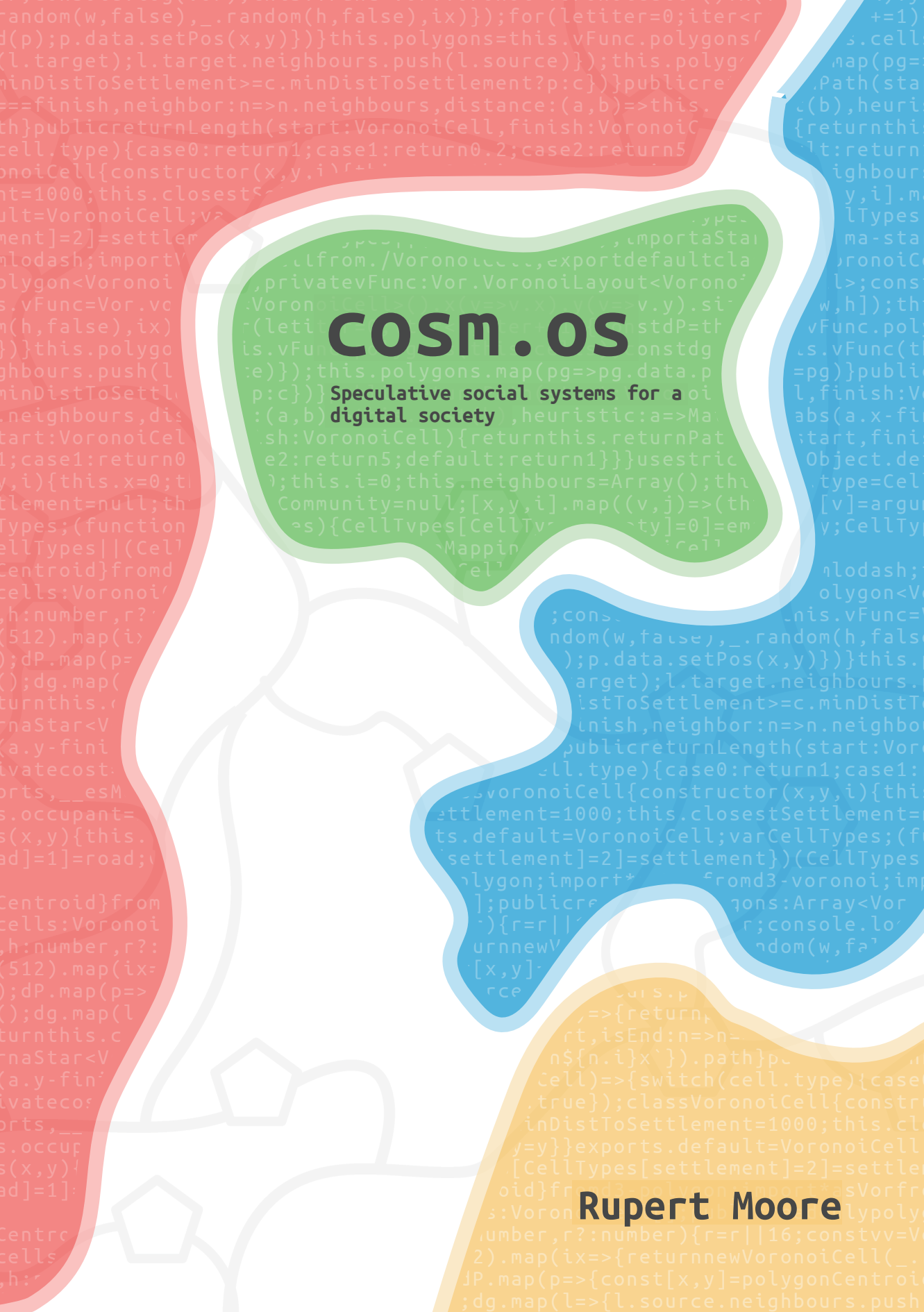

cosm.os
Speculative social systems for a digital society
Rupert Moore

~ Abstract

Our current online social interactions are not satisfying our needs for identity, expression and community. In a context where almost all our social media is experienced through systems and processes that are established by corporations with their own interests in our interactions. This situation has lead to the purposeful or negligent obfuscation of the structural aspects of these systems by those we have empowered as caretakers of our online identities. As the problems with the current system are too entrenched to fix iteratively, this project takes a speculative approach through abstracting, simulating, and presenting these social systems in ways that explain their complexities.

Declaration Of Authenticity

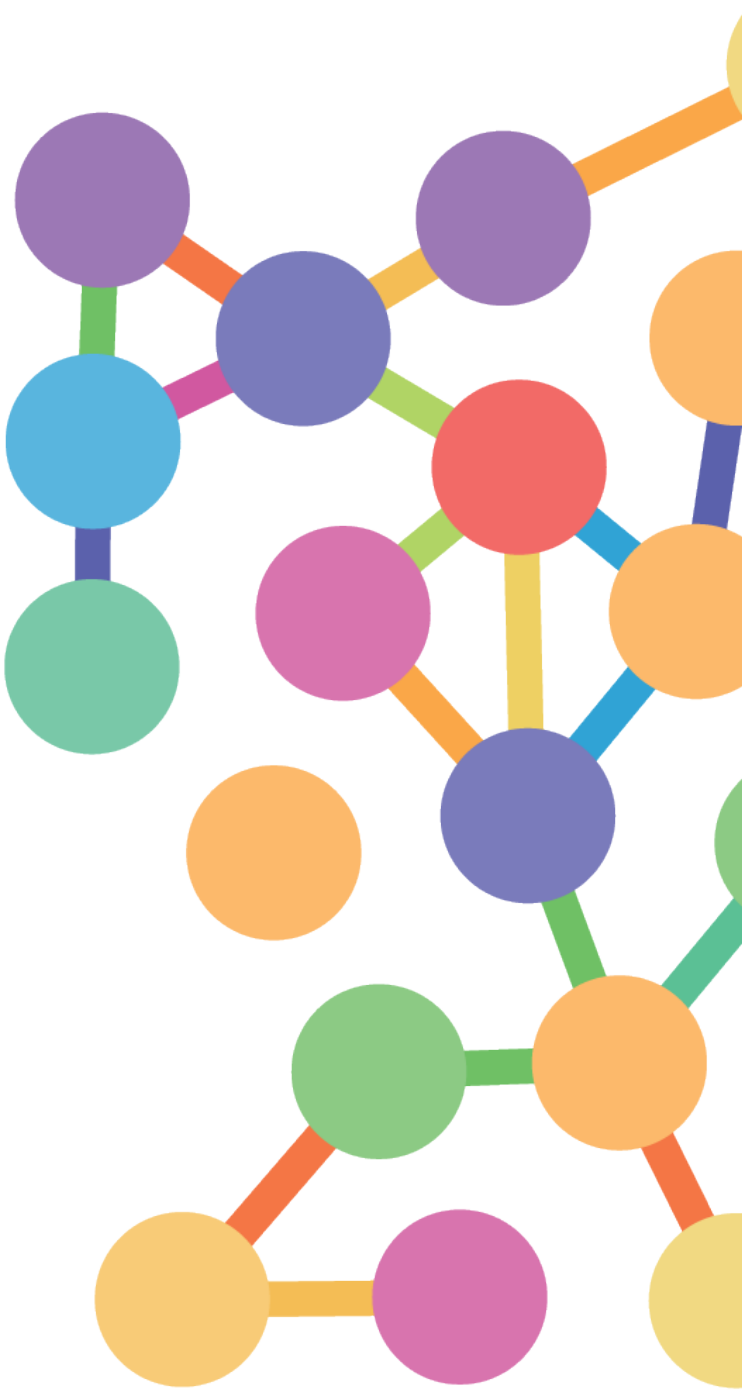
I certify that except where due acknowledgement has been made, the work is that of the author alone: the work has not been submitted previously, in whole or in part, to qualify for any other academic award; the content of the thesis/project is the result of work which has been carried out since the official commencement date of the approved research program; any editorial work, paid or unpaid, carried out by a third party is acknowledged; and, ethics procedures and guidelines have been followed.

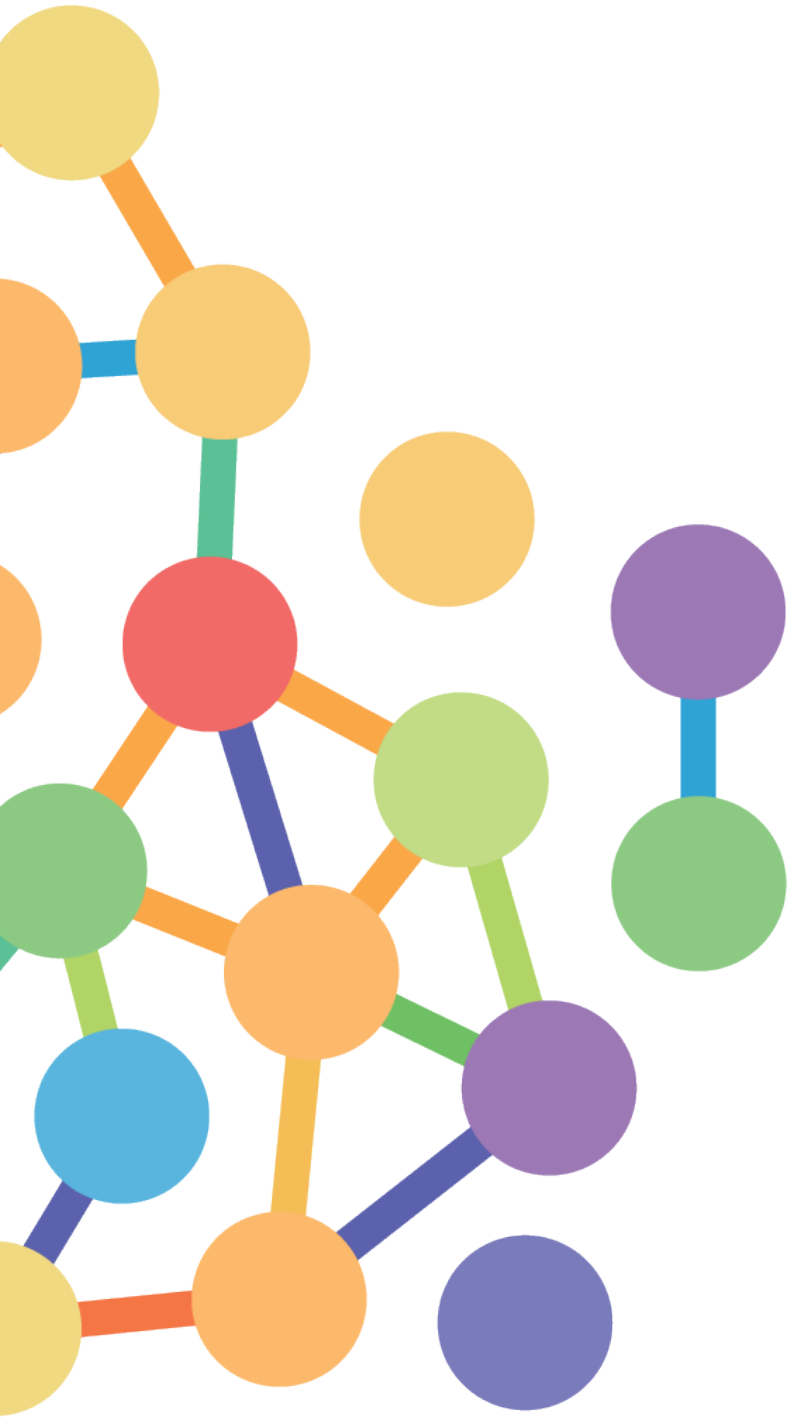
Rupert Quail Moore
November 2018

I would like to thank all my friends and family who have supported me and my project over the past year. In particular, I'd like to thank my father who worked exceptionally hard on many parts of this project, and my supervisor Liam Fennessy for his patience with my weird project. I would also like to thank Juliette Anich and all other industrial design staff who have been incredibly helpful to me. This has been an incredible year for me, and this project and I are in debt to all of you

Table_Of_Contents

~ List_Of_Figures





> Introduction

<head>
<intro>

This project explores the theme of explaining and speculating on complex systems and problems through the use of simulation, interaction, experimentation, and play. Specifically addressing the issues of online social networking, communication, and expression: from the consumption and generation of identity, how the constraints of online interaction affect our behaviour, and how our online footprints are recorded and used by corporations for their own goals.

The thesis then continues to explore the use of interactive digital visualisations to explore and explain complex systems. The final section explores the nature of key design research methods for achieving the project. These include literature review, expert interviews, experimentation, digital prototyping and system/service mapping. These methods allow for a high degree of freedom in the development of the project, as well as a framework to iteratively improve the technical elements, and analyse it's feasibility.

The outcome of this project is an interactive installation, ideally located in a science museum, gallery or library. This installation would enable participants to explore and interact with speculative online social simulations. Through this, participants would gain deeper understandings of the systems that underpin our current online social interactions.

The first chapter reviews a selection of literature in the field of online social interactions. The chapter is divided into five sections:

- social media behaviour,
- digital production of social identity data,
- digital consumption and how it affects our communities and behaviours,
- how social data is used to understand behaviours better and manipulate them,
- speculation on future online social systems.

The second chapter delves into the context the projected is situated in, and the the third explores the applications of this project. The second chapter begins with a discussion of the methodology used to interpret and communicate complex situations and how this applies to online social behaviour. It continues with the case study of American designer Nicky Case's methodology for developing interactive explanations to complex issues. The last section of the context chapter briefly discusses skeuomorphism in digital design and the advantages and drawback it brings.

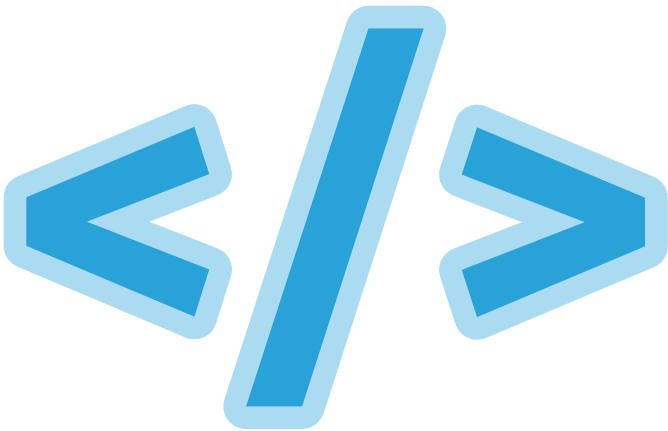
The applications chapter begins with a brief explanation of interactive exhibits before jumping into the case study of San Francisco's Exploratorium, and how the act of "play" is used in this project.

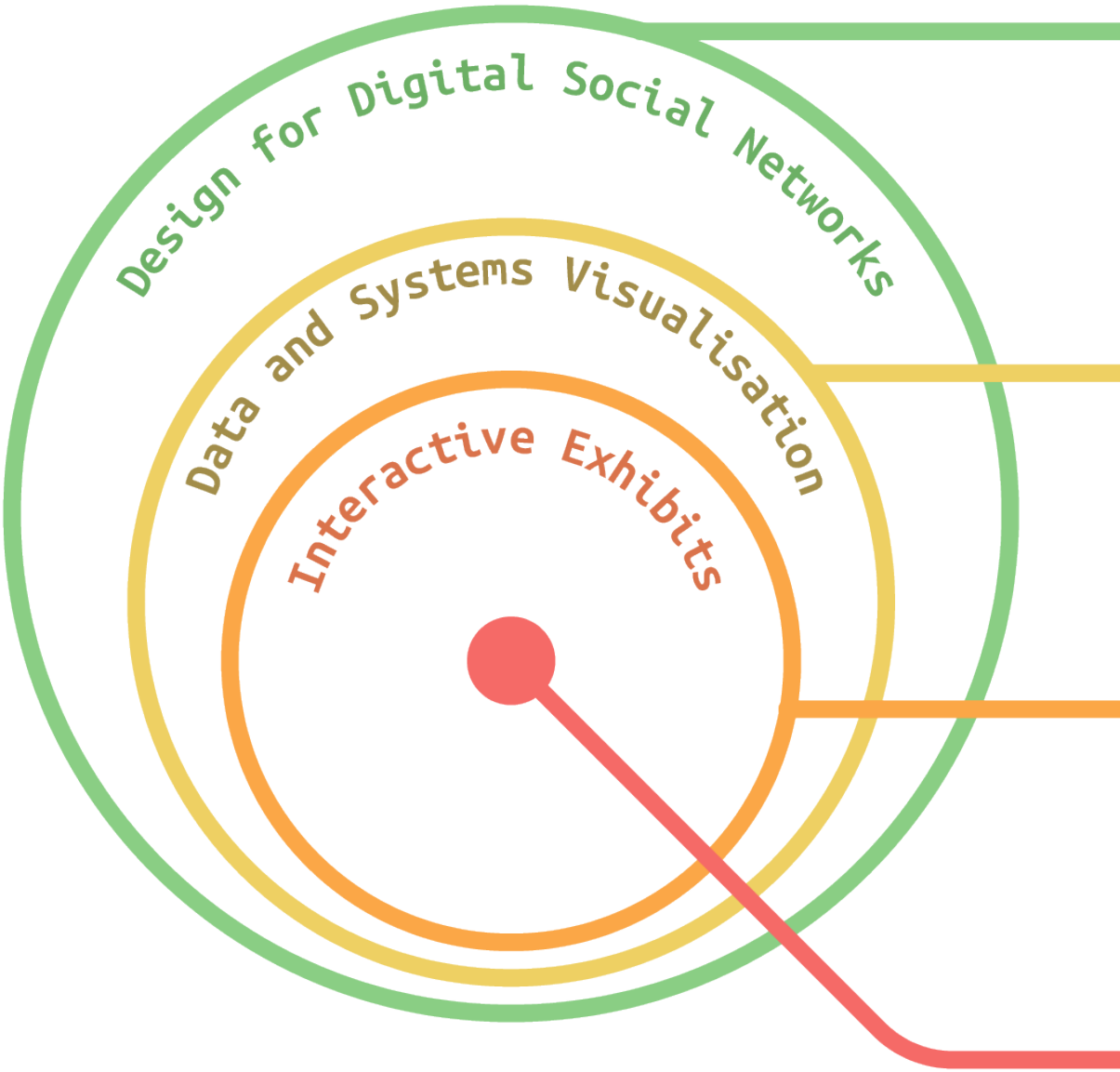
This section is followed by a discussion of smart objects and how they are already a large part of both installations, as well as our online social experiences. Followed by a description of how these objects are incorporated into the project. Finally, this project discusses and defines machine learning in the context of this project and speculates on how it can be applied to the problems faced by these complex social systems.

After prototyping, comes a section on creative coding. This section explains how the application of programming and automation in design improves the project's outcome. The final section discusses journey mapping and service/system blueprinting and how it is used to define the bounds and capabilities of the project and identify key problems and inefficiencies.

The next chapter details the methods used to accomplish the project. It begins with a section on literary research and review. This method is used to gain a greater understanding of the fields this design is positioned within. Following that is a section on digital experimentation. This method is used to explore possible systems, methods and behaviours for simulation or presentation. Next, prototyping, user testing and how these methods are used to refine the technical and interactive elements of the project is explained. After prototyping, comes a section on creative coding. This section explains how the application of programming and automation in design improves the project's outcome. The final section discusses journey mapping and service/system blueprinting and how it is used to define the bounds and capabilities of the project and identify critical problems and inefficiencies.

The design development is divided into four different sections, each detailing a different aspect of the project. Firstly detailing the development of the simulation, then the development of the physical interactive objects, before finally discussing the visual design of the project, and the physical exhibition artefacts of the plinths and table. Finally, the design evaluation and outcome are detailed and concluded.



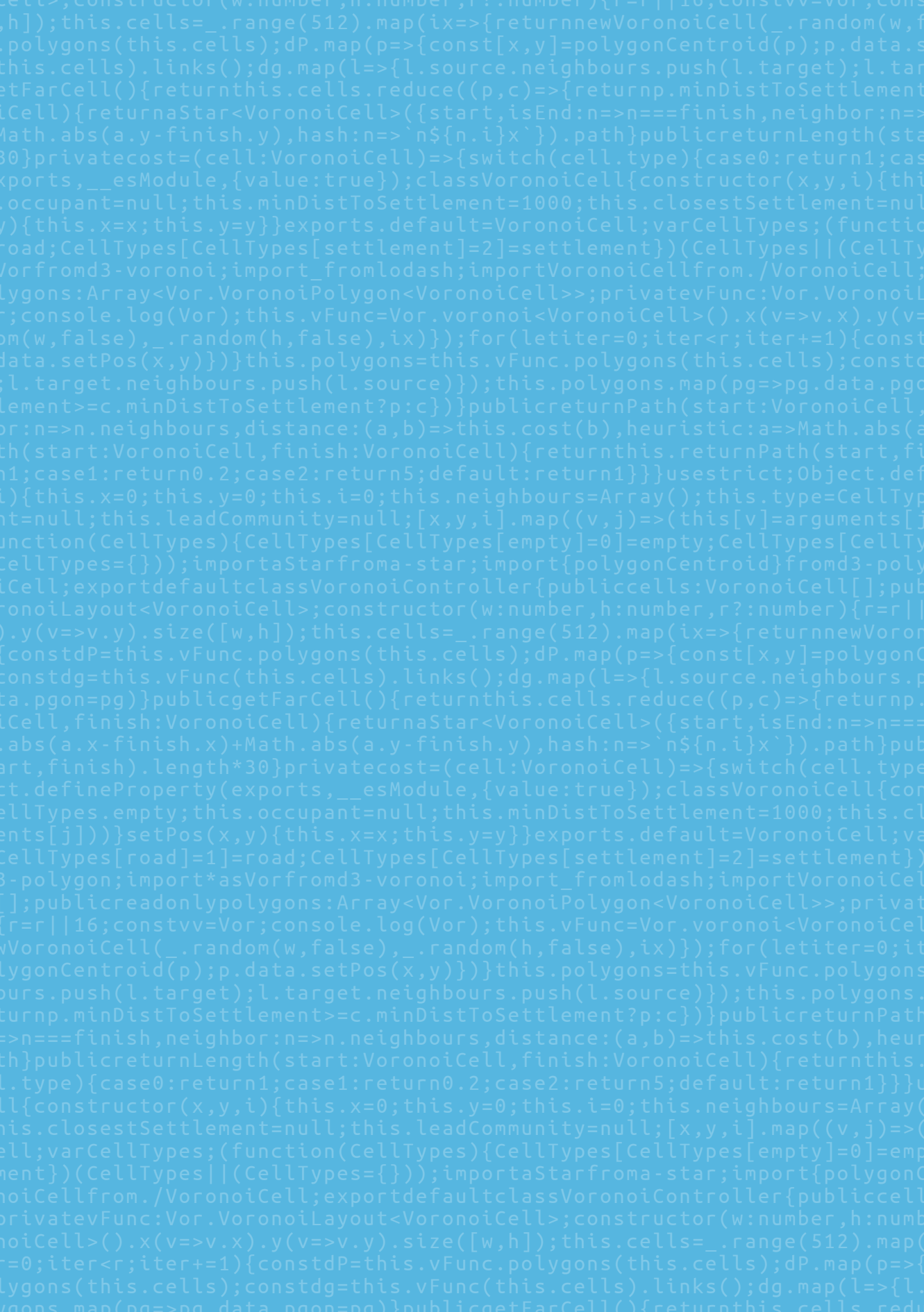

Design for Digital Social Networks
Data and Systems Visualisation
Interactive Exhibits

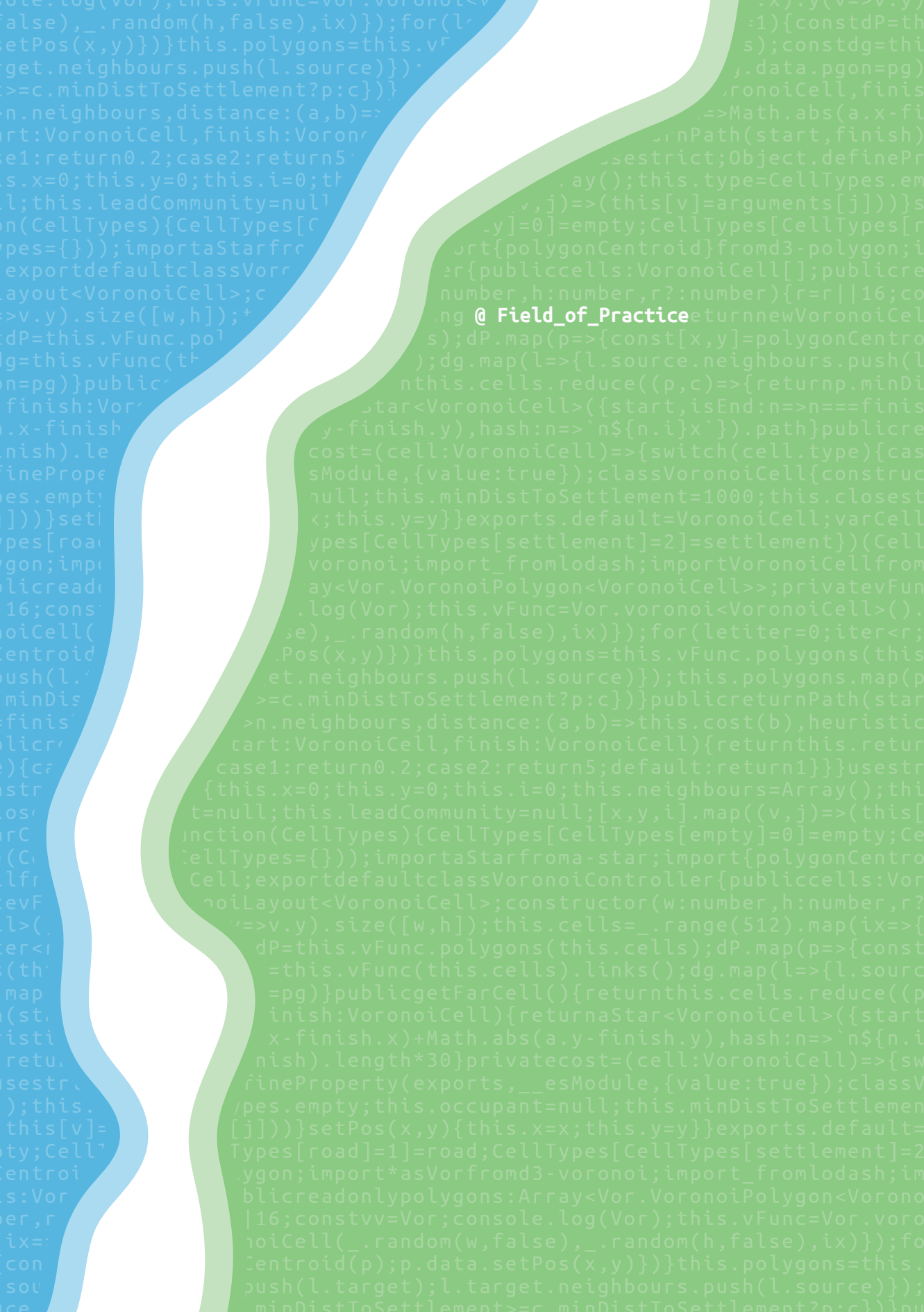
Field of Research and Design

Project Context

Project Application

My Project




@ Field_of_Practice

Introduction

The establishment of an online 'self' is a messy proposition, rife with metaphor and miscommunication. Online we strive to assert ourselves in an incomprehensible network of electric signals transferring many binary digits. We do this through our communication with others; over text in instant messages; audio through VOIP phone calls; internet radio or podcasts; video calls through services such as Skype or FaceTime; and though streaming 'ourselves' to the broader world through social media platforms such as Facebook, Periscope or Twitch.

All internet users continuously leave a footprint consciously or not, in the form of written text, photos, and videos. Health data such as weight loss and daily step counts and consumption choices are similarly left. This mass of data constitutes the trove of metadata that we are constantly producing. The overwhelming volume of data generated by online behaviours often leads to the processes that sort and present this data becoming obfuscated, partly due to companies desire to protect their IP, and partly to disguise their occasional malevolence to these interactions.

With social media in particular this confusion leads to presentations of identity, that while being genuine, are firmly established within the norms enforced by the media service's culture, often discriminating against any difference. In 1999 Bob Stein Said

> "When 1984 came and went, 'Americans congratulated themselves on the fact that Orwell's Big Brother had not materialised in the West. But what people missed, of course, was that Huxley's infinitely darker vision had come true. As Postman put it, In Brave New World, Huxley saw a time coming when 'people will come to love their oppression, to adore technologies that. undo their capacities to think'" (Stein, 1999, p. 204).

Unfortunately, now in our digital internet society, we now contend with both dystopian realities.

Social Media Behaviour

Social Media produces different kinds of inter-personal communication than those prior to the internet. Marcel Danesi, (2009, ch. 4) discusses how pop culture and the mediums of communication have affected our social status as well as well our modes of communication. Through interactions such as social media likes being broadcast, and publicly facing music streaming playlists (such as in the music streaming and social service, Spotify) our connections and knowledge on various pop culture commodities are a key aspect of modern online identity.

We have adopted pop cultural symbolism into our communication, through the technical aspects of text messages, tweet character limits, and the imperfections of autocorrect systems. These technical factors along with the vastness of information available to consume online, have led to linguistic changes and shorthands in our language, and particularly to an increase in the use of sensational and dramatic language in order to attract the attention of others in the vastness of social media interaction. All of these changes have contributed to a post-modern consumerist method of identity curation, centred around media consumption and association.

Hogan (2010) discusses how theories established by Goffman (1969) can be used to analyse and describe social media behaviour, with the essential difference that current social media behaviour has a far more exhibitory basis. in contrast to the more performative face to face communication that Goffman described. Additionally the concept of a “backstage” which Goffman put forward as a private space, is no longer as private is it used to be. Hogan introduces the part of the “Curator”. The curator acts as a medium for artefacts and performances to travel through, but controls both their access and their perception. Other researchers have examined how social media, in combination with the modern world, has led to a decline in “social capital”.

This research however, fails to recognise two key values that online communication can have for many people in the community: those with non-normative (and sometimes frowned upon) identities whose socialisation within own or adjacent communities would be difficult or even dangerous offline; and, the low barrier of entry for online communication for many people with social anxieties who are comforted by many aspects of anonymity that internet communication can provide (Young & Lo, 2012), and can assume other or multiple identities for their online 'self'.

Berberick & McAllister (2016) Discusses how social media encourages a very corporate friendly method of identity establishment through add filled pages, claiming: "Even unsponsored copy often fulfils imperatives of digital advertising by attracting visitors and creating comfortable, accessible, and fun content that flows with advertising-based elements. Identities are thus constructed in a commercial environment." (Berberick & McAllister, 2016, p. 3423). Tieing back to the work of Danesi (2009), This is also expressed in the emergence of "Quizzes" that are strongly linked with cultural commodities such as tv shows, or memes. Berberick & McAllister identify the desire for people to use social media in order to establish identities through these quizzes. The quizzes themselves offer plenty for the service provider: from advertising opportunities to data gathering. Interestingly they establish that these quizzes, often avoid the more grounded aspects of gender, sexuality, race, age, and social class of the representative categories, instead focus on simplified readings of character or branded attributes.

Looking at the drop in participation in social groups or clubs (such as bowling clubs) Putnam (2000, ch. 9) asserts that this is due to growth in individualism, and while seeing the promising nature of digital communication, it has many problems to overcome such as the lack of accountability inherent in it.

Users who are identified as "Ross" by a "Which friends character are you?" quiz, do not need to be a white cisgender middle-class male palaeontologists, which the character's more common traits, to be considered a "Ross" but they do need to have watched "F.R.I.E.N.D.S", in order to grasp its relevance to their own identity constructions.

> Quiz responses are not right or wrong, but purportedly declare something about the quiz taker, whether about the perceived applicability of the results themselves to the taker's interests or personality, or the spirit of fun and community in which quizzes are located (Berberick & McAllister, 2016, p. 3424)


000
00101010
1010010010
10010100000
0010101010
0100101001
0000000101010
1010010010100
1010000000101
0101010010010
1001010000000
1010101010010
0101001010000
0010101010100
0010100101000
0001010101010
00010101
010100
100101
001010
0000001
0101010100
1001010010100
0000101010101010100100
1010010100000101010101
0100100101001010010100
0000001010101010010010
1001010000000101010101

Digital Data Production

In January of 2018, the National Gallery of Victoria, as part of their "Triennial Extras" program hosted a debate on the validity of online presentations of identity, this was recorded by the ABC to be then broadcast on radio national ("The virtual you and the real you," 2018). They had a diverse group of speakers, including video game producers, broadcasters, and science journalists. Of principal interest to this project was the position of the first speaker for the affirmative, Vanessa Toholka, and the experiences of the first speaker for the negative Lisy Kane.

Vanessa's argument asserted that while the internet might not encompass the material self, it contains a record of both their actions and also in many ways their desires, which she asserts could be more valuable than the material. In opposition of this, Lisy, while referring to online multiplayer video games, asserted that as a woman of colour, she could not present that aspect of her identity, as it was not a safe environment for her to do so, therefore the virtual world could not capture the authentic Lisy.

This alienation can be seen as a design problem at the interaction and service level, rather than a problem with the medium itself. Many other points were brought up including the value of experimentation and exploration of identity that anonymity allows, and what value should be ascribed to that. From this discussion, It is evident that even the concept of a digital identity is perceived in a variety of different ways. A key aspect of healthy communities is the strength of both the collective identity, and its ability to acknowledge the identities of it's members (Paech, 2018).

Digital Data Consumption

> "suggestions of greater development of technologies (referring to Social Media), and greater usage of those technologies in varying degrees, must also include the possibilities of new forms of democracies and democratic practices.
> (Bebawi & Bossio, 2014, p. 116)

In discussing the sociopolitical ramifications that social media has on journalism, democratic practice, and revolution, Bebawi & Bossio (2014) touch on how due to the accessible nature of services such as Twitter, it has become a resource to view significant events as they happen, by people who are affected by them. This is, however, is not as helpful when governments suppress access to these social networks to control both protests and organised responses to events, as well as the message, such as the continued protests that happened in Iran, which where solely organised through social media, and lost all momentum when access was revoked (Bebawi & Bossio, 2014).

Even in circumstances such as this though, new methods of networking through ad-hoc social tech have emerged. In Hong Kong during the umbrella protests, mobile phone service was shut down revoking access to social media for the protesters. However, many in the movement had begun using FireChat, which uses Bluetooth and Peer to Peer (P2P) network protocols to create a communication network when there is no internet access (LEE & Chui-Shan Chow, 2016).

The diversity of fandom communities is startling and often very particular. Communities such as 'femslash' (a queer fan fiction category that takes female-coded popular figures and places them in Sapphic romantic narratives), to communities that focus on the nature and representation of disability in comics, to the re-representation of wrestling by wrestling fans, and the practice of broadcasting the mimicking of the pivotal moments in key matches by European football fan communities. Click and Scott assert that due to the low barrier of entry, and the freedom in identity development allowed fan works to reach levels of popularity the would not have been possible without social media (2017, ch. 24).

Click and Scott (2017) offer a fantastic text collecting various academic contributions on the nature of the 'fandom' phenomena. Fandom refers to very heavily involved subcultures and communities that revolve around a shared interest in specific cultural, material, technological or political things or systems. These subcultures often become quite invested, and can seem to be obsessive from people outside of the fandom. Of particular note to the project are the chapters relating to the intersection of fandom and Identity. The internet and social media platforms such as LiveJournal, Tumblr, Twitter, and Youtube, allow these sometimes micro-niche and highly dispersed communities to find a shared space. These works include fan art, fan fiction, cosplay, amateur cover songs, and vidding (which involves using music overlaid on preexisting video content in order change its meaning) or other forms of remix culture. In contrast to straight male masculine social makeups of a lot of pop culture communities, in particular those relating to "geeky" subjects (such as comic-books, and tabletop roleplaying games such as Dungeons and Dragons). Online fandom communities are often queer, or femininely coded, suggesting a form of refuge for those who are often turned away from, of turned off by the larger more visible communities.

Due to the highly visible nature of online toxic communication, and the damage it can cause (such as through cyber bullying) our ability to empathise online becomes a factor in grading the value of online relationships. However "Virtual" Communication, according to Carrier, Spradlin, Bunce, & Rosen (2015), has little negative impact on empathy. The report found that while there was a link between activities such as video gaming and lower measurable levels of empathy, generally "virtual" empathy was shown to be positively correlated with empathy in 'real life'.

Digital Data Analysis and Manipulation

The way social data is harnessed has severe ramifications. Nicholas Christakis (2017) discusses the results of some experiments he has conducted with vast amounts of social data that he has harnessed. He shows how various models of social behaviour can be used to analyse online social interactions and to extract powerful data-sets. This data can be harnessed for interventions as diverse as the distribution of aid and education in impoverished communities, to the identification of a particular social groups quality of cooperation.

While ideally used to work out what can be done to improve the social quality of such communities Christakis also discusses system understandings of machine learning's capacity to increase peoples' ability to solve complex network problems. Data modelling and feedback into online communities is however not deployed as form of smart altruism or to build social cohesion.

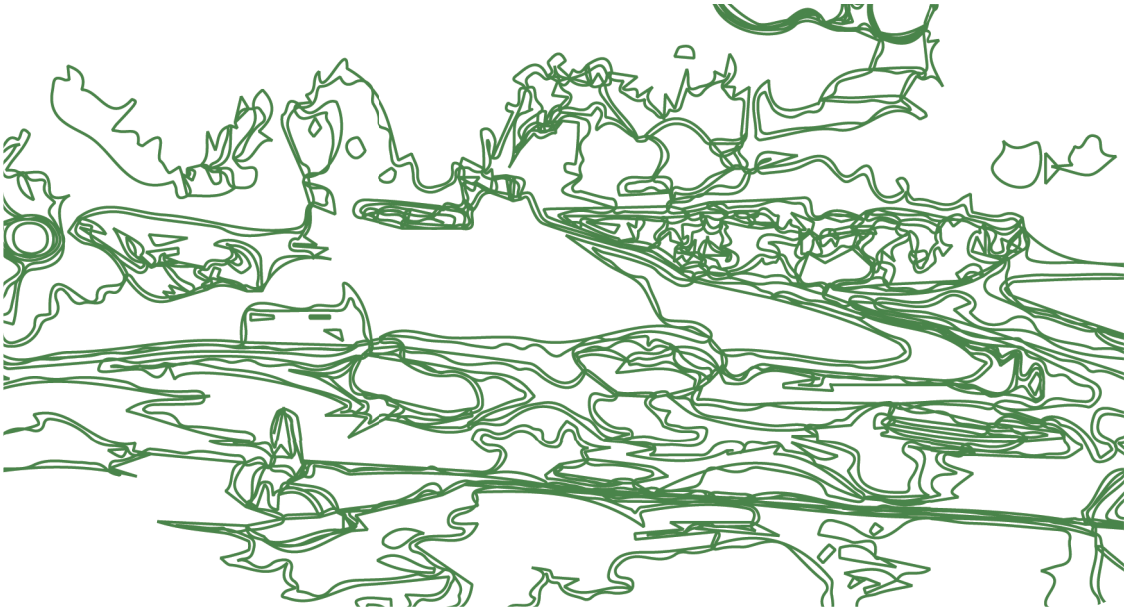
Helen Kennedy (2016) approaches the other side of the data analysis and manipulation by looking at social media data mining and identifies four significant problems: 1) it results in less privacy and more surveillance; 2) it is mobilised for the purposes of discrimination and control; 3) access to it is unequal, and this results in inequality; 4) it is methodologically troubling, because the data it generates shape the social world in opaque ways.

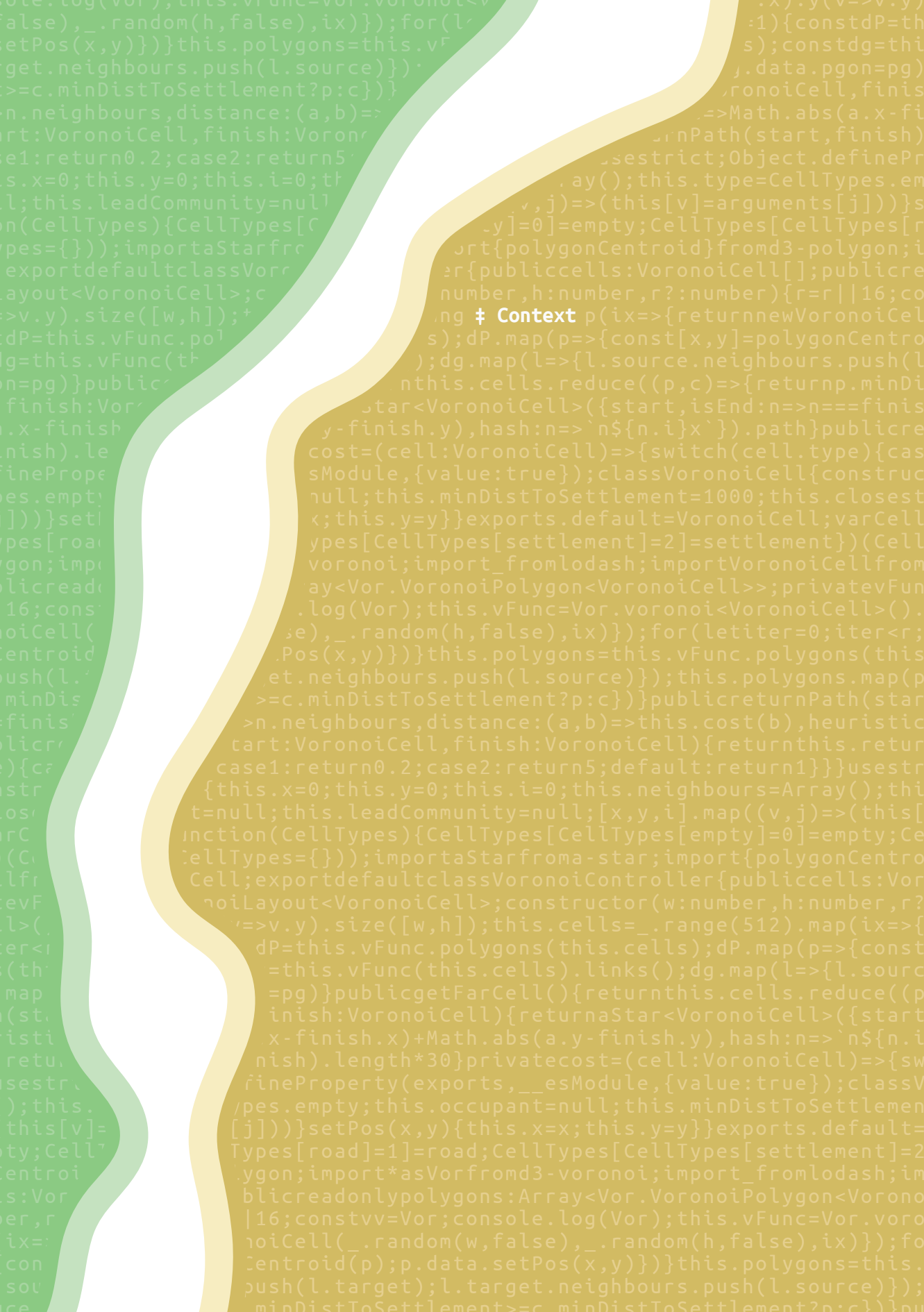
Kennedy suggests that organisations deploying data mining methods must be as transparent as possible, both in presentation, as well as in function to combat these problems. They must be inclusive and rigorous in identifying and avoiding their own biases, aware of the damage it may cause and the take their responsibilities seriously.

Speculative Digital Interactions

Due to the impenetrability, complexity and level of maturity of the social media sector, it is difficult to make meaningful difference to the nature of the current system through an isolated design project. Consequently, I have decided to take a speculative approach to the design problem. Speculative design is a useful tool for harnessing imagination to devise how certain design interventions could affect the reality we are in. It also acts as a tool to conceptualise how the world could be, and to prepare (back-cast) for such a reality (???).

Thinking of far-flung future possibilities is useful for design that attempts to tackle seemingly intractable problems. For instance, Wu, Ott, & Morie (???) speculate on a future in which long distance human spaceflight takes place. Their project “ANSIBLE” aims to combat behavioural health issues that may arise on these long trips with periodic communication with earth, through the use of virtual reality in both human to human, and human to object behaviours. Preliminary research, conducted as part of the HI-SEAS Mars simulation program, showed distinct advantages to “ANSIBLE” over more traditional asynchronous forms of digital communication.




‡ Context

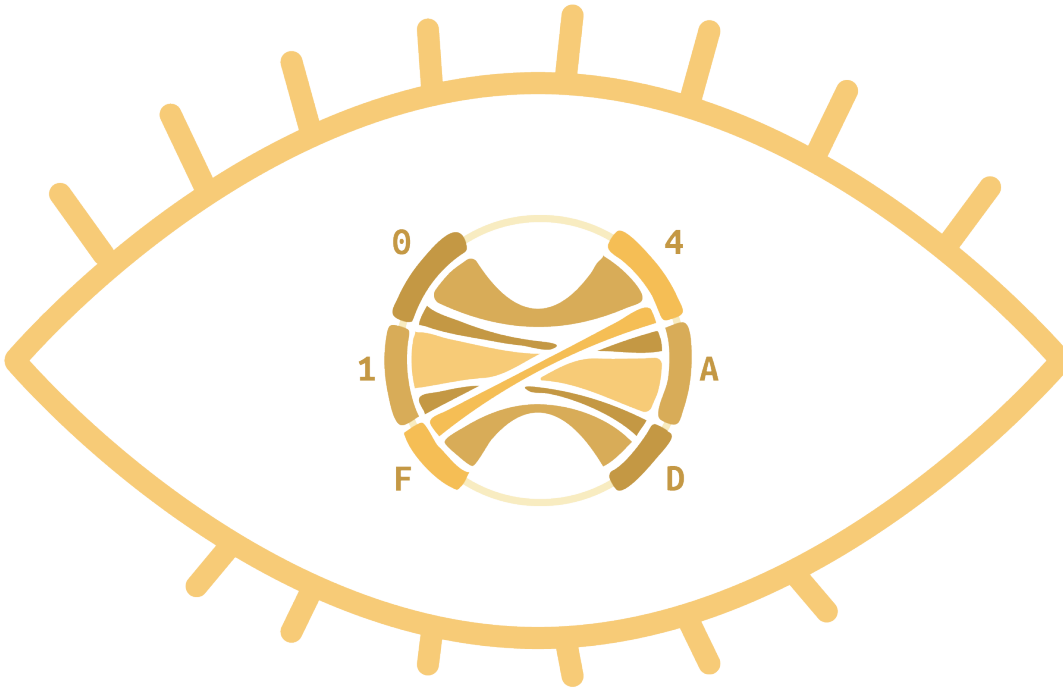
Introduction

This project aims to apply visualisation and interaction design techniques to complex digital social systems in order to interpret and understand the complex ways that we interact, and develop communities and identity online. By applying systems visualisation, design for play, interaction design, artificial intelligence, and speculative methodologies the project seeks to develop an interactive model allowing participants to experiment with online social phenomena and speculate upon their roles in it.

The project also endeavours to inform people about the kind of social models they take part in. I aim to encourage familiarity around how our social networks work at present, and speculate on how they could function in the future. I accomplish this by developing a interactive installation that allows participants, who through interaction, can explore the relationships between various elements that make up the model.

This includes actors in the environment which where created, and act based on parameters set by participants (referred to from here as User Analogues/Actors or UAs), completely autonomous actors/identities that were generated computationally depending on the state of the network (referred to as Computer generated Actors, or CGAs), and the interactions between these elements, along with super or meta groups of these elements (referred to from here as Actor Communities or ACs).

This project is designed to be situated in a moderated public or semi public place with an emphasis on learning or community development, such as a science museum or library. The project has a temporal aspect in that it develops and shifts over time, due to shifting nature of interaction with the project over time, as well as the evolving learning behaviour of machine learning.


0
4
1
A
F
D

Interpreting Complex Systems

Social Media behaviours lead to complex systems, which are often difficult for lay-users to understand. From the code and mathematics that allow them to function, to impenetrable terms and conditions that are confronting in their length, and use of legalese that sets out the terms of engagement in opaque legal structures. These systems are deliberately designed to be incomprehensible so as to lock users into participating in an interaction cycle that generally has a commercial return for service providers.

To abstract away the incomprehensibility of these systems the project looks to the application of various methods of “Data Visualisation” which presents systems in a much more human-friendly way. Data Visualisation consists of mapping data out graphically, similarly to how a cartographer would map out a landscape. The data can then be combined or compared with other systems or statistics, to identify trends or connections.

Nicky Case and Interactive Explanations

American Designer, Nicky Case, creates “Explorable Explanations”. These are interactive online guides to particular topics. Case uses narrative and play to share complex ideas that would be difficult to explain otherwise. They accomplish this through:

Starting With a Question

Define a question to answer, it should be gripping enough to keep people engaged.

Utilising Interest Curves

Introduce concepts at an engaging pace.

Starting Small, then Building Big

Don’t start with the most complex system, explain simpler systems, and then abstract them, for larger systems.

Author Guided - User Driven

Let the user control their journey, under the guidance of the author.

Doing, Showing, AND Telling

Combine interactive elements with graphical, textual, or other methods of conveying information.

Seeing, Modelling, then Applying

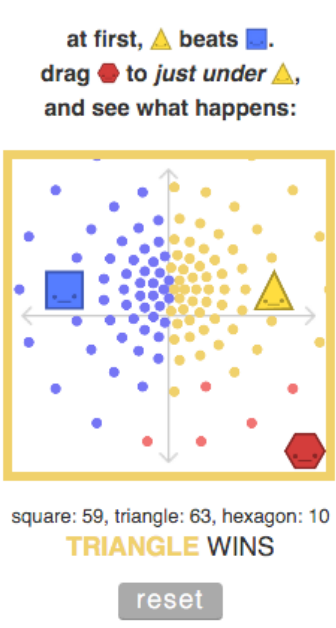
Show examples, let the user experiment, and try solutions, let those experimentations carry forward

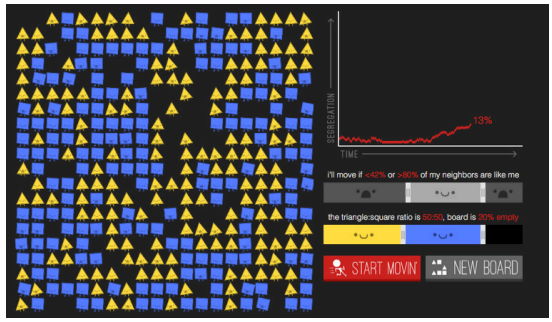
Cognitive Gateing

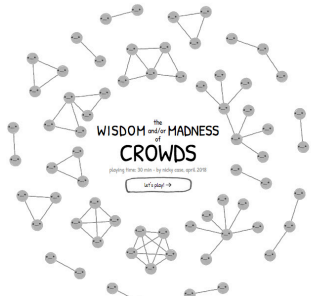
Hide more complex systems from users till they are ready.

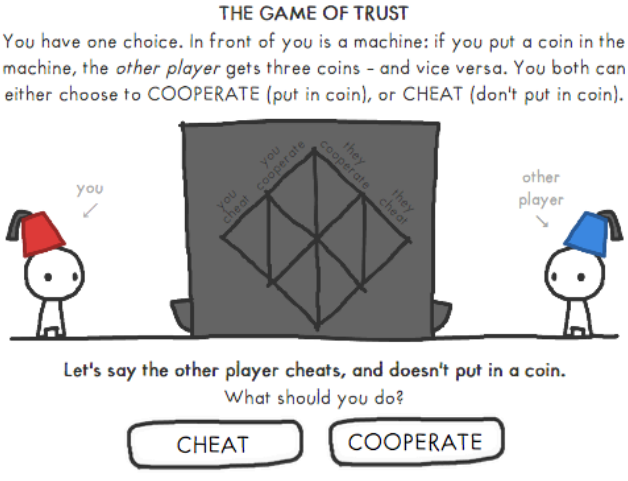
(Nicky Case, 2014a)

Nicky Case has used this methodology to produce new systems to explain everything from network theory and social contagions, Hebbian learning and Cognitive Behavioural Therapy (CBT), and Unconscious racial segregation. They also developed and maintain the Explorable Explanation Website (2015) which acts as a directory of similar interactive experiences that are available online.









Skeuomorphism and Learning through connection and metaphor

In graphical user interface design skeuomorphism refers to using familiar materials, actions, or objects as icons to explain or display things otherwise unfamiliar to the user. The classic example of this is the recycling bin/trash can on the desktop screen of a computer (the idea of "desktop" itself skeuomorphic). In the physical world, recycling bins hold items until they are emptied to be recycled, but there is a distinct association with these objects, and the act of destruction/deletion, that association persists when their likeness is presented digitally. By incorporating skeuomorphic principles into design, we can ease participants into to complex systems and interactions (Oswald & Kolb, 2014).

> "The sociomaterial-design agenda has implications for researchers and practitioners who are designing new coordinative artefacts for specific contexts, for example, when replacing paper-based systems with digital artefacts in a hospital setting" (Bjørn & Østerlund, 2014, p. 103)

Skeuomorphism as a tool, attempts to harken back to the material elements of an artefact and possibly regain some of the sociomaterial aspects that are lost when these artefacts were replaced with digital analogues. The sociomaterial aspects of digital activities and artefacts are primarily linked with the means of interacting with the digital systems. Due to the rapid development of technology, these devices are seldom designed with the full extent of their use in mind. They either focus on one aspect at the cost of others, or on a use case that is dated and irrelevant.

The result of this disconnect is a change in behaviour for the users who are forced to change their practices to fit these digital systems, sometimes at a loss (Bjørn & Østerlund, 2014). An example of this is how in academia, research papers and reports now must consider how they will be considered by internet search algorithms, which are constantly adapting and shifting based on a variety of factors, some of which are not disclosed by the search companies themselves (Orlikowski, 2007).

Simplification of Social Behaviour

Simplifying and modeling Social behaviour has been a particular focus of game theory, sociology and many other fields of social science for quite some time. These fields have developed a raft of systems, metaphors, and vocabularies to explain and discuss social networks. For example, game theory describes a particular problem called “The Prisoner’s Dilemma” which can be used to describe various social phenomena with a simple scenario. These kinds of abstractions can be used to model and explain social behaviour (Mathigon, n.d.)

Applications

Participatory Exhibits & Play

Interactive Exhibits have been a staple of science museums since the mid 20th century. San Francisco's "Exploratorium", set up by Manhattan Project scientist Frank Oppenheimer, was founded to develop and exhibit interactive exhibitions that allow visitors to explore and have tactile interactions with scientific and mathematical ideas, theories, and phenomenon. borrowing heavily from inquiry based teaching methodology, the Exploratorium, aims to increase science literacy and uptake through these experiences. (Exploratorium, 2014) This project aims to use this medium to increase interactivity, engagement and encourage serious play.

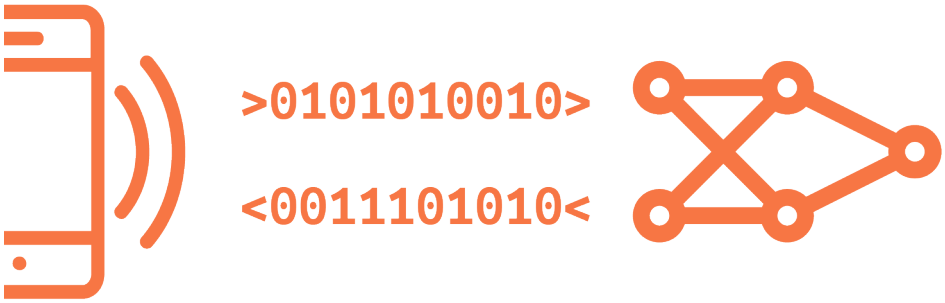
> ”We define serious play by three key characteristics:1. It is an intentional gathering to apply the imagination.2. It is exploring and preparing, not implementing.3. It follows a specific set of rules or language.” (Kristiansen & Rasmussen, 2014, p. 40)

Play is an incredibly powerful tool for the development of ideas and speculation of futures. Through modes of experimentation and communication, this project fits within definitions of “serious play”. As the underpinning contention is to develop healthy and happy social identities and communities, the project is entirely self-contained, and the systems and behaviours modelled do not need to be implementable in uncontrolled situations (as would be the case inside conventional social media platforms). Finally, the model of social behaviours and communities put forward itself acts as both the rule-set and vocabulary for play within the self-contained system.

Smart Objects

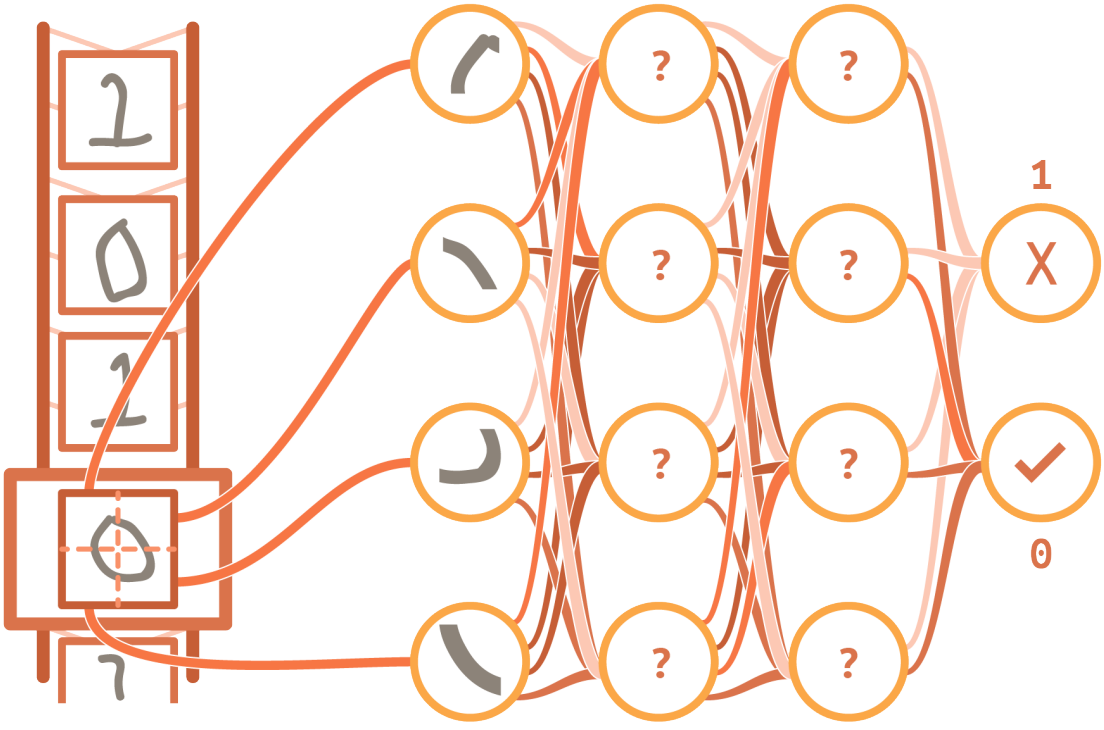
The project aims to interface online identity activities with physical IoT objects. Mesh devices and Bluetooth BLE beacons have become more and more common in a vast range of uses, from tracking devices to data mining and human flow tracking (Newswire, 2017). They have even started gaining use in museums such as the Guggenheim who are using them to share information about their exhibits and events (Jess Anderson, 2017).

Wirelessly Connected IoT play with the growing ubiquity of the plethora of connected devices in our lives, while serving a dynamic role in connecting digital lives with the corporeal. This leads to a break down of the barrier between a physical identity and an online identity. As more and more of ourselves are being quantified and shared as part of online personas, it becomes more difficult to distinguish traits in one realm, from traits in another.



Already digital objects are used extensively in the establishment of online identity, and it's connection to physical identity. Snapchat is a social media platform centred on sharing short-lived clips or photos with others, sometimes incorporating various "filters" that visually change the way someone looks (such as a dog ears or whiskers). In April 2018 Snapchat released the second version of their "Spectacles" these are smart glasses which have a camera allowing users to share what they are looking at to the platform, as well as being presented as a stylish addition to ones public presentation (Constine, 2018).

Another example is the various health and running trackers developed by companies such as Nike and Fitbit, these devices constantly record various health metrics from weight and blood pressure, to heart rate, and cumulative statistics such as daily step counts. These devices are closely linked with services that include social aspects that encourage openly sharing this data and competing with ones friends and colleagues over social media, and encourage the creation of community. Through the use of analogues to these kinds of devices, The project explores how they contribute to presentation of online identity and how changes in their use can affect online presentation.


1
0
1
0
?
?
?
?
?
?
?
?
?
1
X
0

Machine Learning and Neural Networks

In Nicholas Christakis' (2017) exploration of human networks solving complex problems, he found that by adding autonomous bots to the networks, particularly those that were imperfect, the problem-solving abilities of the network significantly improved. We all interact with bots online all the time, they manage and maintain our systems, moderate our messages and data.

This project defines artificial intelligence as a computing field that focuses on the use of computational tools that show some level of autonomy in their function. Commonly, this field deals with computer programs that can 'learn'. Machine Learning refers to the process of computational functions or processes, adapting their process automatically to better achieve the quantifiable goal they were created to achieve.

Machine Learning is often used with Neural Networks which are a type of computer program that behaves somewhat similar to how brains work. They are composed of various "Layers" including at least one input Layer, and one output Layer. These layers are composed of mathematical objects known as "neurons". Neurons each collect the output of the previous layer, then reduce these outputs to a single value between 0 and 1. Based on what is known as their weight matrix, they then pass that value on to the next layer, until eventually the output layer is reached, which then returns the output neuron's value to the user. This process is known as Feedforward.

Neural Networks are often used with a method known as supervised learning in which the networks are "trained" on a series of inputs (such as images of handwritten numbers) of which the correct output is known, and are asked to adapt their weight matrix so that their output matches the correct answer. They accomplish this through a complicated mathematical process known as backpropagation.

In particular, the use of genetic algorithms, and supervised learning with neural networks, can lead to interesting and emergent behaviours as they interact with User Analogues.

§ Methods_of_Reseach

Introduction

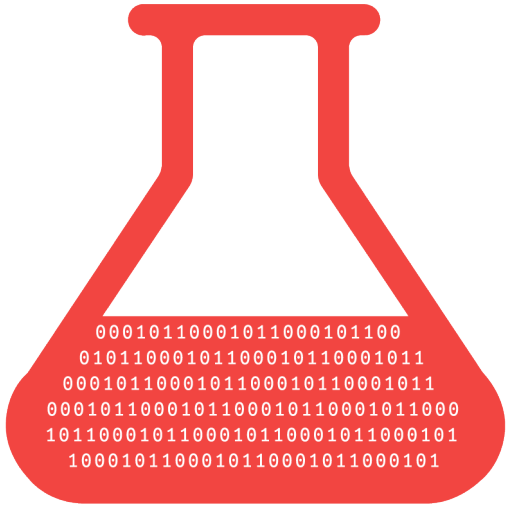
To explore the design opportunities in this project, a series of methods have been deployed to gain a better understanding of both the context of the design project, and experiment with what might be developed and how it might be applied it in practice. Incorporating a range of methods from digital design, and interaction design in particular, but also service/systems design. The three principal methods of primary research that I have employed are digital and electronic experimentation, product/service prototyping and testing, and journey/experience mapping.

Through experimentation I have iteratively identified technical methods and technologies to apply to my project. Through iterative user testing of a series of social network model prototypes, I have improved the quality of interactions and discovered emergent behaviours with in the algorithmic temporal mapping systems. To interpret how particular models developed apply to issues of online identity construction and re-construction, journey and experience mapping, and service design blueprinting have been used to track and analyse the connections between the technical processes and the user interactions. Through this process I have been able to identify technical inefficiencies, as well as interaction scenarios to avoid in future iterations.

Literary Research

Due to the social science aspects of my project, I have reviewed related literature to social networking fields as part of my secondary research. The literature reviewed addressed topics including online social behaviour, and communication, discussions on the presentation and subsequent value of identity in an increasingly digital world, and the data science of social media and the repercussions of its extensive mining for corporate endeavours.

I have also gained insight from experts in the field, such as experienced community manager and head of the Australian community management conference Swarm. Through this, I gained valuable insight into how that profession perceives and manages online communities, and what behaviours it defines as notable and worthwhile integrating into the project.



Experimentation - Qualitative

> **"Experimentation can lead to discoveries, new conclusions, and serendipitous innovation. Historically, experimentation has been a hallmark for designers, demonstrating creativity, astute inquiry, and a synthesis of ideas and methods." (Mark Baskinger, 2010)**

One of the most significant methods for synthesising and understanding a technique, medium, material or behaviour, is to experiment with it. If approached from a creative practice perspective rather than a scientific one experimentation can yield previously unknowable insights to inform design development. The approach taken in this project could be considered as a less deterministic form of rapid prototyping, in that restrictions of feasibility and purpose are set to one side at least temporarily in order to explore possible design directions freely.

Experimentation draws on methods from physical and digital prototyping, but differs in it's purpose, at its core the act of experimentation in the context of this project is about exploring assumptions and ideas through the act of digital making. This form of experimentation is not bound by the restriction of viability (Hall, 2011). Instead learning is centred on translating experiences from making and interacting with the digital medium. Nicky Case (2014b) used experimentation in their 2014 piece, "Coming Out Simulator" as a way of developing aspects of the experience quickly and freely, without having to rely on whether they would be feasible for the outcome.

Throughout this project, I have experimented with the look and feel of various digital and physical elements of the outcome. I have also experimented with more service/ system related elements such as the actions being modelled and the parameters of identity that I want to track.

To explore possible visual methods of presenting complex social networks I've experimented with attempting to recreate other visualisations, swapping out the original data with parameters and behaviours from my field. I've also experimented with new visual tools, such as combining Voronoi diagrams, Lloyds spacing algorithm and basis splines to create soft, fluid like visual groupings.

In regards to the data modelling aspect I have experimented with different methods of categorisation. These include letting participants decide on all the behavioural aspects of their UAs, compared to asking them to create an identity that fits within a preexisting group

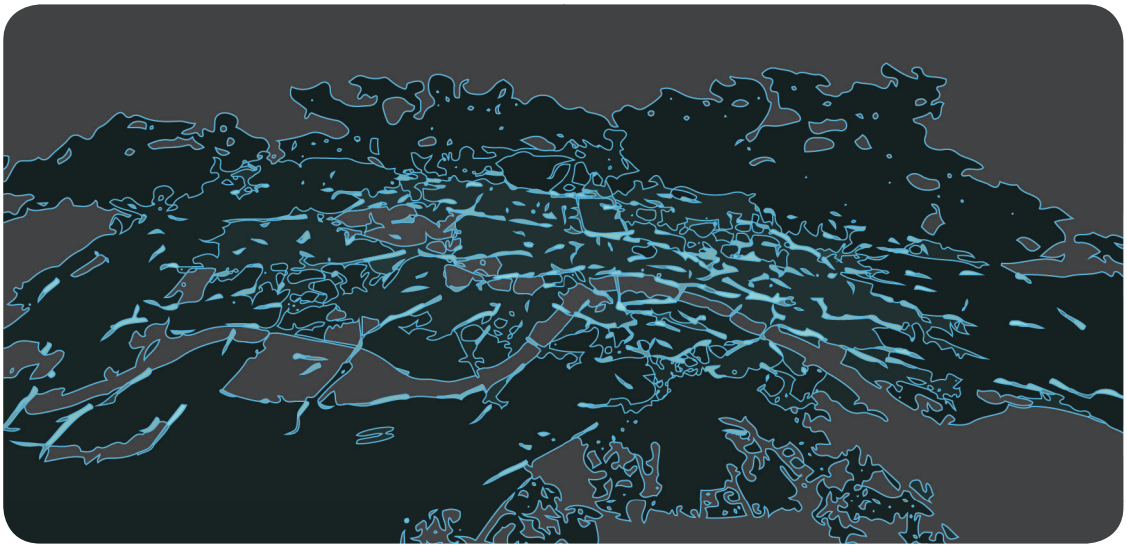
Prototyping and Testing - Qualitative & Quantitative

> **"In brief, prototypes used in system evaluation allow for adjustments to be made to system specifications. Prototypes as complete system specification provide a full and formal description of what a future system will do. Exploratory prototypes are rapidly made and disposable mock-ups that aid the clarification of system requirements. Lastly, cooperative prototypes mediate the capacity of both users and designers to formulate system requirements."**
> **(Wilkie, 2010, p. 141)**

In the case of this project, prototyping activity contributes in two key ways: generation of evaluation prototypes and enabling the generation of cooperative prototypes. Evaluation prototypes are used to analyse technical aspects of the project, from the physical parts and graphical interfaces to the code and parameters of the simulated models. Cooperative prototypes are used to gain user feedback, both qualitatively through discussion with users and through the tracking and recording of interaction data.

IDEO used evaluation, and cooperative prototypes while designing a new kidney vessel for transportation for the Organ Recovery System (ORS) In 2000 (kfaadmin, 2007, sec. 1). Rapid prototyping was used to evaluate and test the form and material choice of various design concepts to better understand whether they are fit for purpose or not. They produced both digital prototypes and physical prototypes in order to balance speed, fidelity, functionality, and freedom in the process. These considerations were also present in this projects use of prototyping. IDEO also developed and produced hi-spec prototypes to grade various engineering aspects such as mechanical and electrical parts and wiring. Some high-quality final pre-production units were produced for clinical trailing and presentation.

Due to the the three layered nature of this project, combining programmatic modelling, digital visual communication, and human computer interaction, the importance of high fidelity mockups, particularly in areas where these various layers interact is key to testing and ensuring the persistence of experience for the final outcome.



Creative Coding

Creative Coding is a programming practice that is focussed on using programmatic functions and processes as a generative form of expression rather than for pure function. Often, this is done in visual programming environments, such as Max, Grasshopper, or purpose build language/language subsets such as Processing or p5.js (Bergstrom & Lotto, 2015). With the rise of the maker movement and the growling popularity of IoT development boards, microcontrollers have become far more popular for visual applications. This newfound popularity has contributed to a rise in the use of the Arduino SDK, and Circuit Python to program these devices (Thilmany, 2014).

Finally Javascript, Python, and the Arduino SDK (a subset of the C programming language were used in the development of microcontroller and electronic elements of the project.

This project incorporates coding both for functional goals as well as incorporating this code with aspects of creative coding. Functionally, coding is required for the programmatic construction of the social network model the project incorporates. In the prototyping stages, the language javascript was used due to its flexibility in running on a variety of platforms, making user testing easier. Javascript is also used by the creative coding library p5.js which is used to develop visual aspects of the project as well as prototype interactions

Journey Mapping and Service Blueprinting

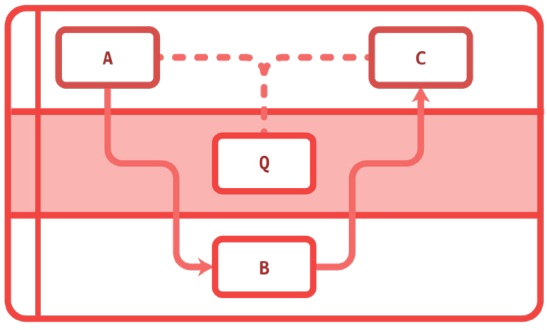
> "A customer journey map provides a vivid but structured visualisation of a service user's experience. The touchpoints where users interact with the service are often used in order to construct a "journey" – an engaging story based upon their experience. This story details their service interactions and accompanying emotions in a highly accessible manner." (Stickdorn & Schneider, 2011, p. 151)

> "Service blueprints are a way to specify and detail each individual aspect of a service. This usually involves creating a visual schematic incorporating the perspectives of both the user, the service provider and other relevant parties that may be involved, detailing everything from the points of customer contact to behind-the-scenes processes." (Stickdorn & Schneider, 2011, p. 201)

Journey mapping and service blueprinting have become staple methods of service design practice. Drawing on methods from systems and user centred design used inside industrial design practice for decades' prior, these methods can be broadly understood as modes of structured (abstracted) visualization used for the purpose of the analysis and critique of human-artefact-information-institutional interactions within particular settings, to inform designed improvements.

A case study for these service mapping methods was DesignThinkers work with the Dutch financial "NL Agency" to redesign how the company presented themselves and communicated with customers. Service design agency DesignThinkers used both journey mapping and service blueprinting to identify customer pain points, such as lack of two-way conversation happening between the customers and NL Agency. Additionally, they used these two tools to help communicate the issues NL Agency had, and the solutions that DesignThinkers Proposed (Stickdorn & Schneider, 2011).

This project uses journey mapping to explore the various actions taken by participants interacting with the project and how they might affect their experience with it. This includes how different contexts such as a science museum or library might affect the experience, whether they are interacting with it alone, or in a group, or whether it's the first time they are contributing to the model, or returning to see how it's changed since they last interacted. Service blueprinting is used to connect the technical aspects of the models and behaviours to the interactive and visual aspects of the installation. Blueprinting was used to identify superfluous elements, as well as core elements of both sides of the project to develop a more resolved outcome.



This project used experimentation as the primary method of developing a set of parameters and behaviours to represent in the model. Experimentation also helped develop ways of visualising and developing representations, as well as identify and emergent behaviours that come from the combination of these aspects. From there various systems and models were prototyped to gain feedback on the interactive aspects. Technical prototypes were also used to test technical aspects of the project. finally, blueprinting and journey maps were used to identify parts of the project that needed strengthening, and aspects that were superfluous and cluttered the direction of the project.


Δ Design_Development

System

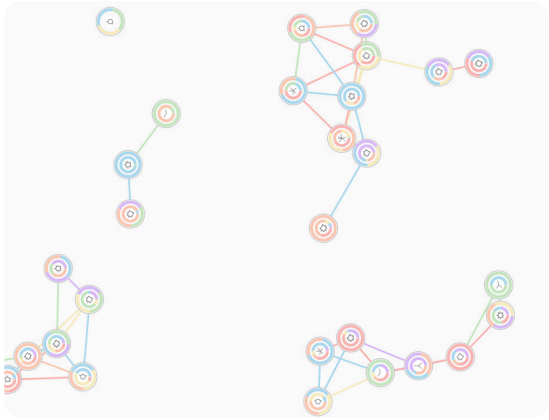
First Prototypes

Friendship Mapping

Completed during the first six weeks of the academic year, this experiment somewhat of a proof of concept of the idea of visually presenting social connections. Data was gathered in the form of a very light survey which presented participants with generated identities with a set of traits and asked them the simple question of whether they get along. This information was then collated together and presented using a “D3.js” based force node graph.

Links and Nodes

The first experiment conducted under the umbrella of social simulation was the links and nodes experiment. This system revolved around the measurement of interactions and memory, and how that would affect future interactions. Participants could create their own identities to insert into the system as nodes and apply to them a set of behaviours which dictated how they moved about the space.

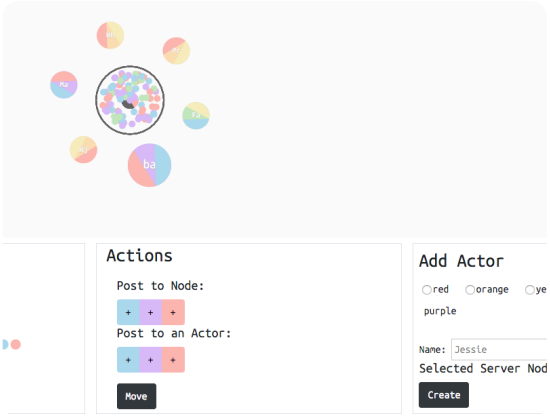


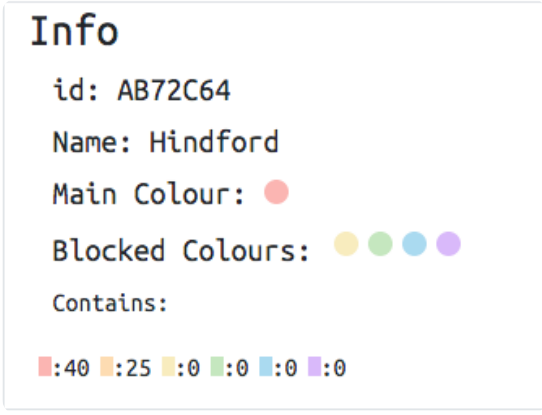
Messages and Clusters

The second experiment centred around the organisation of groups into discrete clusters which self-censored communication between its members and other clusters. The original premise of this was to mimic the communication methodology incorporated by the Mastodon social network platform.

Going Forward

It would have been a stretch to call these experiments successes, but they were insightful. The primary barrier I found to people interacting with complex digital systems was a lack of desire to establish oneself in the system, and due to a complex interface, a lack of ability to Interact with the system.





The Community Model

All the previous experiments lacked a key indicator of value and differentiation. Behaviours were confusing and unpredictable, leading to difficulty in theorising problems. Social behaviour is complex. Nicholas Christakis's work (Christakis & Fowler, 2009) gives a framework for basing social interaction and behaviour on connections and relationships between individuals, but his work does little to address relationships between individuals and entities such as online communities and cultures. While Hogan (Hogan, 2010) used a system of performance and exhibition to frame behaviour and discussed how behaviour online is different to more standard social behaviour. Both of these ways of analysing social behaviour simplify identity and belonging. These are key concepts in an online social environment.

A focus on social broadcasting (indirect social communication, presented as a direct form of sharing, think Instagram's, Snapchat's and Facebooks's Story Features) has made self-identification and internally consistent presentations of identity paramount. Social belonging is not a new priority in social communication, but the online algorithms that dictate our social behaviour punish those who have been set categories that they do not fit in. To solve this, I looked at community management. Vanessa Paech is an expert in online communities. She describes the values that communities give to their members: - Identity - Emotion - Expression - Mobilisation - Relevance. People gravitate towards communities that provide:

Belonging

Influence

Fulfilment

Connection

Finally, thriving communities thrive through having established:

Identity

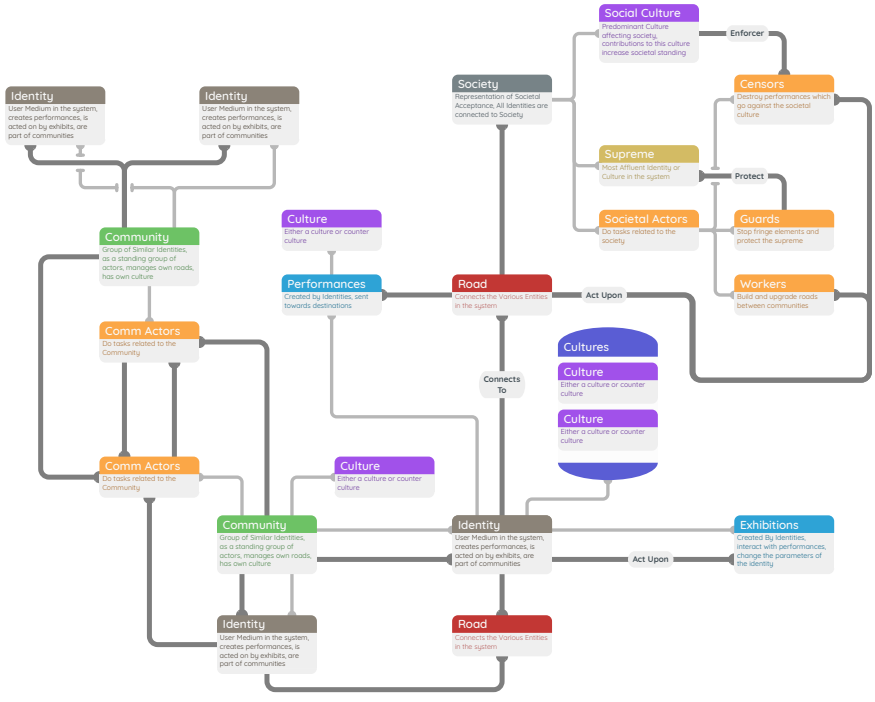
History

Ritual

Social Order

Progression

(Paech, 2018)By combining this model with the previous two models, we have a more comprehensive framework of which to build the system from.



Node, and Other Technical Aspects

From the beginning, the plan has been to write the majority of the system and visualisation code in javascript, which has remained true throughout the whole project with two exceptions. Firstly that the code is now written in typescript, which is a language that is very similar to javascript and compiles to it for runtime. Lastly, the code for the raspberry pi's has moved from javascript to python due to library availability for the components used.Javascript was used for three primary reasons. It was a familiar language for me to write in due to having used it in previous projects. JS is a web language, and JS code runs on any modern computer or phone without downloading a binary file, and finally that it has an extensive list of comparable packages that are easy to access and use through NPM (node package manager). In particular, the D3.js library has had a considerable role in this project. (See Visualisation)

A Modern Identity

The range of online social behaviours and categorisations is as vast as human nature itself. So a simplified model had to be developed. The model had to have enough detail so that the identities could be distinct from their inception into the system, while also being simple enough to be implementable into the scope of this project. The metaphor or hashtags is used to plant the service in the digital space. We already use hashtags to categorise ourselves and our content.

Interaction

Object Mapping and Projection

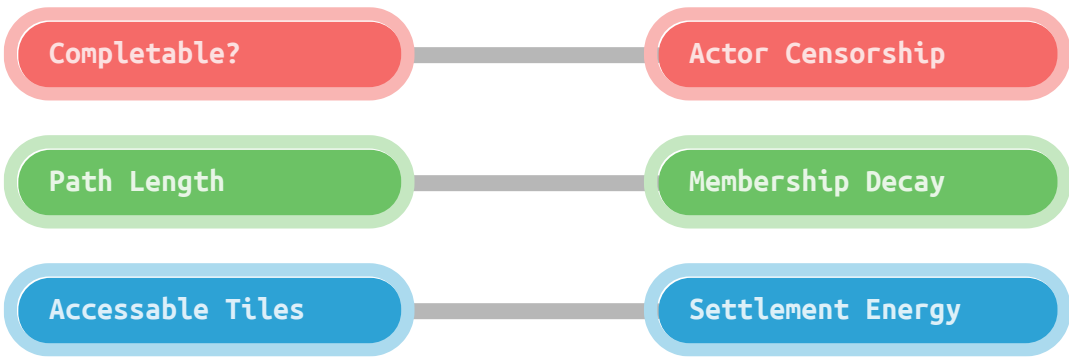
Coming into July the physical design consisted of a central table, projected to from above and with position mapped smart objects resting atop it. The original object set consisted of objects to complete three main tasks.

These tasks were system input, system analysis, and system modification. For input, I wanted to visually reference technologies such as biometrics and fingerprint scanners in its form. The position on the table that the object was, would be the position the identity would incorporate. For analysis I was inspired by eye loupes, often used for looking closer at everything from paintings to electronics, that action I wanted to incorporate into the system. If you wanted to know specific information about a particular object, bring the loupe to it, and it would oblige your curiosity.

Modifying a System

The modification interface was always a tricky one. Originally intended to be a static control panel physically connected to the projection surface. I wanted help to create the feeling of tweaking and perfecting something. However, a control panel is bland and uninspired. After ruminating on it for some time, I came up with the concept of using a maze

.Maze's have been used to represent complex difficult to understand systems in the past. So what better a metaphor for this system. The design of a circular maze in which the walls can become corridors and vice versa allows a considerable amount of complexity in the parameters of the maze, which can then be converted into parameters of this system.


Completable?
Actor Censorship
Path Length
Membership Decay
Accessable Tiles
Settlement Energy

Separating the objects

While spatially aware objects allow for more creativity in interfaces, it quickly becomes evident that the scope of the technical requirements to achieve that interface were vastly outside the scope of the project. The objects were then adapted to function with much more autonomy. These objects changed several times before settling on the current Vessel and Maze forms. Due to these constraints, the Loupe object was dropped from the project.

Receipts and Ticker-tape

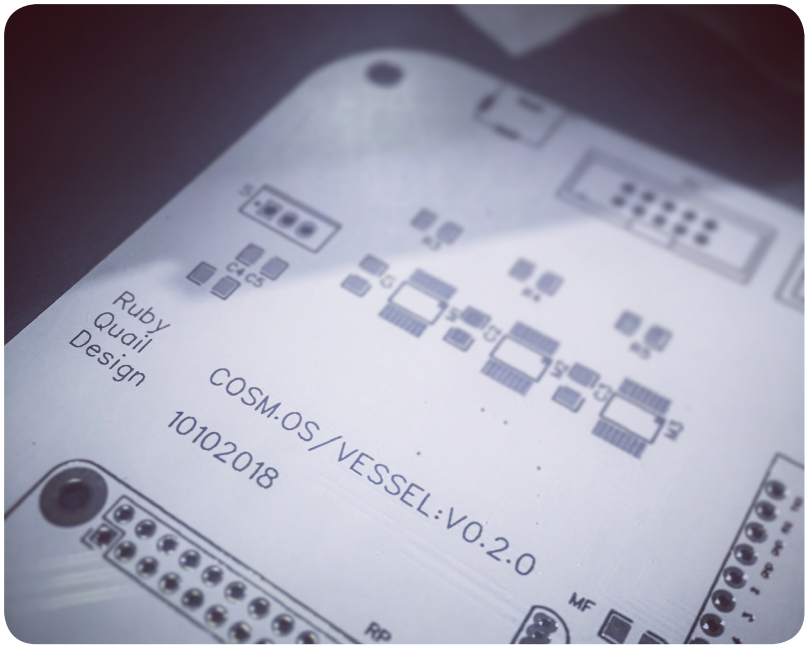
With the Loupe object gone, there was a need specific information to be made available. This was rectified through thermal printing. The Vessel object is equipped with a thermal printer which on the creation of a settlement, prints out a “receipt” which contains information about choices made, an image of the settlement and its USID (Unique Settlement Identifier, A set of three letters or glyphs that represent the settlement). The projection space also has a thermal printer; however, this one is responsible for continuously printing a record of events, creating a literal paper trail of the system.

Vessel

The Vessel object is the primary interaction point between users and the system. It is interacted with through a small OLED screen, five push button knobs for selecting tags, a thermal printer for the receipts, and an infinity mirror. The infinity mirror acts to attract people to the object and represents the gateway to the system from the physical world.

Neural network manipulation

Another planned but ultimately scrapped object was the neural network interface. This was designed to emulate the standard way of displaying the nodes and weights of a neural network. I wanted to give users a chance to play around with how a neural network works at a very surface level. The weights and biases of the neural network interface are meant to affect the behaviours of the actors. Due to time constraints, the development of this interface was determined to be outside the scope of this proof of concept.



Electronics

Once the objects were determined to communicate wirelessly with the system, the electronics needed to be developed. Due to parts required, microcontrollers were a must. For both objects, a raspberry pi zero w was chosen as the central brain behind the objects. The principal reason for this decision was the built-in wifi, and the ability to run Node.js programs. Initially, all the programming for the objects was going to be Node. This was to maintain consistency with the system, which also ran on Node. However, due to difficulties with library compatibility, the raspberry pies were instead programmed using python. Both objects also incorporated a smaller Arduino micro-controller (the Spark fun pro micro was used due to familiarity with the 32u4 board, and the small size). These performed more low-level tasks such as loading and preformatting input coming in and driving the LED Arrays.

Bright RGBW SK6812 LEDs (also known as DotStars) were chosen for both the infinity mirror, and the maze. While I could have gotten away with using simple RGB LEDs for the infinity mirror, the bright natural white phosphor helps brighten the whites and makes it clear when a selection has been locked in.

Simple 12 pole Rotary Encoders were used for Vessel, while the Maze used 10kΩ potentiometers for a more precise feeling of control. For the Vessel's screen, a 128 x 96 1.03" monochrome OLED was chosen, mostly for compatibility and simplicity. For the printers, Adafruit printer guts were chosen due to their flexibility of placement, and availability of comparable libraries.

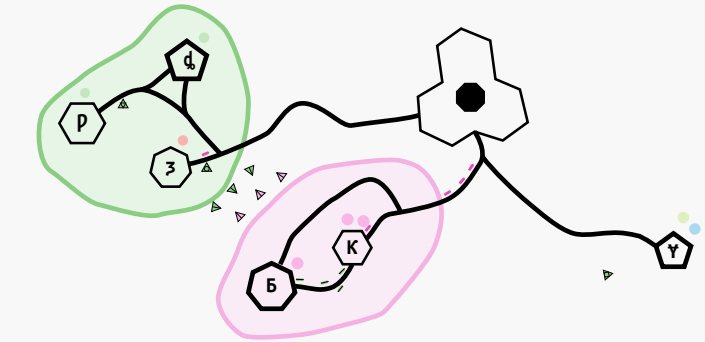
All the components were mounted on custom-made PCBs. Overall four were designed: A small board to mount the LED for the infinity mirror panels, a board to mount the Rotary Encoders, and main boards to house the microcontrollers, voltage regulators and connectors.

Visualisation

Cartography, Colour, and Scale

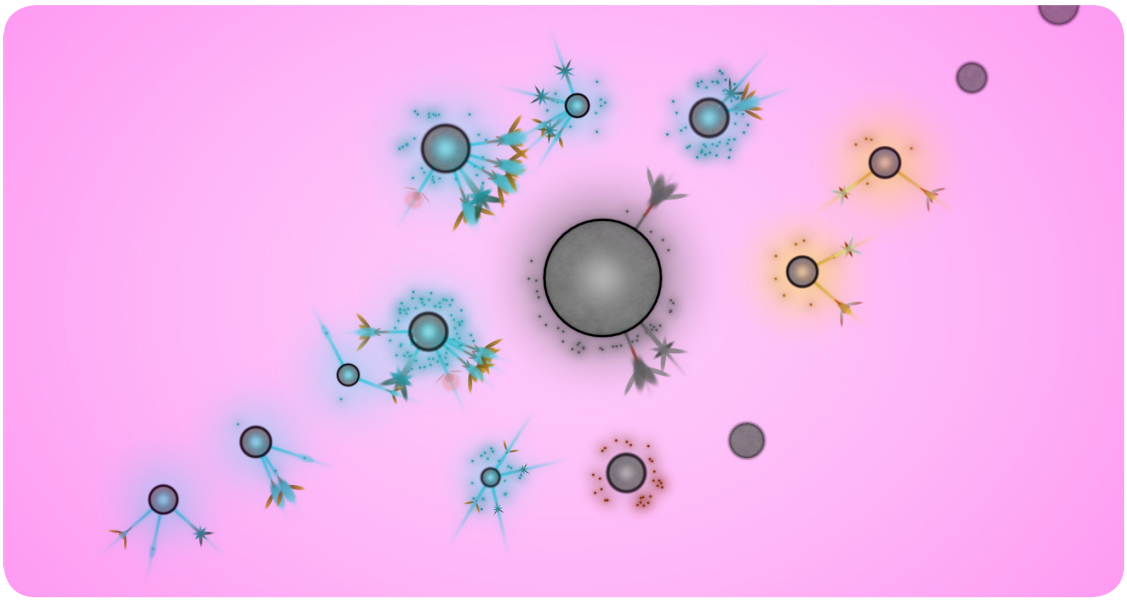
How to visually present the system was also a consideration of this project. The visual design needed to both be flexible enough to display the complexity of the system's elements, as well as simple enough to follow visually. The original experiments were presented very abstractly, and in turn not only hid a lot of information but were very difficult to understand. For the final design, a cartographic metaphor was chosen to help display the system. This visual language gives a stronger starting place to users of the system to understand what's happening as many of them would have seen and used standard maps before.

Another consideration was colour. Colour is an excellent tool to enable differentiation of various categories and aspects of the system. For this project a light pastel like colour palette was chosen. This allows for variation in colour, while still being distinguishable and not too bright that it becomes overwhelming.



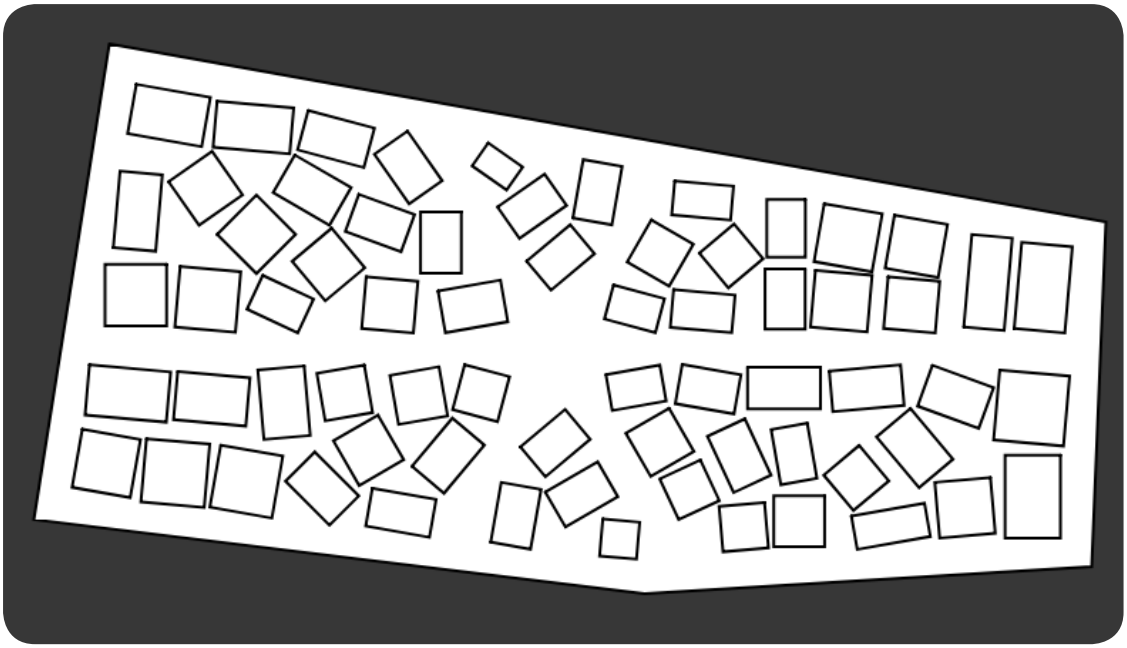
Another area of inspiration were strategy video games, complex strategy games such as Omni Systems Limited's Eufloria (Omni Systems Limited, 2009) have to communicate a lot of complex information about the state of the system, quickly and without misinterpretation. Eufloria does this through colour, shape, and scale. Relying on pre-existing connections users may make to different aspects (e.g., spiky things associated with violence, red with danger, blue and green with health and healing). This decision reduced the amount of text on the screen allowing for cleaner visual design, and one with reduced movement inducing a feeling of calm and zen within the experience helping the experience feel more pleasant.

In order to bring a similar feeling of zen into the project, it too visualises information through shape and colour instead of text. Another valuable effect of this is it removes the implication of an orientation to the visualisation, allowing it to be viewed from any angle.



Parametric Geometry

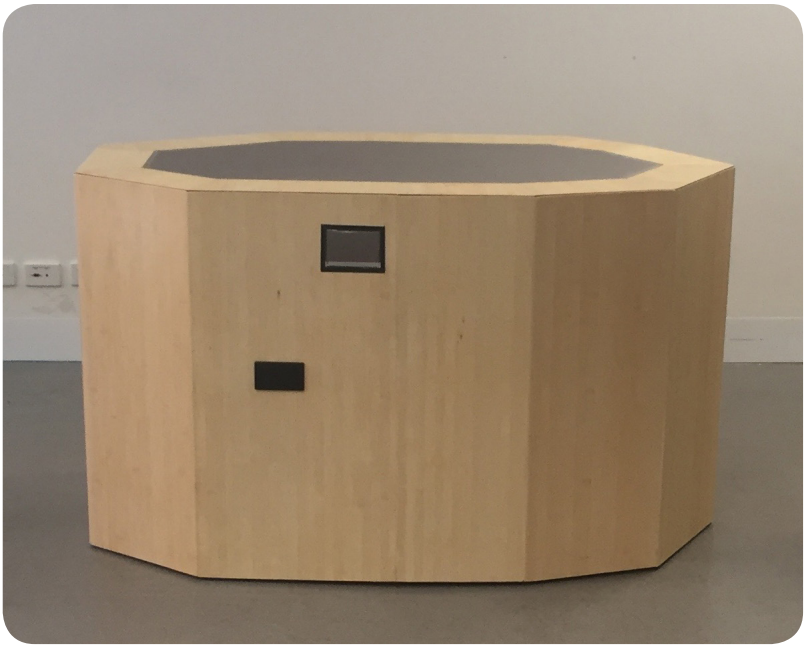
One method that was experimented with was using parameters of the various actors and identities, to inform their geometry programmatically. This method allows for a higher density of information to be included in the objects, but was computationally intense, and was difficult to pass quickly. These problems led to the idea being scrapped but may be revisited in the future.



The Grid and Data Visualisation

The whole visual projection is placed upon a generated Voronoi grid that has been smoothed and spaced using Lloyds algorithm to create a regular yet still somewhat chaotic layout. This choice creates a more organic looking system as there are limited right angles, helping the background elements recede away from the more regular foreground elements.

Physical



Projection Table

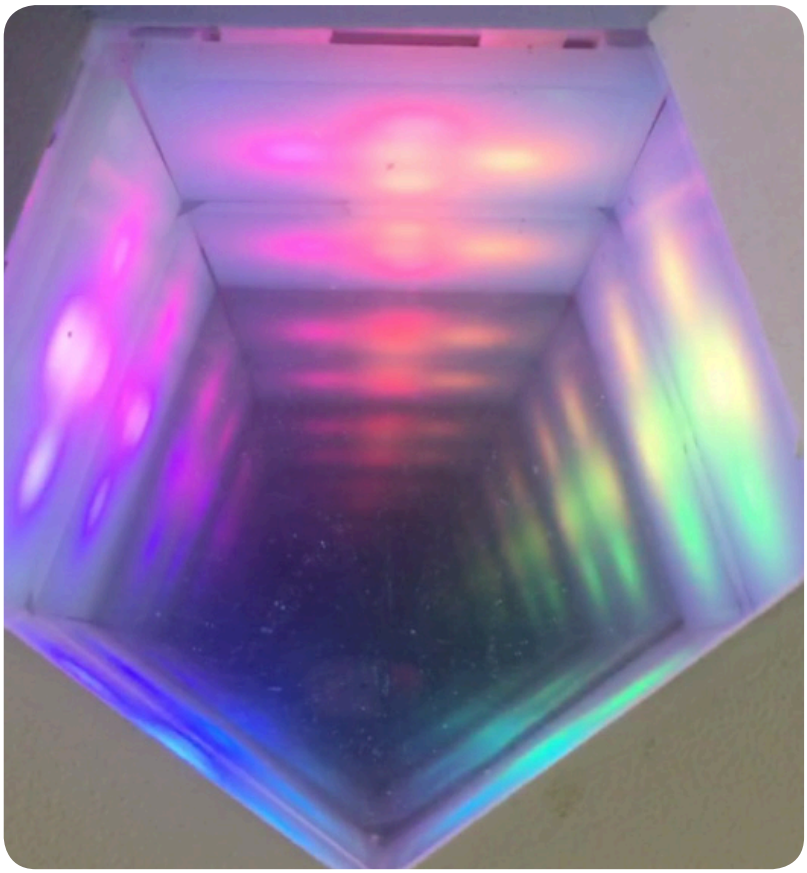
The presence of the installation was another consideration of the project. Originally envisioned to be centralised on a central table, the design contained a large projection screen, with the visuals projected on it from beneath. This screen has remained through the revision to a more distributed object-based installation. As the project is designed to be placed in a dark setting, the light that this projector creates escaping through the screen brings an exciting visual focus to space. The somewhat elliptical shape of the table allows it to display the majority of the 16:9 projection ratio, while still freely allowing it to be viewed from all directions. In an exhibition setting this allows multiple users to surround it and makes it easy to point out, discuss and explore aspects of the visualisation with others, this enables a much more collaborative method of engaging with the system than would otherwise be possible.

Plinths

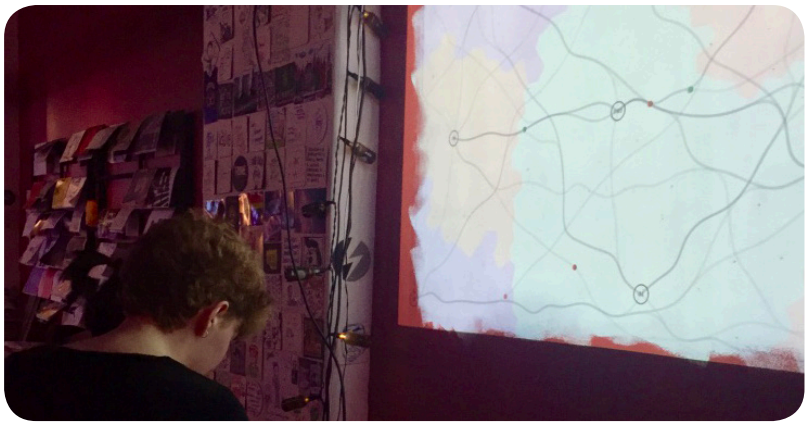
The Plinths, on which sit the objects of interaction, were initially intended to act sort of like homes for the objects, a sort of symbolic tether between the system and the real world. Similarly to the table, they are designed to appear to sit on the edge of the physical and the digital, the blocky geometric design of the plinths references the digital visualisation of the system, while the materiality of the bamboo, connects the plinths back to the world.

Use of light

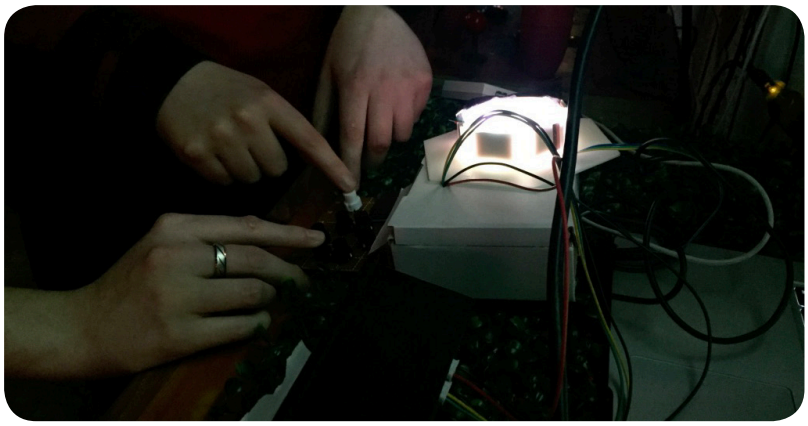
In the plinths and table, as well as the interaction objects, light is used to symbolise the digital. Both the plinths and the table are backlit surfaces, giving the effect that the information they display, as well as the objects placed on them, are rising out of the light similarly, the softness of Maze's lights, and the infinite tunnel of Vessel's mirror arrangement, symbolically are glimpses of the system in motion.



≈ Methods_of_Evaluation



To evaluate the success of this project I am taking a two pronged approach. Firstly, to evaluate the outcome on its technical capacity, and lastly to evaluate the project on how well the concept answers the brief. Due to the complexity of the desired outcome, iterative testing and evaluation was pushed outside the scope of the project. This project is being positioned in way that is difficult to evaluate outside of it's functionality. The unreliability of the systems being modelled, and the rudimentary level to which they are being simulated leads to difficulty in identifying their accuracy. Therefore the technical side of this project is better evaluated by it's purely technical capabilities. Conceptually the project is evaluated both on its ability to reflect aspects of complex social systems back at the user, and on its ability to engage and enthral.



This project presented many, many challenges to overcome in both areas. Technically this project encompassed electronic design, digital design, installation and interaction design, systems engineering, and software development. Additionally, a significant amount of programming was required, both for micro-controllers and more traditional computers, and in a variety of different languages, and using multiple different communication methods. Conceptually the project needs to have the capacity to represent at least a nominal level of social behaviours and systems, and through these behaviours, communicate it's internal logic. It needs to be engaging and needs to encourage users not only to engage with the system but also to be curious about how the system works, and its various elements contribute. Finally it also needs to be fun and enjoyable for the users to interact with.

In many ways this project functions as a proof of concept for further development in this area. Many of the goals this project is trying to achieve, are outside what is capable of being accomplished in a year. Common tools of evaluation, such as user testing, were unable to satisfactorily be conducted in the time frame. The intention of this project is to be iterated on and developed over a longer time period in the future, and as such the criteria of evaluation (particularly conceptual) takes this into account.

Technical Evaluation

The project, whilst looking at reimagining the complex system of social interaction, is itself a complex system. For the project to achieve it's desired goals at the conclusion of this stage of it's development, it needs to be capable of..

- **Interactivity, users should be able to interact with the system and the system should react to these changes and reflect them in some way back to the user**
- **The ability for users to describe an identity, and the system to incorporate it**
- **The ability for users to adjust the running parameters of the system itself**
- **Reliability, the system needs to be able to run for extended periods of time (up to 12 hours without crashing or having it's function impaired)**
- **The interactive objects need to communicate with the system reliably.**
- **The interactive objects need to function electronically and seamlessly**

Specifically this means…

- **The system code needs to be able to deal with unexpected states without crashing**

- All elements in the system need to behave in accordance with how they've been designed to behave
- All elements that need to be visually presented to the user need to do so in a clear and distinguished manner.
- The System needs to maintain a speed of 12 cycles a second, or if that isn't possible, it needs to be able to compensate for the slower speed.
- The systems visuals need to be able to be clearly and completely be projected onto the table screen.
- Users need to be able to point out and discuss the system with others while viewing or interacting with it.
- The logger needs to be able to create a private wifi network and communicate with the vessel and maze objects over it.
- The system needs to be able to receive, and act on information it receives over the socket.io connection
- Both the maze and vessel objects need to be able to communicate their state to the system
- LED boards need to effectively transfer signal and power to all LEDs
- LED's need to illuminate correctly, and with low latency
- OLED screen needs to display text and graphics clearly without glitches
- Rotary encoders need to be read and interpreted quick enough so that no inputs are missed in standard use.
- Thermal printer needs to promptly print receipt, accurately, and cleanly
- All LEDs in the maze need to correctly illuminate to display current layout
- All potentiometers need to work accurately to allow precise manipulation of the layout
- Maze code needs to correctly path-find the maze and identify accessible, inaccessible tiles and possible routes

Conceptual Evaluation

In essence, this project aims to enable the production of collaborative speculative fiction. It creates a technical backend and ruleset for the fiction, and the participants, through interaction, to develop its narrative. This aspect of the project is significantly more challenging to evaluate than the technical side due to the chaotic nature of the behaviours being modelled. Ideally, and going forward, the project would be evaluated through iteration. The technical backend of the project allows for hot swapping of interaction objects, as well as the addition of new ones. The installation, therefore, could regularly be adapted and updated, based on feedback gained through observing the interaction between users and the system.

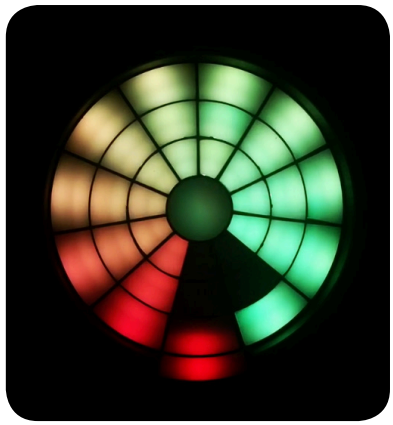
In order to meet its conceptual goals, the project endeavours to create a sense of curiosity in its users. As it is a proof of concept, to evaluate it on its ability to create a sense of reflection on one's behaviour, pre-supposes that the young simulation would generate a fiction that mirrors one's own experiences with social systems. In order for the project to meet this goal, it would have to possess a great deal more complexity than would have been possible within the confines of this project. Therefore user feedback of curiosity and discovery in an exhibition context are more valuable points of evaluation.


‹ Project_Outcome

Project Outcome

The project outcome as it currently stands consists of two interactive objects, a table on which the simulation is projected on, a router/logger, and plinths to support the two objects. The first object, Vessel, is a 3D printed white painted PLA device consisting of a monochrome OLED screen, a thermal printer, five rotary encoder knobs, and an infinity mirror assembly. The assembly consists of 5 LED printed circuit board modules, each containing 5 RGBW LEDs and a laser cut acrylic light diffuser, along with a one-way mirror and regular mirror. Controlling these is a Raspberry pi Zero W and an Arduino pro micro (uses an Atmel 32u4

The second, Maze, is a 3D printed black ABS device that loosely resembles a circular maze. Maze has ten potentiometer knobs, and a circular array of diffused acrylic panels hides 62 RGBW LEDs which illuminate and go dark to represent the corridors and walls of the maze. Maze is also controlled by a Raspberry pi Zero W and an Arduino pro micro.



The simulation is a node.js program running on a MacBook Pro. The MacBook also runs the visualisation which is an HTML canvas running in chromium. Inside the table is raspberry pi which hosts a wifi network and DHCP server, and logs the output of the simulation to a thermal printer

The simulation simulates a subset of settlement behaviour, including actor creation and community alignment, actor behaviour and connections between actors.


Ω Conclusion

Conclusion

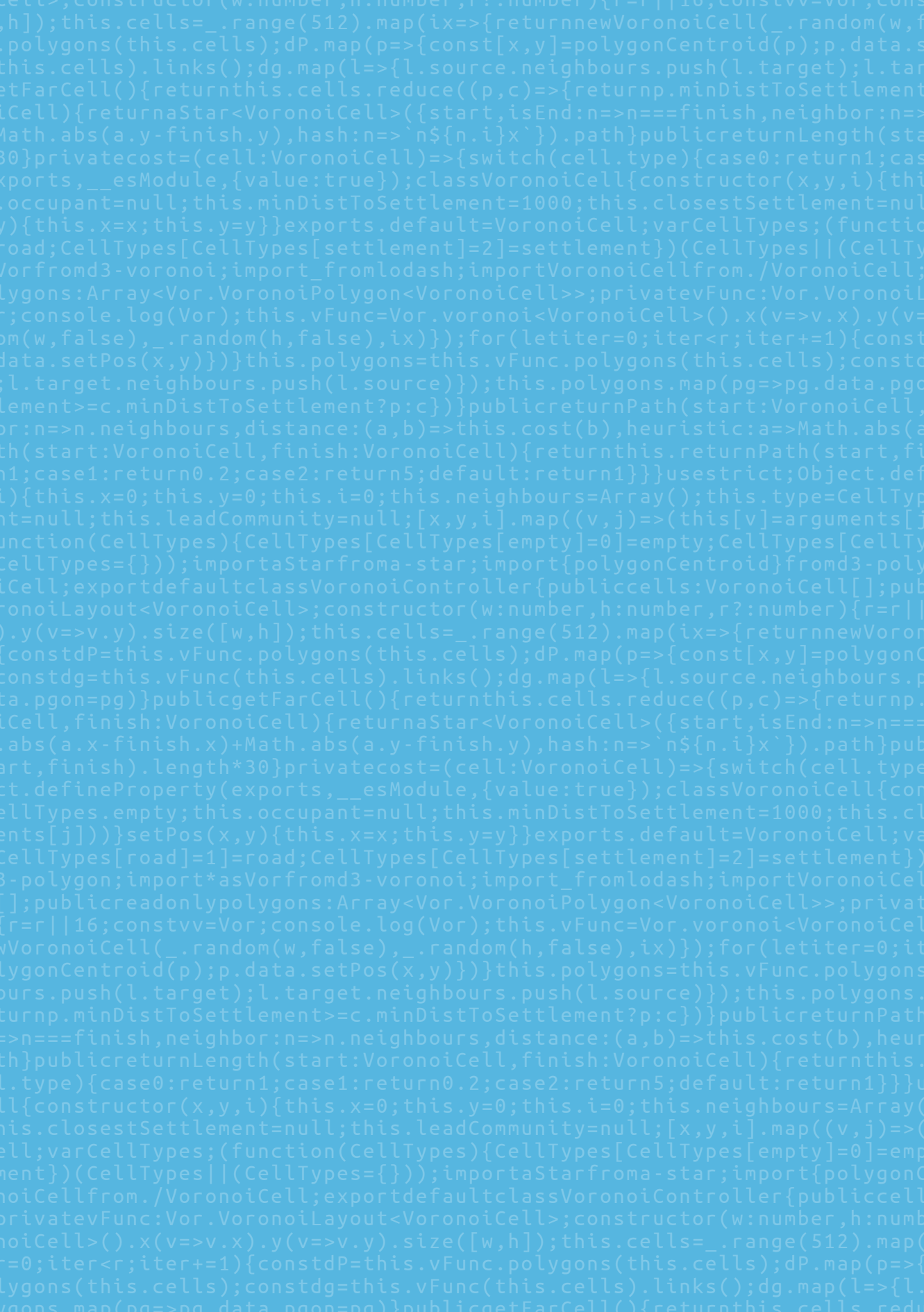
In its essence, this project is just beginning. “cosm.os” borrows from a considerable range of fields and modes of practice. At present the outcome only really addresses the breadth of the project, without yet achieving the depth required to fulfil its final goals of reflection and inspiration truly.

Many elements could not be implemented in the project’s time frame. These include - different types of connections and memberships, giving more depth to settlement relationships, - more specialist actor relationships, - more detail in community structures, - and many more social frameworks that could be reinterpreted into the system. Due to the modular nature of the system, many more objects could be developed allowing for further experimentation with interaction methods.

In order for this project to approach its potential, it needs to be able to be iteratively developed over an extended period of use, partly due to the complexity of the systems it is trying to emulate, but mostly due to how young the field is. Not only does our understanding of online social behaviour change regularly, but the behaviour itself does as well. In order for a stable, reliable system that adequately addresses social issues within the simulation to develop, it needs to be able to be adjusted and tested continually. Additionally, as the technologies and design practices that are applied to social media change, so too should the technologies that comprise cosm. os match the current online social environment of it's users.

This project as it stands was developed for the display environment of a graduate exhibition. The way that it is interacted with, in this environment, would be different from how it might be exhibited in a more public setting such as a public library, or a bar or cafe. Moreover, that would be different from a conceptual art exhibition, or an educational installation. Some objects would perform better or worse in particular environments, and a further developed project should be developed to be adaptable to more display situations

As it stands, the project still functions as an entry point into this world of interactive system simulations, and a proof of the capabilities of this form of systems communication to complex social systems.



◊ References

Anthony Dunne. (2013). Speculative everything: Design, fiction, and social dreaming. The MIT Press.

Bebawi, S., & Bossio, D. (Eds.). (2014). Social Media and the Politics of Reportage. London: Palgrave Macmillan UK. https://doi.org/10.1057/9781137361400

Berberick, S. N., & McAllister, M. P. (2016). Online Quizzes as Viral, Consumption-Based Identities. International Journal of Communication; Vol 10 (2016).

Bergstrom, I., & Lotto, R. B. (2015). Code Bending: A New Creative Coding Practice. Leonardo, 48(1), 25–13.

Bjørn, P., & Østerlund, C. (2014). Implications of Sociomaterial-Design. In Sociomaterial-Design (pp. 103–106). Springer, Cham. https://doi.org/10.1007/978-3-319-12607-4_10

Carrier, L. M., Spradlin, A., Bunce, J. P., & Rosen, L. D. (2015). Virtual empathy: Positive and negative impacts of going online upon empathy in young adults. Computers in Human Behavior, 52, 39–48. https://doi.org/10.1016/j.chb.2015.05.026

Case, N. (2015, March). Explorable Explanations. Explorable Explenations. Directory, http://explorabl.es/.

Christakis, N. A., & Fowler, J. H. (2009). Connected: The Surprising Power of Our Social Networks and How They Shape Our Lives. New York: Little, Brown and Company.

Click, M. A., & Scott, S. (2017). The Routledge Companion to Media Fandom. Milton, UNITED KINGDOM: Routledge.

Constine, J. (2018, April). Snapchat launches Spectacles V2, camera glasses you'll actually wear. TechCrunch.

Exploratorium. (2014, July). Our Story. About Page.

Goffman, 1.-1., Erving. (1969). The presentation of self in everyday life. London: Allen Lane.

Hall, A. (2011). Experimental Design: Design Experimentation. Design Issues, 27(2), 17–26. https://doi.org/10.1162/DESI_a_00074-Hall

Helen Kennedy. (2016). Post, mine, repeat: Social media data mining becomes ordinary. London: Palgrave Macmillan UK : Imprint: Palgrave Macmillan.

Hogan, B. (2010). The Presentation of Self in the Age of Social Media: Distinguishing Performances and Exhibitions Online. Bulletin of Science, Technology & Society, 30(6), 377–386. https://doi.org/10.1177/0270467610385893

Jess Anderson. (2017, February). The icon of modern art puts Estimote beacons on display. Reality matters: Estomate Team Blog.

kfaadmin. (2007). Collecting, Archiving and Exhibiting Digital Design Data (Project Documentation). Art Institute of Chicago.

Kristiansen, P., & Rasmussen, R. (2014). Building a Better Business Using the Lego Serious Play Method. New York, UNITED STATES: John Wiley & Sons, Incorporated.

LEE, J. T.-H., & Chui-Shan Chow, C. (2016). Almost

Democratic: Christian Activism and the Umbrella Movement in Hong Kong, 45, 252–268.

Marcel Danesi. (2009). X-rated! The power of mythic symbolism in popular culture (1st ed.). New York: Palgrave Macmillan.

Mark Baskinger. (2010). Playing in the Sandbox. UX Magazine, 566.

Mathigon. (n.d.). World of Mathematics. Mathigon. Learning Platform, https://mathigon.org/world/Game_Theory.

Newswire. (2017). Bluetooth Beacons Market Analysis By Technology (iBeacon, Eddystone), By End-use (Retail, Travel & Tourism, Healthcare, Financial Institutions), By Region, And Segment Forecasts, 2014 - 2025. PR Newswire; New York.

Nicholas Christakis. (2017, November). IDSS Distinguished Seminar Speaker Nicholas Christakis - Fall 2017. Seminar, MIT Institute for Data, Systems,; Society.

Nicky Case. (2014a, June). Coming Out Simulator 2014. Writing In Case.

Nicky Case. (2014b, September). Explorable Explanations. Writing In Case. Blog.

Omni Systems Limited. (2009). Eufloria. Eufloria. Product Page, http://www.eufloria-game.com/news.php.

Orlikowski, W. J. (2007). Sociomaterial Practices: Exploring Technology at Work. Organization Studies, 28(9), 1435–1448.

Oswald, D., & Kolb, S. (2014). Flat Design vs. SkeuomorphismEffects on Learnability and Image Attributions in Digital Product Interfaces. In DS 78: Proceedings of the 16th International conference on Engineering and Product Design Education (E&PDE14), Design Education and Human Technology Relations, University of Twente, The Netherlands, 04-05.09. 2014.

Paech, V. (2018, May). Content for Community. Talk, General Assembly.

Putnam, R. D. (2000). Bowling alone: The collapse and revival of American community. New York: Simon & Schuster.

Stein, B. (1999). The digital dialectic : New essays on new media. Cambridge, Mass.: MIT Press.

Stickdorn, M., & Schneider, J. (2011). This is service design thinking: Basics, tools, cases. Wiley.

The virtual you and the real you. (2018, March). Big Ideas. ABC: Radio National.

Thilmany, J. (2014). The Maker Movement and the U.S. Economy. Mechanical Engineering; New York, 136(12), 28–29.

Wilkie, A. (2010, October). User Assemblages in Design: An Ethnographic Study (PhD thesis). Goldsmiths, University of London.

Wu, P., Morie, J., Wall, P., Ott, T., & Binsted, K. (2016). ANSIBLE: Virtual Reality for Behavioral Health. Procedia Engineering, 159, 108–111. https://doi.org/10.1016/j.proeng.2016.08.132

Young, C. M. Y., & Lo, B. C. Y. (2012). Cognitive appraisal mediating relationship between social anxiety and internet communication in adolescents. Personality and Individual Differences, 52(1), 78–83. https://doi.org/10.1016/j.paid.2011.09.001

♦ Appendices

Experiment One: Circles

http://rupertqmoore.xyz/ExperimentOneCircles/

Screens

```
// @ts-check
/* eslint no-undef: 0 */
/* eslint new-cap: 0 */
/* eslint no-unused-vars:
[“error”, { “varsIgnorePattern”:
“setup|draw|preload|recordFrame
|recordSetup|p5Canvas0|
mousePressed|previewP5” }] */
/** Colours
 * @enum {string}
 */
// Enum allowing easy access to Colours
const colourEnum = [
  getC(hues.reds, 5).hex,
  getC(hues.apricots, 5).hex,
  getC(hues.yellows, 5).hex,
  getC(hues.greens, 5).hex,
  getC(hues.blues, 5).hex,
  getC(hues.purples, 5).hex
];
// Declaring globals for the canvas and database
let oFirebaseDatabase;
/**
 * canvas
 * @type {HTMLCanvasElement}
 */
let eP5Canvas0;
/**
 * Checks whether sprite will go
over edge and bounces it
 *
 * @param {p5.Sprite} _sprite
 */
function edgeCollide(_sprite) {
  // Checks if sprite is near left edge
and if so inverts it’s velocity
  if (_sprite.position.x < 25) {
    _sprite.position.x = 26;
    _sprite.velocity.x = abs(_sprite.velocity.x);
  }

  if (_sprite.position.x > width - 25) {
    _sprite.position.x = width - 26;
    _sprite.velocity.x = -abs(_sprite.velocity.x);
  }

  if (_sprite.position.y < 25) {
    _sprite.position.y = 26;
    _sprite.velocity.y = abs(_sprite.velocity.y);
  }

  if (_sprite.position.y > height - 25) {
    _sprite.position.y = height - 26;
    _sprite.velocity.y = -abs(_sprite.velocity.y);
  }
}
// Angle for Identifier
const GOLDERN_RATIO = 6.2831 / 1.618;

/**
 * Actor Class
 *
 * Class for Participant generated Actors (UAs)
 * @class Actor
 *
 */
class UA {
  /**
   *
   * Constructs the UA
   * Creates an instance of Actor.
   * @param {p5} p p5 instance to use
```

```
   * @param {number} _x x pos
   * @param {number} _y y pos
   * @param {object} args fingerprint arguements
   * @memberof Actor
   *
   */

  constructor(p, _x, _y, args) {
    // Create the Sprite
    /** @type {p5.Sprite} */
    this.spr = createSprite(_x, _y, 30, 30);

    // apply the fingerprint to the new UA
    this.fingerprint = {
      aPersonalities: [args.
d1c, args.d2c, args.d3c],
      nIDHash: args.cH,
      oBehaviours: args.b
    };
    // Initialise the Actors Strengths
    this.strengths = [0.5, 0.5,
0.5, 0.5, 0.5, 0.5];

    // Initialise the PrevGen (UNUSED);
    this.prevGen = args.pg;

    this.lifeData = { 0: 0, 1: 0,
2: 0, 3: 0, 4: 0, 5: 0 };
    this.relationships = [];
    // set Default maxSpeed, restitution (
retained energy on collision ) and multipliers
    this.spr.maxSpeed = 0.6;
    this.spr.restitution = 0.8;
    this.growMult = 1;
    this.drainMult = 1;

    // set default rotation speed
and initial wander velocity
    this.spr.rotationSpeed = 0.3;
    this.spr.velocity.x = Math.
cos(Math.random() * TWO_PI) / 10;
    this.spr.velocity.y = Math.
sin(Math.random() * TWO_PI) / 10;

    // Set Collision Circle
    this.spr.setCollider('circle', 0, 0, 25);

    // Asign fill colours to sections
    let fill1 = color(colourEnum[this.
fingerprint.aPersonalities[0]]);
    let fill2 = color(colourEnum[this.
fingerprint.aPersonalities[1]]);
    let fill3 = color(colourEnum[this.
fingerprint.aPersonalities[2]]);

    // Calculate Behaviours

    // Speed / Intro-Extro behaviour
    if (this.fingerprint.oBehaviours[0] !== '0') {
      if (this.fingerprint.oBehaviours[0]
=== 'E') this.spr.maxSpeed = 1;
      else this.spr.maxSpeed = 0.3;
    }

    // Drain / Intuit-Sense behaviour
    if (this.fingerprint.oBehaviours[1] !== '0') {
      if (this.fingerprint.oBehaviours[1]
=== 'N') this.drainMult -= 0.5;
      else this.drainMult += 0.5;
    }

    // Spin / Think-Feel behaviour
    if (this.fingerprint.oBehaviours[2] !== '0') {
      if (this.fingerprint.oBehaviours[2] === 'T')
        this.spr.rotationSpeed -= 0.2;
      else this.spr.rotationSpeed += 0.2;
    }

    // Grow / Percep-Judge behaviour
    if (this.fingerprint.oBehaviours[3] !== '0') {
      if (this.fingerprint.oBehaviours[3]
=== 'J') this.growMult += 0.2;
      else this.growMult -= 0.2;
    }
    // Initialise sprite draw loop
    this.spr.draw = () => {
      // Draw two concenetric boundry Circles
      fill(getC(hues.neutrals, 4).hex);
      ellipse(0, 0, 50, 50);
      fill(getC(hues.neutrals, 6).hex);
      ellipse(0, 0, 48, 48);

      // Draw personality arcs circles
      let nAng0 = 0;
      let nAng1 = (TWO_PI / 3) * 1;
      let nAng2 = (TWO_PI / 3) * 2;
      fill(fill1);
      arc(0, 0, 46, 46, nAng0, nAng1);
      fill(fill2);
      arc(0, 0, 46, 46, nAng1, nAng2);
      fill(fill3);
      arc(0, 0, 46, 46, nAng2, TWO_PI);

      // Draw inner circle
      fill(getC(hues.neutrals, 6).hex);
      ellipse(0, 0, 36, 36);

      // calculate current life/memory data
      this.calculateLifeData();
```

```
      // Draw UA's life data

      // Set start angle taking
into account current rotation
      let nStartAng = 0 -
radians(this.spr.rotation);

      // Loop throught life data catagories
      for (let nDataKey in this.lifeData) {
        // Check that the data does exist
        if (this.lifeData[nDataKey]) {
          // Update start Angle
          let nEndAng = nStartAng + TWO_
PI * this.lifeData[nDataKey] - 0.01;
          fill(colourEnum[nDataKey]);
          arc(0, 0, 30, 30, nStartAng, nEndAng);
          nStartAng = nEndAng;
        }
      }

      // draw white center of sprite
      fill(color(255));
      ellipse(0, 0, 20, 20);

      // check for collision
      edgeCollide(this.spr);

      // draw IDHash
      stroke(120);
      strokeWeight(1);

      let sID = this.fingerprint.nIDHash;
      for (let i = 0; i < sID % 8; i++) {
        // calculate angles of lines
        let nStartAng = GOLDERN_RATIO * i;
        let nEndAng = nStartAng + PI / 2;
        strokeWeight(0.8);
        let l = 7;
        line(0, 0, sin(nEndAng)
* l, cos(nEndAng) * l);

        // Decide wether to draw square or circle
        rectMode(RADIUS);
        ellipseMode(RADIUS);
        if (Math.floor(sID * 2.33)
% 3 === 0) rect(0, 0, 4, 4);
        else if (Math.floor(sID * 2.33)
% 3 === 1) ellipse(0, 0, 4, 4);
      }
    };
  }
  /**
   * Formates Life data from
relationships, runs every frame
   */
  calculateLifeData() {
    // Loops backwards UAs current relationships
    for (let nRelInd = this.relationships.
length - 1; nRelInd >= 0; nRelInd--) {
      // If nolonger linked, remove
the relationship from the list
      if (!listOfLinks.includes(this.
relationships[nRelInd])) {
        this.relationships.splice(nRelInd, 1);
        continue;
      }
      let oLink = this.relationships[nRelInd];

      // check if UA has colour recorded,
if not, adds it to the record
      if (
        this.lifeData[oLink.colour] === undefined ||
        this.lifeData[oLink.colour] === null
      )
        this.lifeData[oLink.colour] = 0;

      // Increases the value of
the link colour for this UA
      this.lifeData[oLink.colour] += 0.001;
    }

    // loops over all current life entries
    for (let c in this.lifeData) {
      // Skips if 0
      if (this.lifeData[c] === 0) continue;

      /*  Every second frame drains %0.05 (
multiplied by the drain modifier) from the entry */
      if (frameCount % 2 === 0) this.
lifeData[c] -= 0.0005 * this.drainMult;

      // Resets entry to 0 if it goes below
      if (this.lifeData[c] <= 0) {
        this.lifeData[c] = 0;
      }
    }

    // gets the sum of all lifedata entries
    let nLifeDataSum = Object.values(this.
lifeData).reduce((a, b) => {
      return a + b;
    });

    // loops over all current life entries
    for (let sLDcat in this.lifeData) {
      // Normalise value
      if (!nLifeDataSum)
        this.lifeData[sLDcat] = this.
lifeData[sLDcat] / nLifeDataSum;
```

```
      let nLDcat = Number(sLDcat);
      let nStr = this.lifeData[sLDcat];

      // Set attraction level based on strength
      if (nStr < 0.3) this.strengths[nLDcat]
= 0.5 * this.growMult;
      else if (nStr > 0.8) this.
strengths[nLDcat] = -0.5 * this.growMult;
      else this.strengths[nLDcat] =
map(nStr, 0.3, 0.8, 0.5, -0.5);
    }
  }
}
// Class to manage links
class Link {
  /**
   * Creates an instance of link.
   * @param {UA} n1
   * @param {UA} n2
   * @param {number} c
   * @memberof link
   */
  constructor(n1, n2, c) {
    // set default values
    this.n1des = 1;
    this.n2des = 1;
    this.node1 = n1;
    this.node2 = n2;
    this.angle = 0;
    /** @type{number} */
    this.colour = c;
    this.baseStrength = 1;
    this.length = p5.Vector.dist(n1.
spr.position, n2.spr.position);

    // update loop
    this.draw = () => {
      // if strength is negative xero it
      this.baseStrength = 1 - 0.2
* (this.length / 150);
      if (this.length < 75 && (this.
n1des < 0 || this.n2des < 0))
        this.baseStrength = 0.01;
      if (this.n1des < 0) this.n1des *= 1;
      if (this.n2des < 0) this.n2des *= 1;

      // Normalise strengths
      this.n1str = this.n1des * this.
baseStrength * 0.001;
      this.n2str = this.n2des * this.
baseStrength * 0.001;

      // increase strength if negative
      if (this.n2str < 0) this.n2str *= 100;

      let actor1pos = this.node1.spr.position;
      let actor2pos = this.node2.spr.position;

      // set actors attraction
      this.node1.spr.attractionPoint(this.
n2str, actor2pos.x, actor2pos.y);
      this.node2.spr.attractionPoint(this.
n1str, actor1pos.x, actor1pos.y);

      // if actors are too close
get them to move away
      if (this.length < 75) {
        this.node1.spr.attractionPoint(-0.001,
actor2pos.x, actor2pos.y);
        this.node2.spr.attractionPoint(-0.001,
actor1pos.x, actor1pos.y);
      }
      // draw line
      noFill();
      stroke(colourEnum[this.colour]);
      strokeWeight(3);
      line(
        this.node1.spr.position.x,
        this.node1.spr.position.y,
        this.node2.spr.position.x,
        this.node2.spr.position.y
      );
    };
  }
}

// actor preview canvas
var preview = p => {
  p.setup = () => {
    // set up canvas
    p.createCanvas(50, 50);
    p.background(255);
    p.frameRate(3);
  };
  p.draw = () => {
    // draw base
    p.push();
    p.noStroke();
    p.translate(25, 25);
    p.fill(getC(hues.neutrals, 4).hex);
    p.ellipse(0, 0, 50, 50);
    p.fill(getC(hues.neutrals, 6).hex);
    p.ellipse(0, 0, 48, 48);

    // draw the arcs
    let ang = [0, (TWO_PI / 3) * 1,
(TWO_PI / 3) * 2, TWO_PI];
    for (let i = 1; i <= 3; i++) {
      // get checkbox values from dom
      let q = document.
```

```
querySelector(`input[name="options${i}"]:checked`);
      if (q === null) {
      } else {
        let j = q.id.match(/\d/);
        p.fill(colourEnum[j[0]]);
        p.arc(0, 0, 46, 46, ang[i - 1], ang[i]);
      }
    }
    p.fill(color(255));
    p.ellipse(0, 0, 36, 36);
    p.stroke(0);

    // Calculate and draw the hash
    let d = new Date();
    let cHash = Math.floor(d.getTime() / 1000) % 32;
    for (let i = 0; i < cHash % 8; i++) {
      let ang = GOLDERN_RATIO * i;
      let ang2 = ang + PI / 2;
      strokeWeight(0.8);
      let l = 9;
      p.line(0, 0, sin(ang2) * l, cos(ang2) * l);
      p.rectMode(RADIUS);
      p.ellipseMode(RADIUS);
      if (Math.floor(cHash * 2.33) %
3 === 0) p.rect(0, 0, 3, 3);
      else if (Math.floor(cHash * 2.33)
% 3 === 1) p.ellipse(0, 0, 3, 3);
    }
  };
};
// @ts-ignore
let previewP5 = new p5(preview, 'prev');
let downloadedData = {};

let listOfActors = [];

let listOfLinks = [];
let config;

function setup() {
  // firebase config
  config = {
    apiKey:
'AIzaSyCZh7bDhcHYesPc0FeKxriL7EZ2Kopk2us',
    authDomain: 'awesomesaucerupert.
firebaseapp.com',
    databaseURL: 'https://
awesomesaucerupert.firebaseio.com',
    projectId: 'awesomesaucerupert',
    storageBucket: 'awesomesaucerupert.
appspot.com',
    messagingSenderId: '465094389233'
  };
  firebase.initializeApp(config);
  oFirebaseDatabase = firebase.
database().ref('ExcerciseOne');

  // download data off firebase
  oFirebaseDatabase.once('value', data => {
    downloadedData = data.val();
    for (let a of Object.values(downloadedData)) {
      // add actors
      listOfActors.push(
        new UA(this, Math.random() *
width, Math.random() * height, a)
      );
    }
  });
  // create canvas and assign it to DOM
  eP5Canvas0 = createCanvas(windowWidth
- 100, 720);
  eP5Canvas0.parent('canvas');
  // setup setup button
  let x = select('#submitButton');

  // assign func to button
  x.mousePressed(() => submitNewActor());
}
function draw() {
  // draw background
  background(getC(hues.neutrals, 6).hex);
  // draw links
  for (let l of listOfLinks) {
    l.draw();
  }

  for (let [j, i] of listOfActors.entries()) {
    // draw all actors
    i.spr.display();
    // if experiment just restarted, draw a
red circle around the last created actor
    if (Number(j) === listOfActors.
length - 1 && frameCount < 600) {
      noFill();
      strokeWeight(3);
      stroke(255, 200, 200, 255 - frameCount);
      ellipse(i.spr.position.x,
i.spr.position.y, 60, 60);
    }
    i.spr.bounce(allSprites);
  }
  if (frameCount % 20 === 0) asLinks();
}
function submitNewActor() {
  // checks
  let actorArgs = {};
  let c = [];
  let isAnyNull = false;
  for (let i = 1; i <= 3; i++) {
    let but = document.
```

```
querySelector(`input[name=”options${i}”]:checked`);
    if (but === null) {
      isAnyNull = true;
      break;
    }
    c.push(Number(but.id.match(/\d/)));
  }
  if (isAnyNull) return 0;
  Object.assign(actorArgs, { d1c:
c[0], d2c: c[1], d3c: c[2] });
  console.log(actorArgs);
  let d = new Date();
  let seconds = Math.floor(d.getTime() / 1000);
  actorArgs.cH = seconds % 32;
  const bEnum = [‘E_I’, ‘N_S’, ‘T_F’, ‘J_P’];
  let behaviours = [];
  for (let key of bEnum) {
    let bObj = document.
querySelector(`input[name=”${key}”]:checked`);
    let b;
    if (bObj === null) b = ‘0’;
    else b = bObj.id.length === 3 ? ‘0’ : bObj.id;
    behaviours.push(b);
  }
  actorArgs.b = behaviours;
  oFirebaseDatabase.push(actorArgs);
  listOfActors.push(new UA(this, random(width),
random(height), actorArgs));
  location.reload();
}

// function mousePressed() {
//   if (mouseX > width || mouseY > height) return;
//   console.log(listOfActors);
//   listOfActors.push(
//     new Actor(this, mouseX, mouseY, {
//       pg: {},
//       d1c: Math.floor(random(6)),
//       d2c: Math.floor(random(6)),
//       d3c: Math.floor(random(6)),
//       cH: second() % 8,
//       b: [5]
//     })
//   );
// }
async function asLinks() {
  calculateLinks();
}
function calculateLinks() {
  let tempLinks = [];
  for (let [i, a] of listOfActors.entries()) {
    for (let [j, b] of listOfActors.entries()) {
      if (i >= j) continue;
      let d = p5.Vector.dist(a.spr.
position, b.spr.position);
      if (d < 150 && d > 50) {
        let l = new Link(a, b, 2, 0);
        tempLinks.push(l);
        a.relationships.push(l);
        b.relationships.push(l);
      }
    }
  }

  for (let link of tempLinks) {
    let n1 = link.node1;
    let n2 = link.node2;
    /** @type {p5.Vector} */
    let p1 = n1.spr.position;
    let ns = p5.Vector.sub(p1,
n2.spr.position).heading();
    link.angle = (ns + PI + radians(-(n1.
spr.rotation % 360) + 720)) % TWO_PI;
    let cat1 = Math.floor((link.
angle / TWO_PI) * 2.999);
    link.colour = n1.fingerprint.
aPersonalities[cat1];
    let cat2 = Math.floor(
      ((link.angle + PI + radians(n2.spr.
rotation - 90)) % TWO_PI) * 2.999
    );
    let n1c = n1.fingerprint.aPersonalities[cat1];
    let n2c = n2.fingerprint.aPersonalities[cat2];
    if (n1.strengths[n2c] >= n2.strengths[n1c]) {
      link.colour = n2c;
    } else {
      link.colour = n1c;
    }
    link.n1des = n1.strengths[link.colour];
    link.n2des = n2.strengths[link.colour];
    if (
      link.length > 70 &&
      n1.fingerprint.aPersonalities[cat1] !==
        n2.fingerprint.aPersonalities[cat2]
    ) {
    }
  }
  listOfLinks = tempLinks;
}
```

Experiment Two: Nodal

http://rupertqmoore.xyz/ExperimentTwoNodal/

```
// @ts-check

// HTMLCanvasElements

/** @type {HTMLCanvasElement} */
let cExpEnv;
// region usefullGlobals
/* eslint-disable no-unused-vars */
// enviroment
const OBSERVE = 1;
const SELECT = 2;
const TARGET = 3;
// msg
const M_ENUM = [“”, “select node”,
“select”, “select recepient”];
const NODE_SELECT = 1;
const ACTOR_SELECT = 2;
const CITY_NAMES = [
  “Harhaven”,
  “Ashhampton”,
  “Hartburgh”,
  “Henwike”,
  “Eastmouth Head”,
  “Langley”,
  “Nether Westdon”,
  “Oxhampton”,
  “Foxthorpe”,
  “Dunthorpe Cross”,
  “Northwood”,
  “Barcaster”,
  “Millburn Green”,
  “Market Henbury”,
  “Dunwell Head”,
  “Kirwich”,
  “Guildport”,
  “Westley”,
  “Heath”,
  “Kinington”,
  “Upper Moorburgh”,
  “Kinburgh”,
  “Marshampton”,
  “Hartington”,
  “Woolmouth”,
  “Redhurst”,
  “Westpool Heath”,
  “Oulhaven”,
  “Blackchester”,
  “Oakbrook”,
  “Guildcaster Green”,
  “Overwell”,
  “Ashmoor”,
  “Westchester”,
  “Hazelfield”,
  “Market”,
  “Hindford”,
  “Hartwich”,
  “Overbury”,
  “Ilhall”,
  “Great Woolwich”,
  “Preswich”,
  “Oxley”,
  “Upper Ashburgh”,
  “Normoor”,
  “Thornthorpe”,
  “Loxley”,
  “Foxworth”,
  “Great”,
  “Hazelby”,
  “Little Langwell”,
  “Blackfield Lake”,
  “Little Kinham”,
  “Upper Oulwick”,
  “Northfold”,
  “Newbrook”,
  “Cross”,
  “Westwick”,
  “Thornfold Head”,
  “Blackleigh”,
  “Oakfield Green”,
  “Nether Hartby”,
  “Oxleigh”
]; // eslint-disable-next-line no-unused-vars
// eslint-disable-next-line no-unused-vars
/* eslint-enable no-unused-vars */
// @ts-ignore
/** @type {cEnum} */ let colourEnum = [
  getC(hues.reds, shades.light).hex,
  getC(hues.oranges, shades.light).hex,
  getC(hues.yellows, shades.light).hex,
  getC(hues.greens, shades.light).hex,
```

```
  getC(hues.blues, shades.light).hex,
  getC(hues.purples, shades.light).hex
];
colourEnum.r = colourEnum[0];
colourEnum.o = colourEnum[1];
colourEnum.y = colourEnum[2];
colourEnum.g = colourEnum[3];
colourEnum.b = colourEnum[4];
colourEnum.p = colourEnum[5];

// endregion

/** Class representing an Actor */
class Actor {
  /**
   * Creates an instance of Actor.
   * @constructor
   * @param {p5} p5
   * @param {number} colour
   * @param {string} parent
   * @param {?string} id
   * @param {string} name
   * @memberof Actor
   */ /**
   */
  /**
   * Creates an instance of Actor.
   * @constructor
   * @param {p5} p5
   *
   * @param {any} p5
   * @param {any} colour
   * @param {any} parent
   * @param {any} name
   * @memberof Actor
   */

  constructor(p5, colour, pId, name) {
    if (arguments.length === 2) {
      this.id = arguments[1];

      let p = Object.values(importedData.
actorList).filter(
        a => this.id === a.id
      )[0];
      Object.assign(this, p);
    } else {
      /** @type {string} */
      this.id = hash(
        Number(new Date().getTime() + Math.
floor(random() * 200)).toString()
      );
      while (usedHashes.includes(this.id)) {
        this.id = hash(
          Number(new Date().getTime() +
Math.floor(random() * 200)).toString()
        );
      }
      usedHashes.push(this.id);
      d.ref("exerciseTwo").update({
        "/usedHashes": usedHashes
      });
      this.pId = pId;

      this.type = colour;
      this.name = name;
    }
    this.timeSincePost = 5 * 60;
    this.parent = nodeList.filter(value => {
      return value.id === this.pId;
    })[0];
    this.parent.members.push(this);
    /** @type {Sprite} */
    this.s = createSprite(0, 0, 40, 40);
    this.s.mouseActive = true;
    // this.s.debug = true;
    this.s.setCollider("circle", 0, 0, 20);
    this.s.draw = () => {
      this.timeSincePost++;
      if (selected === this) scale(1.6);
      noStroke();
      fill(colourEnum[this.type]);
      ellipse(0, 0, 40, 40);
      fill(colourEnum[(6 + this.type + 1) % 6]);
      arc(0, 0, 40, 40, PI / 2,
PI + PI / 3, CHORD);
      fill(colourEnum[(6 + this.type + -1) % 6]);
      arc(0, 0, 40, 40, TWO_PI -
PI / 3, PI / 2, CHORD);
      rotate(radians(-this.s.rotation));
      fill(255);
      stroke(55);
      strokeWeight(0.5);
      textSize(12);
      textAlign(CENTER, CENTER);
      text(this.name.substring(0, 2), 0, 0);
      if (!this.parent.members.includes(this))
this.parent.members.push(this);
    };

    this.s.onMousePressed = () => {
      switch (mode) {
        case SELECT:
          updateSelected(this);
          break;
        case NODE_SELECT:
          updateSelected();
          break;
        case TARGET:
          confirmSend(this);
```

```
          break;
      }
    };
  }
}

/** A Class representing serverNodes */
class ServerNode {
  /**
   * Creates an instance of ServerNode.
   * @constructor
   * @param {p5} p5
   * @param {number} mainType
   * @param {number[]} blockedTypes
   * @memberof ServerNode
   */

  constructor(p5, mainType, blockedTypes) {
    this.evaluateOn20 = false;
    if (arguments.length === 2) {
      /** @type {String} */
      this.id = arguments[1];
      let p = Object.values(importedData.
nodeList).filter(d => {
        return this.id === d.id;
      })[0];
      Object.assign(this, p);
      this.mainType = p.mainType;
      this.name = p.name;
      if ("blockedTypes" in p) {
        let bt = [];
        for (let y of Object.values(p.
blockedTypes)) bt.push(y);
        this.blockedTypes = bt;
      } else this.blockedTypes = [];
    } else {
      this.mainType = mainType;

      /** @type {number[]} */
      this.blockedTypes = blockedTypes;
      this.id = hash(
        Number(new Date().getTime() + Math.
floor(random() * 200)).toString()
      );
      while (usedHashes.includes(this.id)) {
        this.id = hash(
          Number(new Date().getTime() +
Math.floor(random() * 200)).toString()
        );
      }
      usedHashes.push(this.id);
      d.ref("exerciseTwo").update({
        "/usedHashes": usedHashes
      });
      let pickNew = false;
      while (!pickNew) {
        this.name = random(CITY_NAMES);
        if (!usedNames.includes(this.
name)) pickNew = true;
      }
      usedNames.push(this.name);
      d.ref("exerciseTwo").update({
        usedNames: usedNames
      });
    }
    this.s = createSprite(0, 0, 100, 100);
    /** @type {Actor[]} */
    this.members = actorList.
filter(a => this.id === a.pId);
    /** @type {Message[]} */
    this.messages = messageList.
filter(m => m.targetId === this.id);
    this.rotation = 0;
    this.s.mouseActive = true;
    this.s.setCollider("circle", 0, 0, 50);
    this.s.draw = () => {
      if (selected === this) scale(1.4);
      let sAng = 0;
      let eAng = TWO_PI / 6;
      for (let i = 0; i < 6; i++) {
        for (let i = 0; i < 6; i++) {
          noStroke();
          if (this.blockedTypes.
includes(i)) fill(100);
          else fill(colourEnum[i]);
          arc(0, 0, 25, 25, sAng, eAng);
          sAng = eAng;
          eAng += TWO_PI / 6;
        }
        fill(colourEnum[this.mainType]);
        ellipse(0, 0, 10, 10);
        stroke(100);
        strokeWeight(2);
        noFill();
        ellipse(0, 0, 100, 100);
      }

      this.s.onMousePressed = () => {
        switch (mode) {
          case SELECT:
            updateSelected(this);
            break;
          case NODE_SELECT:
            confirmMove(this);
            break;
          case TARGET:
            updateSelected();
            break;
        }
      };
```

```
      // this.s.debug = true;
    };
  }
  update() {
    if (this.messages.length % 20 >
10) this.evaluateOn20 = true;
    if (frameCount % 30 === 0) {
      this.messages = [];
      for (let m of messageList) {
        if (m.target === this)
this.messages.push(m);
      }
      if (
        this.messages.length % 20 === 0 &&
        this.messages.length > 3 &&
        this.evaluateOn20
      ) {
        let cCounts = [0, 0, 0, 0, 0, 0];
        for (let m of this.messages)
cCounts[m.targetType] += 1;
        // Get Most Populous
        let highestP = max(cCounts);
        let mostPop = cCounts.
findIndex(d => d === highestP);
        this.mainType = mostPop;

        if (this.blockedTypes.
includes((mostPop + 6 + 1) % 6)) {
          if (random() > 0.4)
            this.blockedTypes.splice(
              this.blockedTypes.
indexOf((mostPop + 6 + 1) % 6),
              1
            );
        } else if (this.blockedTypes.
includes((mostPop + 6 - 1) % 6)) {
          if (random() > 0.4)
            this.blockedTypes.splice(
              this.blockedTypes.
indexOf((mostPop + 6 - 1) % 6),
              1
            );
        }
        let lowestP = min(cCounts);
        let leastPop = cCounts.
findIndex(d => d === lowestP);
        if (!this.blockedTypes.
includes((leastPop + 6 + 1) % 6)) {
          if (random() > 0.4) this.
blockedTypes.push((leastPop + 6 + 1) % 6);
        } else if (!this.blockedTypes.
includes((leastPop + 6 - 1) % 6)) {
          if (random() > 0.4) this.
blockedTypes.push((leastPop + 6 - 1) % 6);
        }
        d.ref("exerciseTwo/
nodeList/" + this.id).update({
          mainType: this.mainType,
          blockedTypes: this.blockedTypes
        });
        this.evaluateOn20 = false;
      }
    }

    drawSprite(this.s);
    let deltaAng = TWO_PI / max(this.
members.length, 1);
    this.rotation += TWO_PI / 600;
    let ang = 0;

    for (let x of this.members) {
      let pos = createVector(0, -100);
      pos.rotate(ang);
      pos.rotate(this.rotation);
      ang += deltaAng;
      pos.add(this.s.position);
      x.s.position = pos;
      let headDir = p5.Vector.
sub(x.s.position, this.s.position);
      x.s.rotation = headDir.
heading() / TWO_PI * 360 - 90;
    }
  }
}

class Message {
  constructor(p5, sourceId, targetId, directed) {
    if (arguments.length === 2) {
      this.id = arguments[1];

      let m = Object.values(importedData.
messageList).filter(
        m => m.id === this.id
      )[0];

      Object.assign(this, m);
      this.added = false;
    } else {
      this.id = hash(
        Number(new Date().getTime() + Math.
floor(random() * 200)).toString()
      );
      while (usedHashes.includes(this.id)) {
        this.id = hash(
          Number(new Date().getTime() +
Math.floor(random() * 200)).toString()
        );
      }
      usedHashes.push(this.id);
      d.ref("exerciseTwo").update({
```

```
        "/usedHashes": usedHashes
      });
      this.destroyed = false;
      this.direct = directed;
      this.sId = sourceId;
      this.sourceType = 0;
      this.tId = targetId;
      this.targetType = 1;
      this.added = true;
    }
    /** @type {Sprite} */

    this.s = createSprite(0, 0, 15, 15);
    this.s.limitSpeed(0.4);
    this.s.draw = () => {
      noStroke();
      fill(colourEnum[this.sourceType]);
      ellipse(0, 0, 10);
      fill(colourEnum[this.targetType]);
      ellipse(0, 0, 8, 8);
    };
  }
  init() {
    this.source = actorList.filter(a
=> this.sId === a.id)[0];
    if (!this.direct) this.target = actorList.
filter(a => this.tId === a.id)[0];
    else this.target = nodeList.
filter(a => this.tId === a.id)[0];
    this.s.limitSpeed(0.4);
    this.source = actorList.filter(a
=> this.sId === a.id)[0];
    if (this.target === undefined) {
      if (this.direct)
        this.target = actorList.
filter(a => this.tId === a.id)[0];
      else this.target = nodeList.
filter(a => this.tId === a.id)[0];
    }

    if (!this.added) {
      this.s.position = createVector(0, -40);
      this.s.position.rotate(random(TWO_PI));
      /** @type {p5.Vector} */
      let velo = this.s.position;
      this.s.position = p5.Vector.
add(this.s.position, this.target.s.position);
      velo.rotate(PI / 2);
      if (random() > 0.5) velo.rotate(PI);
      if (!this.added) velo.
limit(2 + (random() - 0.5));
      else velo.limit(4 + (random() - 0.5));
      this.s.setVelocity(velo.x, velo.y);
    }
    if (this.added) {
      this.s.position = this.source.s.position;
      this.s.limitSpeed(0.0004);
    }
    if (this.direct) {
      this.s.friction = 0.06;
      this.s.setVelocity(0, 0);
    }
  }
  update() {
    if (this.target === undefined) return;
    if (this.added)
      this.s.collide(this.target.s, () => {
        this.added = false;
        if ("blockedTypes" in this.target) {
          if (this.target.blockedTypes.
includes(this.targetType)) {
            this.destroyed = true;
            d
              .ref("exerciseTwo/
messageList/" + this.id)
              .update({ destroyed: true });
            let i = messageList.indexOf(this);
            messageList.splice(i, 1);
            return;
          }
        }
        if (!this.direct) this.init();
        else {
          if (this.recip === undefined) {
            let i = messageList.indexOf(this);
            messageList.splice(i, 1);
            return 0;
          } else {
            this.added = true;
            this.target = this.recip;
            this.recip = undefined;
          }
        }
      });

    if (p5.Vector.dist(this.target.s.position,
this.s.position) > 15) {
      let t = this.target.s.position;
      this.s.attractionPoint(0.4, t.x, t.y);
    }
    if (p5.Vector.dist(this.target.s.position,
this.s.position) > 45) {
      let t = this.target.s.position;
      this.s.attractionPoint(0.3, t.x, t.y);
    }

    drawSprite(this.s);
  }
}
```

```
let usedHashes = [“000001”, “000002”,
“000003”, “000004”, “000005”];
// #region global Variables
let actorList = [];

let usedNames = [
  “Tom from MySpace”,
  “Metropolis”,
  “West Camden”,
  “East Camden”
];
let messageList = [];
/** @type {ServerNode[]} */
let nodeList = [];
let selectedColor = 0;
// #endregion
/** @type {p5.Element} */
let choiceRadio;

let mode;
/** @type {Actor|ServerNode|null} */
let selected = null;
let selectedServerNodeElt;
// #region fireBaseSetupStuff
var config = {
  apiKey:
“AIzaSyCZh7bDhcHYesPc0FeKxriL7EZ2Kopk2us”,
  authDomain: “awesomesaucerupert.firebaseapp.com”,
  databaseURL: “https://
awesomesaucerupert.firebaseio.com”,
  projectId: “awesomesaucerupert”,
  storageBucket: “awesomesaucerupert.appspot.com”,
  messagingSenderId: “465094389233”
};

firebase.initializeApp(config);
let d = firebase.database();
let database = firebase.database().
ref(“exerciseTwo”);
let importedData;
// @ts-ignore
p5.prototype.getFirebaseData = function() {
  let ret = {};
  let self = this;

  database.once(“value”, snapshot => {
    let d = snapshot.val();
    for (let prop in d) {
      ret[prop] = d[prop];
    }
    self._decrementPreload();
  });
  return ret;
};
// @ts-ignore
p5.prototype.
registerPreloadMethod(“getFirebaseData”,
p5.prototype);
function preload() {
  // @ts-ignore
  importedData = getFirebaseData();
}
// #endregion
function postMsg(type) {
  /** @type {Actor} */
  let s;
  if (“type” in selected) s = selected;
  if (s.timeSincePost < 5 * 60) {
    select(“#actionError”).html(“Please
Wait 5 Seconds between Posts”);
    return;
  } else select(“#actionError”).html(“”);
  let m = new Message(this, s.id,
s.parent.id, false);
  m.targetType = type;
  m.sourceType = type;
  m.added = true;
  m.init();
  messageList.push(m);
  d.ref(“exerciseTwo/messageList/” + m.id).set({
    id: m.id,
    sourceType: m.sourceType,
    targetType: m.targetType,
    sId: m.source.id,
    tId: m.target.id,
    direct: false
  });
  s.timeSincePost = 0;
}
// endregion
function setup() {
  mode = ACTOR_SELECT;
  console.log(importedData);
  let wh = document.
getElementById(“canvasContainer”);
  cExpEnv = createCanvas(wh.
offsetWidth, wh.offsetHeight);
  cExpEnv.parent(“canvasContainer”);
  background(getC(2, 5).hex);
  textFont(“Ubuntu Mono”);

  setupAddBox();
  let noOfNodes = Object.
keys(importedData.nodeList).length;
  usedHashes = importedData.usedHashes;
  usedNames = importedData.usedNames;

  let sites = getSomePointsYall(noOfNodes, 5);
  let runningCount = 0;
  for (let [i, n] of Object.
```

```
entries(importedData.nodeList)) {
    // @ts-ignore
    let sN = new ServerNode(this, n.id);
    sN.s.position = createVector(...
sites[runningCount]);
    nodeList.push(sN);
    runningCount++;
  }
  for (let a of Object.
values(importedData.actorList)) {
    let nA = new Actor(this, a.id);
    actorList.push(nA);
  }
  for (let m of Object.
values(importedData.messageList)) {
    if (m.destroyed === true) continue;
    if (m.direct === true) continue;
    let nM = new Message(this, m.id);
    nM.init();
    nM.sourceType = m.sourceType;
    nM.targetType = m.targetType;
    messageList.push(nM);
  }
}

function draw() {
  if (frameCount % 60 === 0) serverNodeAdder();
  background(getC(2, 6).hex);
  if (mode !== SELECT) {
    fill(0, 60);
    rect(0, 0, width, height);
  }
  textSize(18);
  text("Mode: " + M_ENUM[mode], 20, 20);
  for (let n of nodeList) {
    n.members = actorList.
filter(a => n.id === a.pId);
    n.update();
  }
  for (let a of actorList) {
    a.s.display();
  }
  for (let m of messageList) {
    m.update();
  }
}
let tripUp = false;
function mousePressed() {
  if (mouseY > height) return 0;
  if (mode !== SELECT && !tripUp) tripUp = true;
  else updateSelected();
}
function setupAddBox() {
  // @ts-ignore
  // @ts-nocheck
  createP("")
    .id("createError")
    .class("body")
    .parent("#addBox")
    .style("color", "red");
  choiceRadio = createRadio();

  choiceRadio.option("red", 0);
  choiceRadio.option("orange", 1);
  choiceRadio.option("yellow", 2);
  choiceRadio.option("green", 3);
  choiceRadio.option("blue", 4);
  choiceRadio.option("purple", 5);
  choiceRadio.parent("#colourChooser");
  choiceRadio.style("padding", "5px");
  let n = createSpan("Name: ");
  n.parent("#addBox");
  let name = createInput("", "text");
  name.attribute("placeholder", "Jessie");
  name.parent("#addBox");
  createP("Selected Server Node: ")
    .class("h5")
    .id("serverNodeAdd")
    .parent("#addBox");
  // @ts-check
  selectedServerNodeElt = createSpan("Name").
parent("#serverNodeAdd");
  let createBut = createButton("Create")
    .parent("#addBox")
    .class("btn btn-dark");
  createBut.mousePressed(() => {
    let x = choiceRadio.value();
    let y = String(name.value());
    if (selected === null) {
      select("#createError").
html("You need to select a Node");
      return;
    }
    if (!("mainType" in selected)) {
      select("#createError").
html("You need to select a Node");
      return;
    }
    if (x === "") {
      select("#createError").html("You
need to select a colour");
      return;
    }
    if (y === "") {
      select("#createError").html("Input a name");
      return;
    }
    if (usedNames.includes(y)) {
      select("#createError").
html("That name is taken");
```

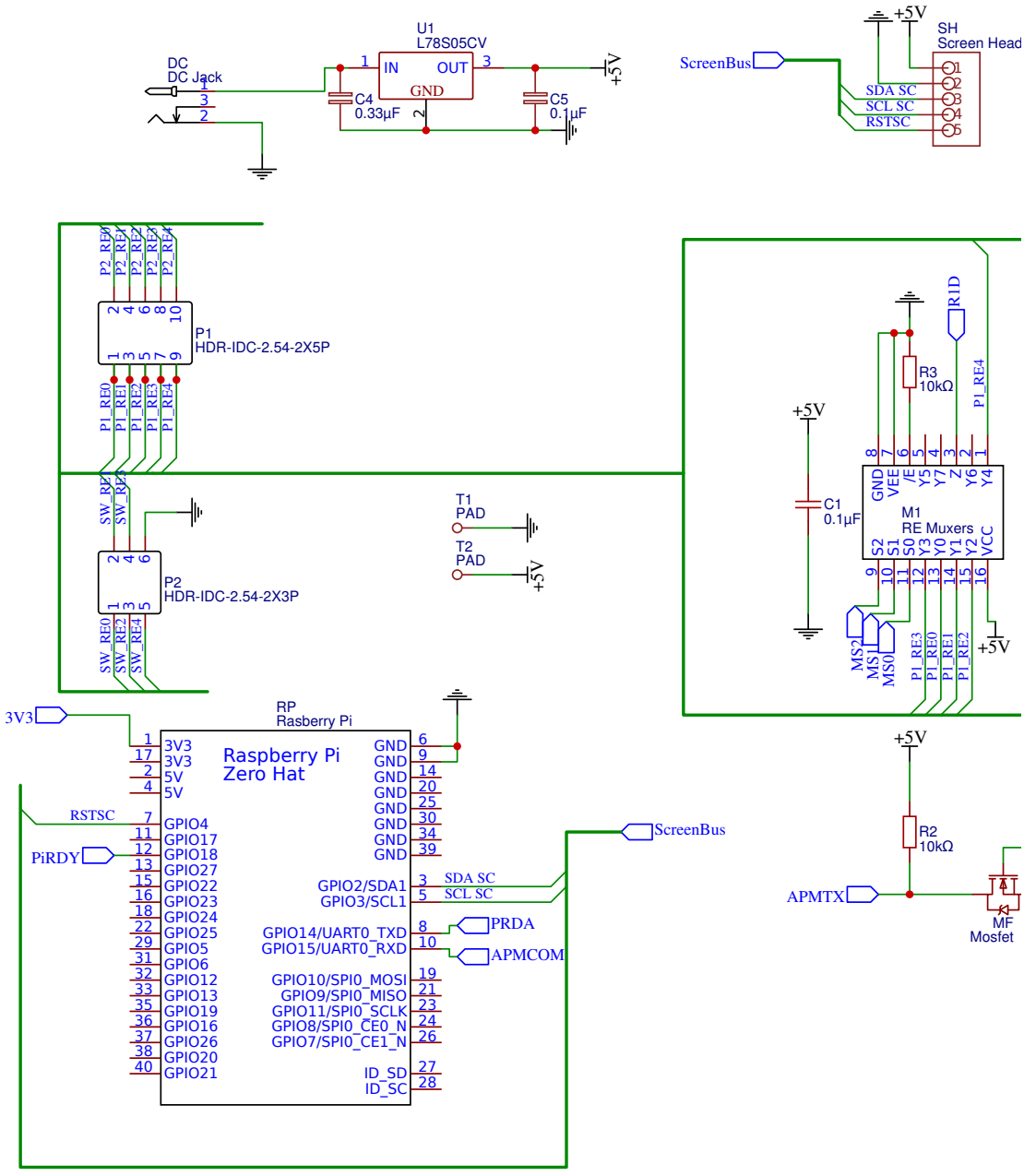
```
      return;
    }
    let colo = colourEnum.
indexOf(colourEnum[x[0]]);
    let a = new Actor(this, colo, selected.id, y);
    usedNames.push(y);
    d.ref("exerciseTwo").update({
      usedNames: usedNames
    });
    d.ref("exerciseTwo/actorList/" + a.id).set({
      id: a.id,
      name: a.name,
      pId: selected.id,
      type: colo
    });
    actorList.push(a);
  });
}
function getSomePointsYall(noPoints, noRelax) {
  /**
   * @type {[number,number][]}
   */
  // @ts-ignore
  let points = d3.range(noPoints).map(() => {
    /** @type {[number,number]} */
    return [random() * width - 100,
random() * height - 100];
  });
  let vor = d3.voronoi().size([width
- 100, height - 100]);
  for (let r = 0; r < noRelax; r++) {
    let polys = vor(points).polygons();
    for (let [i, p] of polys.entries()) {
      points[i] = d3.polygonCentroid(p);
    }
  }
  return points.map(i => [i[0] + 50, i[1] + 50]);
}
function updateBoxes() {
  let b = select("#infoC");
  b.elt.innerHTML = "";
  b.remove();
  b = createDiv()
    .id("infoC")
    .parent("infoBox");
  b.elt.innerHTML = "";
  let c = select("#aBox");
  c.remove();
  c = createDiv("")
    .class("container")
    .id("aBox")
    .parent("#actionBox");
  if (selected === null) return null;
  createP("id: " + selected.id)
    .class("h5")
    .parent("#infoC");
  if ("type" in selected) {
    // Actor
    createP("Name: " + selected.name)
      .class("h5")
      .parent("#infoC");
    createP("Main Colour: ")
      .class("h5")
      .parent("#infoC")
      .id("PrimaryColour");
    createDiv(" ")
      .style("border-radius", "50%")

      .style("background-color",
colourEnum[selected.type])
      .style("width", "15px")
      .style("height", "15px")
      .style("display", "inline-block")
      .parent("#PrimaryColour");
    createP("Seconday Colours: ")
      .class("h5")
      .parent("#infoC")
      .id("SecColour");
    createDiv(" ")
      .style("border-radius", "50%")
      .style("background-color",
colourEnum[(selected.type + 6 - 1) % 6])
      .style("width", "15px")
      .style("height", "15px")
      .style("display", "inline-block")
      .parent("#SecColour");
    createDiv(" ")
      .style("border-radius", "50%")
      .style("background-color",
colourEnum[(selected.type + 6 + 1) % 6])
      .style("width", "15px")
      .style("height", "15px")
      .style("display", "inline-block")
      .style("margin-left", "3px")
      .parent("#SecColour");
    createP("Node: " + selected.parent.name)
      .parent("#infoC")
      .class("h6");
    prepActions();
  }
  if ("mainType" in selected) {
    // Node
    createP("Name: " + selected.name)
      .class("h5")
      .parent("#infoC");
    createP("Main Colour: ")
      .class("h5")
      .parent("#infoC")
      .id("MainColour");
    createDiv(" ")
```

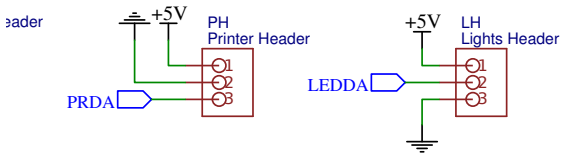
```
      .style("border-radius", "50%")
      .style("background-color",
colourEnum[selected.mainType])
      .style("width", "15px")
      .style("height", "15px")
      .style("display", "inline-block")
      .parent("#MainColour");
    createP("Blocked Colours: ")
      .class("h5")
      .parent("#infoC")
      .id("BColour");
    for (let b of selected.blockedTypes)
      createDiv(" ")
        .style("border-radius", "50%")
        .style("background-color", colourEnum[b])
        .style("width", "15px")
        .style("height", "15px")
        .style("display", "inline-block")
        .style("margin", "3px")
        .parent("#BColour");
    createP("Contains: ")
      .class("body")
      .parent("#infoC");
    let row = createDiv("")
      .class("row align-items-start")
      .parent("#infoC");
    let cCounts = [0, 0, 0, 0, 0, 0];
    for (let m of selected.messages)
cCounts[m.targetType] += 1;
    for (let [i, x] of cCounts.entries()) {
      let div = createDiv("")
        .class("")
        .parent(row)
        .style("display", "inline-block");
      createDiv(" ")
        .style("border-radius", "5%")
        .style("background-color", colourEnum[i])
        .style("width", "8px")
        .style("height", "12px")
        .style("display", "inline-block")
        .style("margin-left", "6px")
        .parent(div);
      createSpan(":" + x + "")
        .class("body")
        .style("display", "inline")
        .parent(div);
    }
  }
}
function updateSelected(obj) {
  if (selected !== null) {
    if (mode !== ACTOR_SELECT) {
      mode = ACTOR_SELECT;
    }
    selected = null;
  }
  if (obj !== undefined) {
    obj.selected = true;
    selected = obj;
    if (obj.mainType !== undefined)
      selectedServerNodeElt.elt.
innerHTML = obj.name;
  } else {
    selectedServerNodeElt.elt.innerHTML = "None";
  }

  updateBoxes();
}
function prepActions() {
  if (!("type" in selected)) throw selected;
  let actions = [
    (selected.type + 6 - 1) % 6,
    selected.type,
    (selected.type + 6 + 1) % 6
  ];
  createP("Post to Node:")
    .class("h5")
    .parent("#aBox");
  let post = createDiv("")
    .class("btn-group")
    .parent("#aBox");
  for (let a of actions) {
    /** @type {p5.Element} */
    let b = createButton("+")
      .class("btn")
      .style("background-color", colourEnum[a])
      .parent(post);
    b.mousePressed(() => postMsg(a));
  }
  createP("Post to an Actor:")
    .class("h5")
    .parent("#aBox");
  let send = createDiv("")
    .class("btn-group")
    .parent("#aBox");
  for (let a of actions) {
    /** @type {p5.Element} */
    let b = createButton("+")
      .class("btn")
      .style("background-color", colourEnum[a])
      .parent(send);
    b.mousePressed(() => sendMessage(a));
  }
  createP("").parent("#aBox");
  let move = createButton("Move")
    .class("btn btn-dark")
    .parent("#aBox");
  move.mousePressed(() => {
    tripUp = false;
    mode = NODE_SELECT;
```

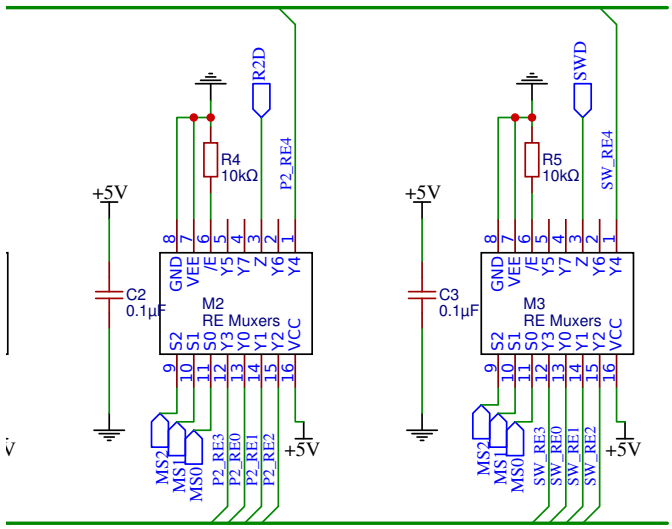
```
  });
}
function sendMessage(sourceCol) {
  if (!("type" in selected)) throw sendMessage;
  if (selected.timeSincePost < 5 * 60) {
    select("#actionError").html("Please
Wait 5 Seconds between Posts");
    return;
  }
  tripUp = false;

  selected.timeSincePost = 0;
  mode = TARGET;
  selectedColor = sourceCol;
}
function confirmSend(target) {
  if (!("type" in selected)) throw confirmSend;
  if (target === selected) {
    select("#actionError").html("it
can't send a message to itself");
    updateSelected();
    return;
  }
  select("#actionError").html("");
  let m = new Message(this, selected.
id, target.parent.id, true);
  m.sourceType = selected.type;
  m.targetType = selectedColor;
  m.target = target.parent;
  m.recip = target;
  m.added = true;
  m.direct = true;
  m.init();
  messageList.push(m);
  d.ref("exerciseTwo/messageList/" + m.id).set({
    id: m.id,
    sourceType: m.sourceType,
    targetType: m.targetType,
    sId: m.source.id,
    tId: m.recip.id,
    direct: true
  });
  updateSelected();
}
function confirmMove(target) {
  if (!("type" in selected)) throw confirmSend;
  if (target === selected.parent) {
    select("#actionError").html("You
can't move a Node to itself");
    updateSelected();
    return;
  }
  if (target.members.length > 2) {
    select("#actionError").
html("That Node is too full");
    updateSelected();
    return;
  }
  select("#actionError").html("");
  selected.parent = target;
  selected.pId = target.id;
  d.ref("exerciseTwo/nodeList/" + selected.
id).update({ pId: selected.pId });
  updateSelected();
}
function serverNodeAdder() {
  let isReadyForNode = true;
  for (let node of nodeList) {
    if (node.members.length < 3)
isReadyForNode = false;
  }
  if (!isReadyForNode) return null;
  let colourCount = [].fill(0, 0, 5);
  actorList.map(a => {
    colourCount[a.type] += 1;
  });
  let mostPopularCol = colourCount.
indexOf(max(colourCount));
  mostPopularCol = abs(mostPopularCol);
  mostPopularCol = mostPopularCol
> 6 ? 6 : mostPopularCol;
  let leastPopularCol = colourCount.
indexOf(min(colourCount));
  let newBlockedTypes = [];
  for (let i of colourCount.keys()) {
    if (i !== leastPopularCol && abs(i
+ 6 - (leastPopularCol + 6)) > 1)
      newBlockedTypes.push(i);
  }
  let node = new ServerNode(this,
mostPopularCol, newBlockedTypes);
  d
    .ref("exerciseTwo/nodeList/" + node.id)
    .set({
      id: node.id,
      name: node.name,
      mainType: node.mainType,
      blockedTypes: node.blockedTypes
    })
    .then(() => location.reload());
  nodeList.push(node);
  let points = getSomePointsYall(nodeList.
length, 5);
  for (let [i, x] of nodeList.entries()) {
    x.s.position = createVector(points[i]
[0], points[i][1]);
  }
}
// #region Creation
/* d.ref("exerciseTwo").update({
```

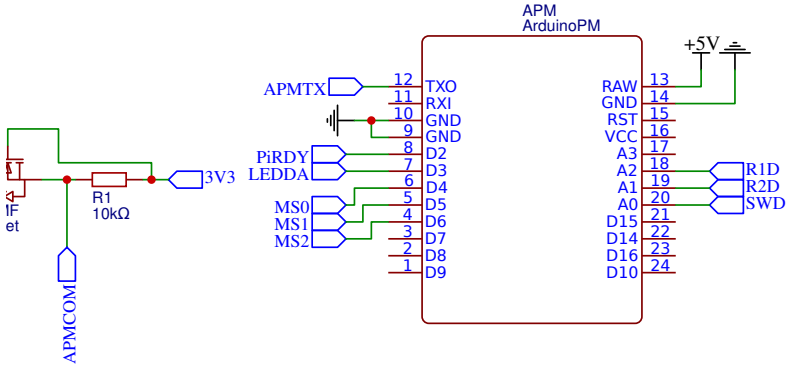
```
  usedHashes: [“000001”, “000002”, “000003”],
  usedNames: usedNames,
  nodeList: {
    “000001”: {
      id: “000001”,
      name: “Metropolis”,
      mainType: 0,
      blockedTypes: []
    },
    “000002”: {
      id: “000002”,
      name: “East Camden”,
      mainType: 1,
      blockedTypes: [3, 4, 5]
    },
    “000003”: {
      id: “000003”,
      name: “West Camden”,
      mainType: 4,
      blockedTypes: [0, 1, 2]
    }
  },
  actorList: {
    “000004”: {
      id: “000004”,
      name: “Tom from Myspace”,
      pId: “000001”,
      type: 4
    }
  }
}); */
// #endregion
```

Schematic: Vessel



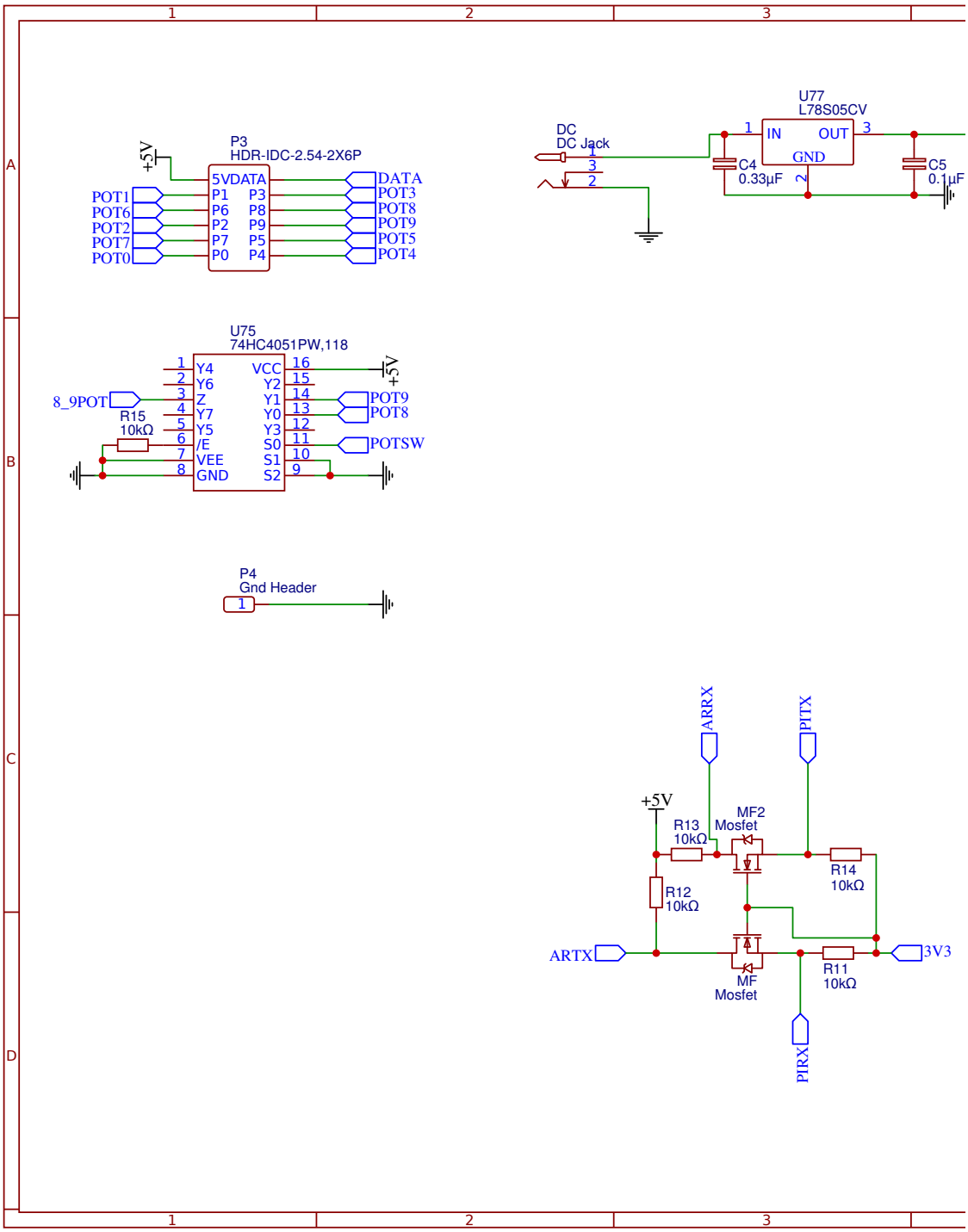


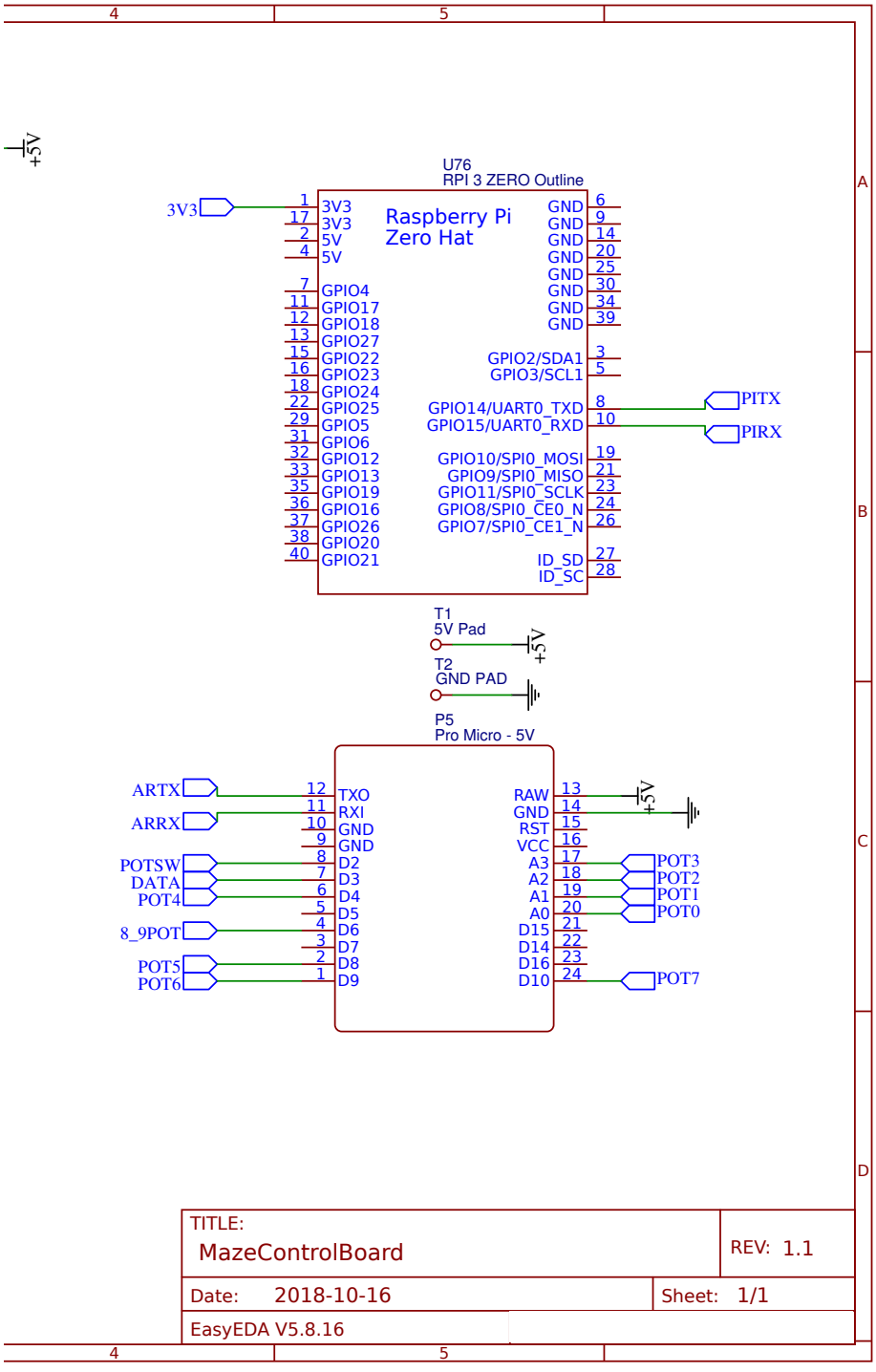




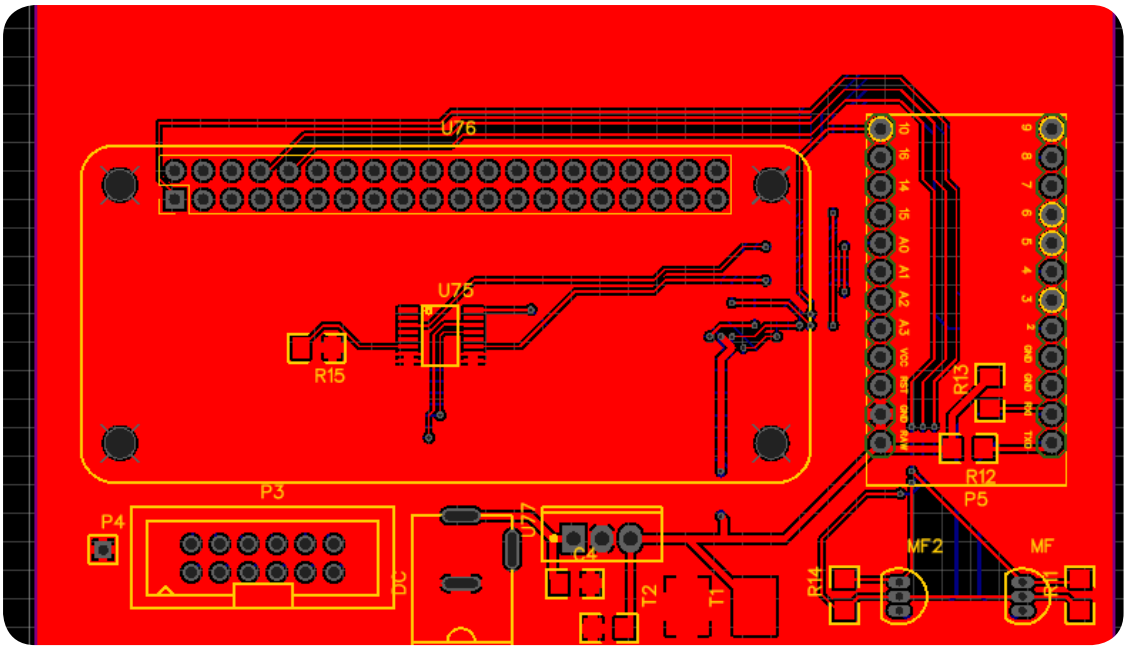
TITLE: Vessel		REV: 1.0
Date: 2018-10-09		Sheet: 1/1
EasyEDA V5.7.26	Drawn By: Rupert Moore	

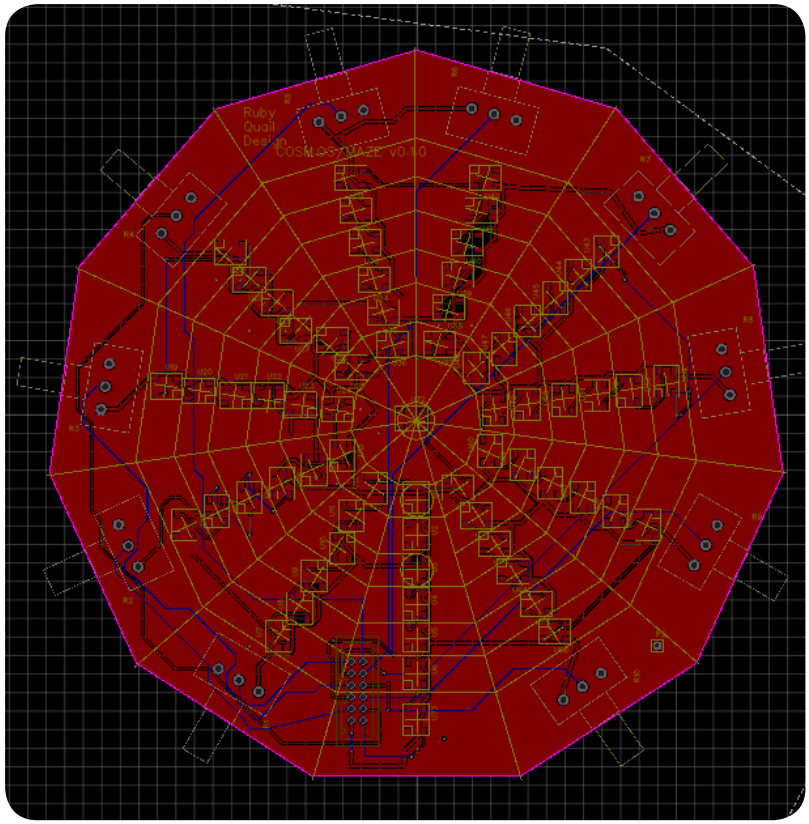
Schematic: Maze

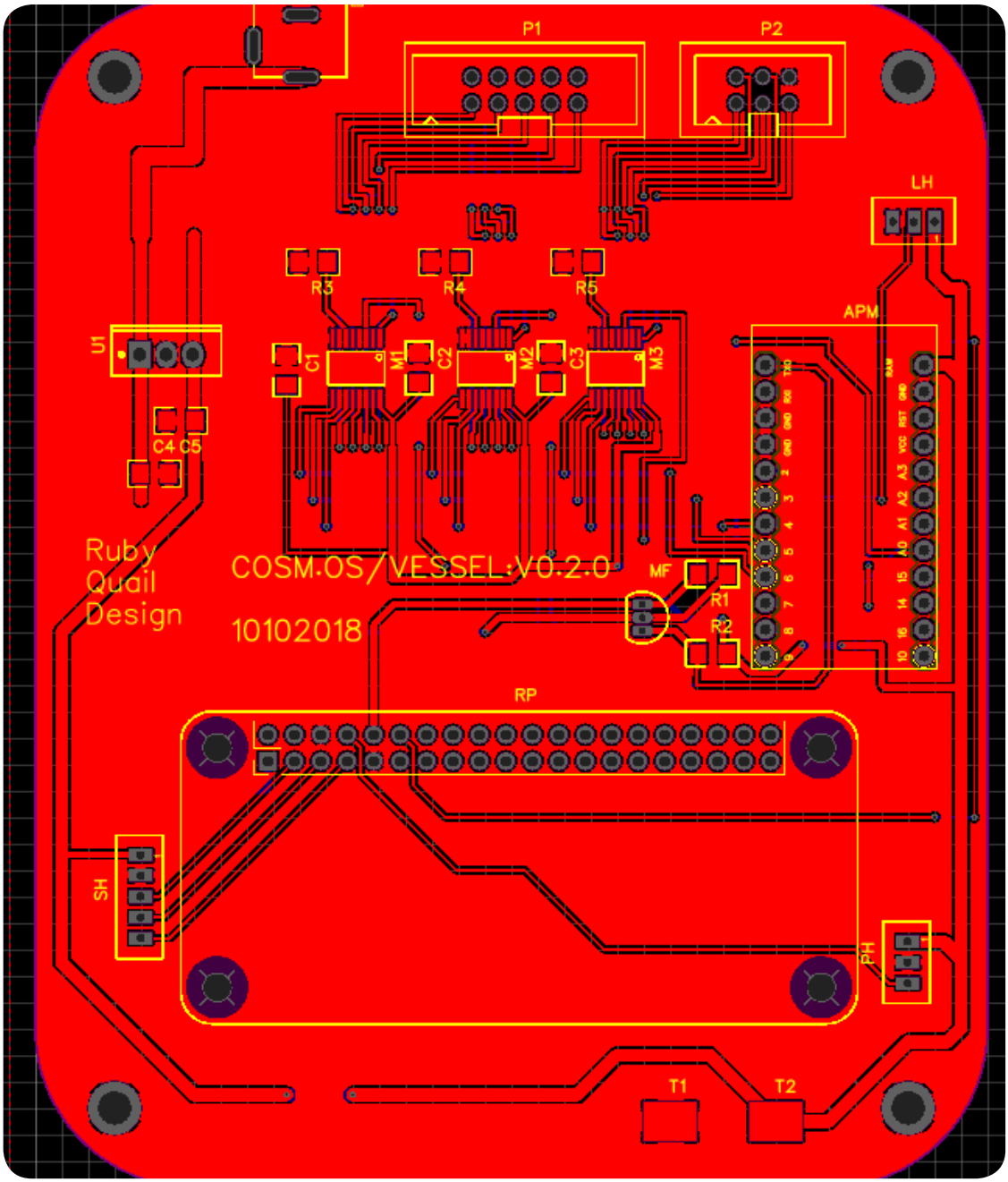



4
5
A
B
C
D
+5V
U76
RPI 3 ZERO Outline
3V3
1
17
2
4
7
11
12
13
15
16
18
22
29
31
32
33
35
36
37
38
40
3V3
3V3
5V
5V
GPIO4
GPIO17
GPIO18
GPIO27
GPIO22
GPIO23
GPIO24
GPIO25
GPIO5
GPIO6
GPIO12
GPIO13
GPIO19
GPIO16
GPIO26
GPIO20
GPIO21
Raspberry Pi
Zero Hat
GND
GND
GND
GND
GND
GND
GND
GND
GND
6
9
14
20
25
30
34
39
GPIO2/SDA1
GPIO3/SCL1
3
5
GPIO14/UART0_TXD
GPIO15/UART0_RXD
8
10
PITX
PIRX
GPIO10/SPI0_MOSI
GPIO9/SPI0_MISO
GPIO11/SPI0_SCLK
GPIO8/SPI0_CE0_N
GPIO7/SPI0_CE1_N
19
21
23
24
26
ID_SD
ID_SC
27
28
T1
5V Pad
+5V
T2
GND PAD
P5
Pro Micro - 5V
ARTX
ARRX
POTSW
DATA
POT4
8_9POT
POT5
POT6
12
11
10
9
8
7
6
5
4
3
2
1
TXO
RXI
GND
GND
D2
D3
D4
D5
D6
D7
D8
D9
RAW
GND
RST
VCC
A3
A2
A1
A0
D15
D14
D16
D10
13
14
15
16
17
18
19
20
21
22
23
24
+5V
POT3
POT2
POT1
POT0
POT7
TITLE:
MazeControlBoard
REV: 1.1
Date: 2018-10-16
Sheet: 1/1
EasyEDA V5.8.16
4
5

Maze and Vessel PCBs






P1
P2
LH
R3
R4
R5
APM
U1
C1
M1
C2
M2
C3
M3
C4 C5
Ruby
Quail
Design
COSM.OS/VESSEL:V0.2.0
10102018
MF
R1
R2
RP
SH
PH
T1
T2