, Toward Transformable Materials. *Interactions*, 19(1):38–51, 2012.
- [18] Hiroshi Ishii, Daniel Leithinger, Sean Follmer, Amit Zoran, Philipp Schoessler, and Jared Counts. TRANSFORM: Embodiment of "Radical Atoms" at Milano Design Week. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, pages 687–694, 2015.

-
- [19] Hiroshi Ishii and Brygg Ullmer. Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, pages 234–241, 1997.
- [20] Hiroo Iwata, Hiroaki Yano, Fumitaka Nakaizumi, and Ryo Kawamura. Project FEELEX: Adding Haptic Surface to Graphics. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, pages 469–476, 2001.
- [21] Yvonne Jansen, Pierre Dragicevic, and Jean-Daniel Fekete. Evaluating the Efficiency of Physical Visualizations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2593–2602, 2013.
- [22] Yvonne Jansen, Pierre Dragicevic, Petra Isenberg, Jason Alexander, Abhijit Karnik, Johan Kildal, Sriram Subramanian, and Kasper Hornbæk. Opportunities and Challenges for Data Physicalization. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3227–3236, 2015.
- [23] Haotian Jiang, James Starkman, Menghan Liu, and Ming-Chun Huang. Food Nutrition Visualization on Google Glass: Design Tradeoff and Field Evaluation. *IEEE Consumer Electronics Magazine*, 7(3):21–31, 2018.
- [24] Yoshiyuki Kawano and Keiji Yanai. Real-time Mobile Food Recognition System. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–7, 2013.
- [25] Daniel Leithinger, Sean Follmer, Alex Olwal, Samuel Luescher, Akimitsu Hogge, Jinha Lee, and Hiroshi Ishii. Sublimate: State-changing Virtual and Physical Rendering to Augment Interaction with Shape Displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1441–1450, 2013.
- [26] Daniel Leithinger and Hiroshi Ishii. Relief: A Scalable Actuated Shape Display. In *Proceedings of the 4th International Conference on Tangible, Embedded, and Embodied Interaction*, pages 221–222, 2010.
- [27] Daniel Leithinger, David Lakatos, Anthony DeVincenzi, Matthew Blackshaw, and Hiroshi Ishii. Direct and Gestural Interaction with Relief: A 2.5D Shape Display. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, pages 541–548, 2011.

- [28] Henry Robinson Luce. *Fortune*, volume 23, page 94. Time, June 1941.
- [29] Stefan Marks, Javier E Estevez, and Andy M Connor. Towards the Holodeck: Fully Immersive Virtual Reality Visualisation of Scientific and Engineering Data. In *Proceedings of the 29th International Conference on Image and Vision Computing*, pages 42–47, 2014.
- [30] Tom Moher. Embedded Phenomena: Supporting Science Learning with Classroom-sized Distributed Simulations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 691–700, 2006.
- [31] Mathieu Nancel, Julie Wagner, Emmanuel Pietriga, Olivier Chapuis, and Wendy Mackay. Mid-air Pan-and-zoom on Wall-sized Displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 177–186, 2011.
- [32] Ryuma Niiyama, Lining Yao, and Hiroshi Ishii. Weight and volume Changing Device with Liquid Metal Transfer. In *Proceedings of the 8th International Conference on Tangible, Embedded and Embodied Interaction*, pages 49–52, 2014.
- [33] Don Norman. *The Design of Everyday Things: Revised and Expanded Edition*. Basic Books, 2013.
- [34] Laura Jones Perovich. *Data Experiences: Novel Interfaces for Data Engagement Using Environmental Health Data*. PhD thesis, Massachusetts Institute of Technology, 2014.
- [35] Ivan Poupyrev, Tatsushi Nashida, Shigeaki Maruyama, Jun Rekimoto, and Yasufumi Yamaji. Lumen: Interactive Visual and Shape Display for Calm Computing. In *ACM SIGGRAPH 2004 Emerging Technologies*, page 17. 2004.
- [36] Gary Priestnall, Jeremy Gardiner, Jake Durrant, and James Goulding. Projection Augmented Relief Models (PARM): Tangible Displays for Geographic Information. *Electronic Visualisation and the Arts (EVA 2012)*, pages 180–187, 2012.
- [37] Manika Puri, Zhiwei Zhu, Qian Yu, Ajay Divakaran, and Harpreet Sawhney. Recognition and Volume Estimation of Food Intake Using a Mobile Device. In *2009 Workshop on Applications of Computer Vision (WACV)*, pages 1–8. IEEE, 2009.

-
- [38] Wolf-Dieter Rase. Creating Physical 3D Maps Using Rapid Prototyping Techniques. In *True-3D in Cartography*, pages 119–134. Springer, 2011.
- [39] Majken K Rasmussen, Esben W Pedersen, Marianne G Petersen, and Kasper Hornbæk. Shape-changing Interfaces: A Review of the Design Space and Open Research Questions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 735–744, 2012.
- [40] Vincent F Scalfani and Thomas P Vaid. 3D Printed Molecules and Extended Solid Models for Teaching Symmetry and Point Groups. *Journal of Chemical Education*, 91(8):1174–1180, 2014.
- [41] D Schmandt-Besserat. *Before Writing: Volume 1: From Counting to Cuneiform*. University of Texas Press, December 1992.
- [42] Bertrand Schneider, Patrick Jermann, Guillaume Zufferey, and Pierre Dillenbourg. Benefits of a Tangible Interface for Collaborative Learning and Interaction. *IEEE Transactions on Learning Technologies*, 4(3):222–232, 2010.
- [43] Beat Signer and Timothy J Curtin. Tangible Holograms: Towards Mobile Physical Augmentation of Virtual Objects. *arXiv preprint arXiv:1703.08288*, 2017.
- [44] Simon Stusak, Jeannette Schwarz, and Andreas Butz. Evaluating the Memorability of Physical Visualizations. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3247–3250, 2015.
- [45] Faisal Taher, John Hardy, Abhijit Karnik, Christian Weichel, Yvonne Jansen, Kasper Hornbæk, and Jason Alexander. Exploring Interactions with Physically Dynamic Bar Charts. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3237–3246, 2015.
- [46] Faisal Taher, Yvonne Jansen, Jonathan Woodruff, John Hardy, Kasper Hornbæk, and Jason Alexander. Investigating the Use of a Dynamic Physical Bar Chart for Data Exploration and Presentation. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):451–460, 2016.
- [47] John Tiab, Sebastian Boring, Paul Strohmeier, Anders Markussen, Jason Alexander, and Kasper Hornbæk. Tiltstacks: Composing Shape-changing Interfaces Using Tilting and Stacking of Modules. In *Proceed-*

- ings of the 2018 International Conference on Advanced Visual Interfaces*, pages 1–5, 2018.
- [48] Brygg Ullmer and Hiroshi Ishii. Emerging Frameworks for Tangible User Interfaces. *IBM Systems Journal*, 39(3.4):915–931, 2000.
- [49] Daniel Vogel and Ravin Balakrishnan. Distant Freehand Pointing and Clicking on Very Large, High Resolution Displays. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology*, pages 33–42, 2005.
- [50] Georg Waltner, Michael Schwarz, Stefan Ladstätter, Anna Weber, Patrick Luley, Horst Bischof, Meinrad Lindschinger, Irene Schmid, and Lucas Paletta. Mango-Mobile Augmented Reality with Functional Eating Guidance and Food Awareness. In *International Conference on Image Analysis and Processing*, pages 425–432. Springer, 2015.
- [51] Hsin-Kai Wu and Priti Shah. Exploring Visuospatial Thinking in Chemistry Learning. *Science education*, 88(3):465–492, 2004.
- [52] Lining Yao, Ryuma Niiyama, Jifei Ou, Sean Follmer, Clark Della Silva, and Hiroshi Ishii. PneuUI: Pneumatically Actuated Soft Composite Materials for Shape Changing Interfaces. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, pages 13–22, 2013.
- [53] Beth Yost, Yonca Haciahmetoglu, and Chris North. Beyond Visual Acuity: the Perceptual Scalability of Information Visualizations for Large Displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 101–110, 2007.
- [54] Jeff Zacks, Ellen Levy, Barbara Tversky, and Diane J Schiano. Reading Bar Graphs: Effects of Extraneous Depth Cues and Graphical Context. *Journal of Experimental Psychology: Applied*, 4(2):119, 1998.