RTD

Proceedings of the 4th Biennial Research Through Design Conference 19–22/03/2019

Thorn, E., Quinn, A., Benford, S., Koleva, B., Preston, W., Mortier, R., Egglestone, S. R. 2019. 'We are all programmers now: From bits to blobs'. In: Proceedings of the 4th Biennial Research Through Design Conference, 19-22 March 2019, Delft and Rotterdam, The Netherlands, Article 32, 1-14. DOI: https://doi.org/10.6084/m9.figshare.7855901.v1.

Method& Critique *Frictions and Shifts in RTD*

We Are All Programmers Now: From Bits to Blobs

Emily-Clare Thorn[1], Anthony Quinn[2], Steve Benford[1], Boriana Koleva[1], William Preston[1], Richard Mortier[3] and Stefan Rennick Egglestone[1]

[1] University of Nottingham, Nottingham, UK
emily.thorn@nottingham.ac.uk

[2] University of the Arts London: Central Saint Martins, London, UK
a.quinn@csm.art.ac.uk

[3] University of Cambridge, Cambridge, UK
richard.mortier@cl.cam.ac.uk

Abstract: Presented here is a research-through-design exploration into a collaborative process of developing a computer technology in which the visual designer has control. This paper presents the iterative negotiations and key turning points in the process of developing commissioned artifacts used to create and push the boundaries of the software. During this process visual designers learned how to creatively exploit the technology, hack the parameters, work around rules and play with accidents to create technologically enriched interactive patterns. We discuss the designer's role in this software development process as the link between the technology and the user.

There is a shift from the seperation of programming codes and adding imagery later to designing imagery that is interactive.

Keywords: Visual markers - topological - threshold - patterns - mobile applications patterns

Thorn, Quinn, Benford, Koleva, Preston, Mortier, Rennick Egglestone | Wallpaper by Elizabeth Jeal and Lilli Cowley-Wood

Introduction

It is becoming ever increasingly obvious that interactivity within objects is expected. For example, our adverts on television are becoming interactive with the use of Spotify, images on the street enticing you to interact with them through your smart phone and magazines offer special deals by scanning their pages. Within human-computer interaction (HCI), Tangible and Embedded Interaction (TEI), embedding the digital into the material, is well established and has started to infiltrate our everyday lives. As a result the idea of scanning visual codes to trigger digital material is no longer a new phenomenon. As we have moved from barcodes to QR codes into Augmented Reality (AR) technologies, the world of image recognition has become well established. Barcodes and QR codes are used as physical hyperlinks to digital information. Visual marker technologies such as Blippar [Blippar, 2018], Goggles [Google.co.uk, 2018], String [Poweredbystring.com, 2018] recognise partial or whole images and display AR content.

Our technology, Artcodes, is a visual image recognition technology that uses a topological approach. It builds upon a system called 'd-touch' by Enrico Costanza [Costanza and Huang, 2009] which allows for the construction of machine-readable images by following a series of simple drawing rules. Artcodes has moved beyond the simplicity of d-touch by adding visual additions and enriched interactions through the engagement of designers in collaboration with computer scientists and programmers. As part of this work we had a core team consisting of computer scientists, programmers and designers. For various artefacts we commissioned artist and designers to work with us to create the final pieces. In this paper we refer to designers as a whole, for each product we commisioned specalised designers relavent to the piece, such as ceramists, illustrators, textiles and graphic designers.

This paper describes the iterative creative process through which this system evolved, from its inception where the designers followed the original rules for constructing valid codes, to later versions where the rules were challenged and pushed to make the system adhere to the aesthetic requirements of the designers. Over the course of the project we developed many different artefacts. Here we will comment on ceramic tableware, a guitar and wallpaper to explore the iterative negotiations between designers and programmers to develop a truly open, reconfigurable, and agile interaction system. We describe how the designers learned to work with and creatively exploit the technology, hack the parameters, work around rules and play with and embrace accidents to create enriched interactive patterns. This paper is divided into four themes that emerged through our analysis of the collaborative design process: terminology, opening the system, how the human and computer see differently, and material knowledge. Each section describes how these feed into the artefacts we produced. Figure 1 shows some examples of final artefacts produced using Artcodes.

Figure 1. From left to right: section of wallpaper, Carolan Guitar and Busaba tableware.

Research through Design

In HCI, Zimmerman and colleagues [Zimmerman et al. 2007] states that artefacts are important as 'concrete embodiments of theory and technical opportunities', and Gaver [Gaver 2012] discusses how Research through design (RtD) is a broad umbrella that encompasses a diverse set of practices.
We adopt these processes of exploration and documentation whereby the visual designer feeds back to the programmer, who in turn feeds back to the designer (see Figure 2). These iterations proved vital in developing the software and allowed us to move away from a domino-like image (see Figure 3) into drawing patterns, in any aesthetic style, that happen to house a code. Our aim in this research was for codes to not look like codes, using software that leaves the designers in control.

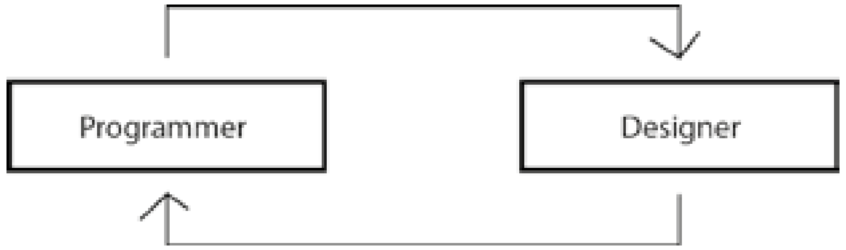

Figure 2. Iteration between programmer and designer as part of our Research through Design method.

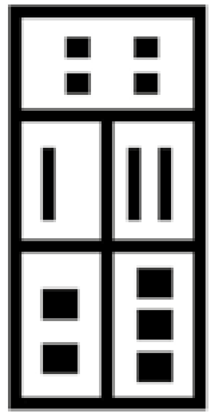

Figure 3. An example of a d-touch code. We refer to this as to looking like a domino.

Terminology

Within disciplines, terminology that is specific to that discipline becomes second nature to those 'in the know'. Every area uses field-specific jargon that becomes entrenched in the daily vocabulary and does not translate outside of the discipline. It became apparent very quickly with both programmers and designers around the same table that their vocabularies were very different. During these initial meetings, programmers used discipline-specific vocabulary which the design team adopted. It was only during the first workshop, where the in-house design team explained the drawing rules to fellow designers unfamiliar with the Artcodes concept, that a new lexicon was established. This was not a conscious decision but an organically developed new terminology to describe the process. They developed a new vernacular around the artistic communication of a complex technological field.
The new terminology needed to work in three ways: 1) we needed computer scientists to understand terminology used within design and art, 2) we needed designers to understand the terminology used in computer science, and 3) we needed the public to understand both.
We started with the 'region adjacency tree', terminology inherited from d-touch, comprising the 'root', 'branches', and 'leaves' (see Figure 4). This terminology comes from the tree data structure the soft- ware creates from the contours of the image and then analyses to detect Artcodes. After working with designers, these terms evolved to 'boundary', 'region', and 'blob' to describe the detectable elements of a design and/or pattern. This new terminology is more geographical in nature and can be more familiar to someone used to arranging visual elements on a page.

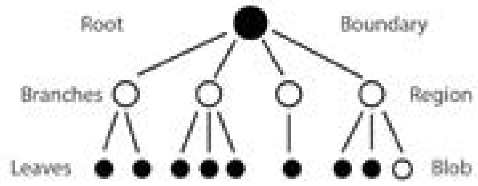

Figure 4. Shows on the left the d-touch terminology used and on the right the new terminology.

The change from the field-specific terminology into a common language was eventually adopted by both programmers and designers, and the new terms are still used to describe Artcode drawing rules. Using more descriptive words, graphic visualisation emerged as a method to explain the structure. This is demonstrated in Figures 5, 6 and 7.

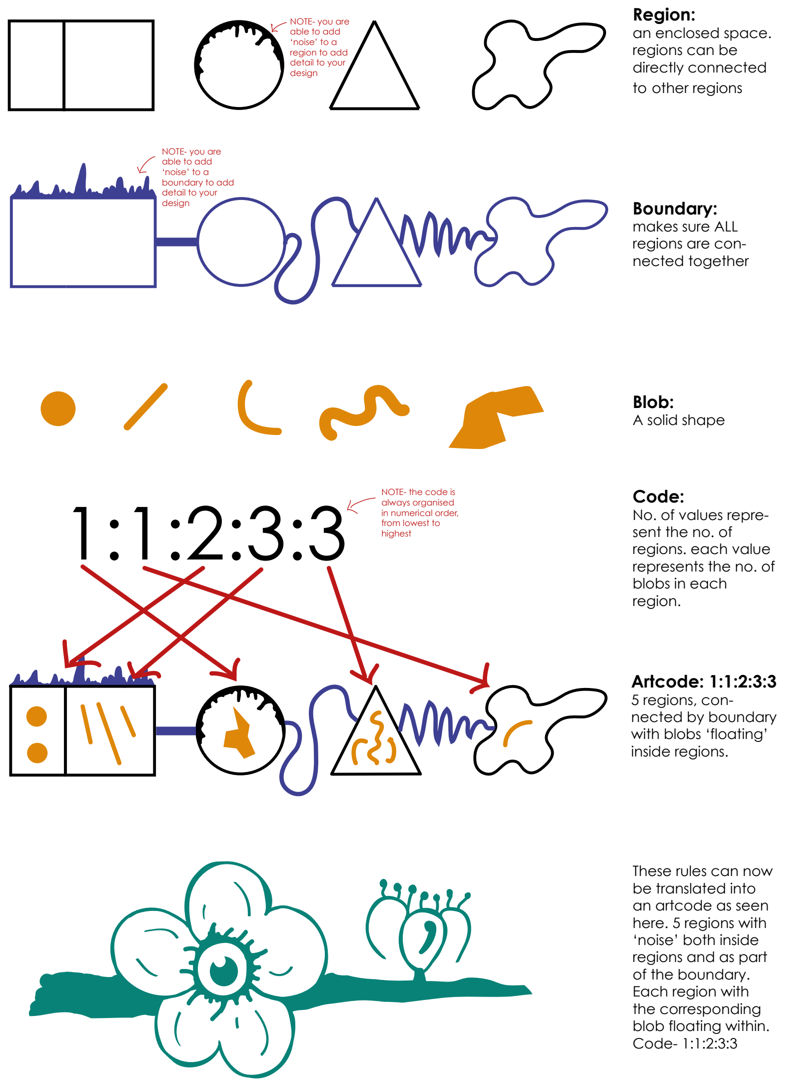

Figure 5. Artcodes drawing rules and elements using new terminology: each Artcode is made up of regions (enclosed shapes), boundaries (connecting the regions) and blobs (solid, unconnected shapes inside regions).

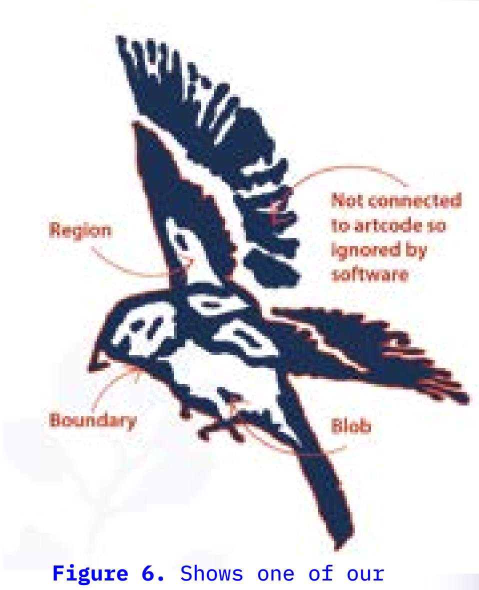

Figure 6. Shows one of our graphical visualisations used in the teaching of the artcodes rules

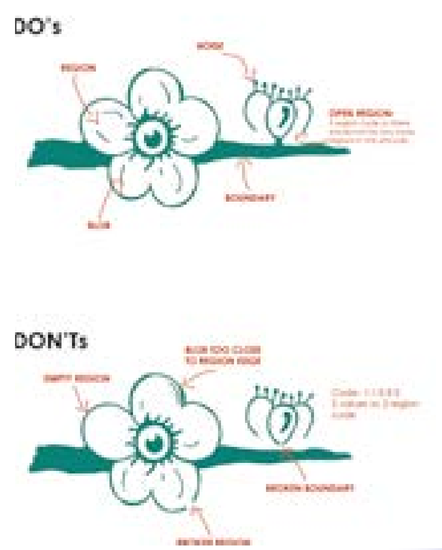

Figure 7. Summary of artcodes drawing rules

Opening the system

Initially, the technology involved a programmer to develop a section and a designer to test how it worked and how it could be used. This produced rudimentary images and did not allow for much freedom for creativity. This is especially notable in the original development of Artcodes, in which a list of 'parameters' were given, which restricted even the size of the image. Our process essentially repeated the domino-like images we were trying to move beyond. These can be seen in examples from our first project involved scannable tableware (see Figure 8), which used the first version of the app. Our own parameters resulted in these simplistic designs. Designers were able to produce images, however, the artcode was separate from the rest of the design and not fully immersed into the composition of the image.

Rules were pushed or ignored by the designers and sent back to the programmers to allow for more freedom because the restrictions did not permit enough artistic license to develop the images from codes into aesthetically pleasing designs

Some of the key breakthroughs occurred when the designer and programmer discussed and tested in depth how the system processes the image. Through this conversation, other areas of exploitation were revealed.
As part of processing the image, the software thresholds the image using Otsu's method [Otsu 1979], reducing the light colours to white and dark colours to black. However, as this method works by finding a group of light colours and a group of dark colours on the image's histogram what is considered light and dark is relative to other colours in the image, and designers have no guaranteed way of perceiving or controlling this while they work.

As part of this process the designers discovered that certain colours were 'ignored' by the system. It was only upon opening the system for the designers to visually see the thresholding process that they could understand how different colours were read by the software. Some colours in one image could be ignored or appear as white while the same colours in another image would be seen as black, all based on the colours within the image as a whole. The exploratory image in Figure 8 demonstrates the thresholding issue. The top image (Figure 8a) shows the original design. Figure 8b shows how, after thresholding, the detection algorithm sees only the main flower and ignores the flowers in the background. Figure 8c shows us zoom in or concentrate on one of the smaller flowers that was ignored previous. Figure 8d shows how thresholding now allows the flower to be seen. In the hands of a designer who understands the principles of Otsu's method, thresholding ambiguity can be used to create depth in interaction within a single image, allowing for various degrees of interaction. For example the large flower could trigger one point of interaction when scanned by the user for afar, then if that user was the to step closer to the image another interaction would be unlocked by the smaller flowers. Furthermore, this collaboration between technologists and designers enhance the aesthetic value of interactive imagery without affecting code recognition.

This exploration of software constraints led to further discussions into the use of colours. The questions brought by the designers concerned whether they were able to use specific colours in the design and allow for only those colours to be detected or filtered out. This led to the development of colour filters to be included in the software. For example, in the original fish design (see Figure 9a) the Artcode is in both of the large fish. However the fish are floating within the image with no depth to the visual design. Figure 9b shows a far more complex design in which the fish and flowers overlap. Under thresholding alone, the codes could not be read. However, with the use of a red colour filter (figure 9c) all of the flowers are removed from the image as recognised by the new version of the software.

To allow for the greatest amount of creativity, the designers needed to not only shift through these filters but also have the tool to check their work while producing it. A filter option was created in the software that allowed them to test images as their designs progressed. Two filters were implemented. An RGB filter removed the named layers (red, green, blue) and retained the remainder of the visual information. RGB describes the colour space used by mobile phone cameras. CMYK (cyan, magenta, yellow, black) colour space, used for printed media, is handled by the CMYK filter. This converts a video feed to CMYK and removes named layers.

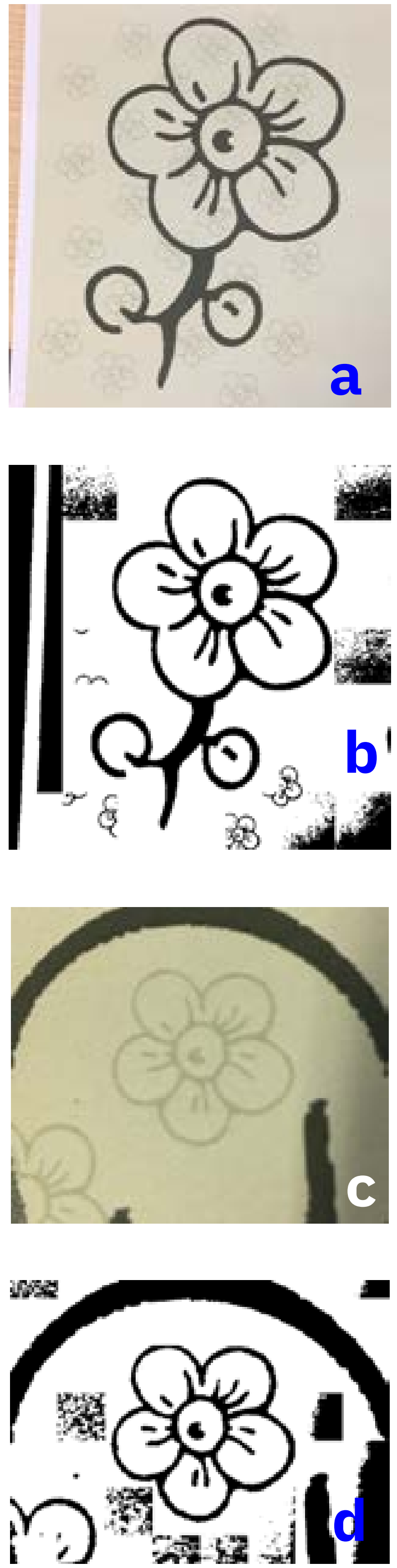

Figure 8a, b, c, d. Thresholding using Otsu method.

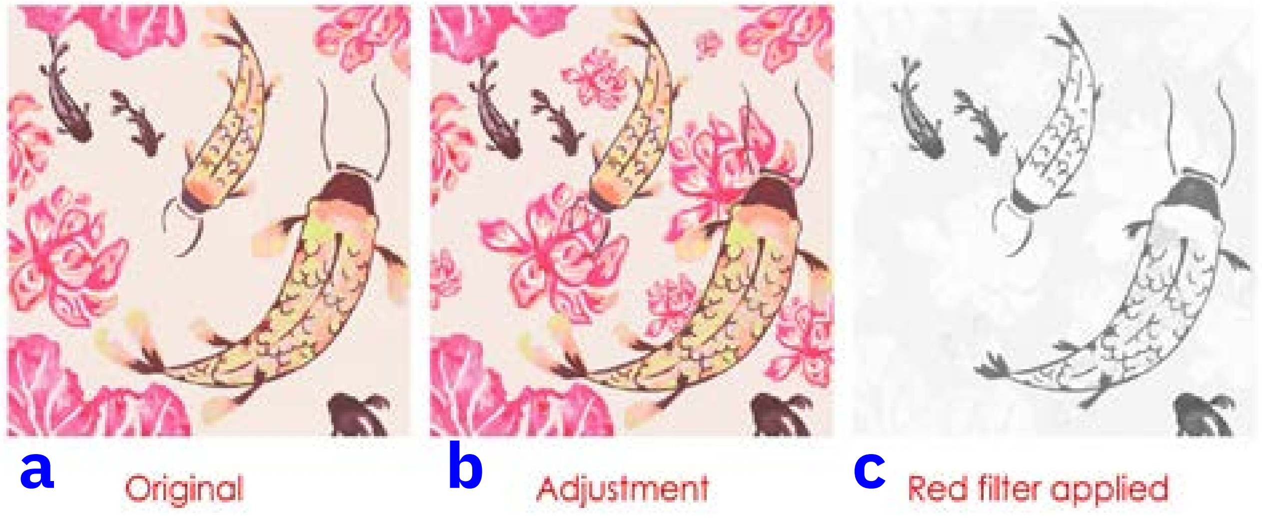

Figure 9a, b, c. Red colour filter explained.

This collaboration established the need to add new layers into the software, which would be visible through the software during development. These layers were used throughout the design process for testing and producing images. Each segment could be tested while being drawn and modified, ensuring the images were still being read by the software. It was envisaged, in some cases, that printing the products could result in the need to return and reduce the contrasts or density of colours as discussed below.

Understanding how the human and computer see differently

It became clear that through the exploration into thresholding and colour filtering that the designer was the bridge between both the software and the finished product as it would be experienced by the viewers. As part of this development, the designer needed to understand how both the computer and the human sees an image. These challenges have been discussed by Morrison and colleagues [Morrison et al 2014]. This 'bridge' became the designers' key role in the software development process. Understanding the topological drawing rules such as disconnected elements, line thickness, thresholding and colour filters, plus the pipeline of visual interpretation, meant that designers could develop more aesthetically pleasing Artcodes.

Designers could use the topological rules to aid in the aesthetics and flow of the design. Drawing rules may not be a nuisance but be used as a benefit. For example, as part of the drawing rules it is essential to avoid broken regions or disconnected elements. While this is still true in order for a code to be read, designers used this as a part of their designs. Instead of creating an Artcode where the code floated within the design, designers (who had become more skilled at hiding codes) would often place the code within both the foreground and background of the image using disconnected elements to hide it. Figure 10a shows a crest as part of the wallpaper design. Figure 10b shows the yellow outline of where an Artcode is located in one design. It is not encased in a single element of the crest but spread across various places. To the viewer these shapes are separate elements of the design. However, sections of what appear to be separate elements together make up a full Artcode.

Figure 10. Section of wallpaper. Top image shows a crest as part of the design. The bottom image highlights in yellow the outline of the artcode. Highlighted in orange is the outline of each region. The software will look for enclosed shapes, regions, count how many are connected, by the boundary, then count the number of blobs inside each region to build the code.the rest of the image is ignored by the software

Material Knowledge

Beyond the development of a rigorous system, the negotiations between technologists and designers in creating artefacts illustrates the capabilities of the system. This has been key to developing a portfolio of viable products resulting from in-the-wild applications of the software. Leading in turn to a significant increase to the design team, growing from two to eight in order to create additional resources and design diversity.

To live test the technology, a research probe was set up in collaboration with Busaba Eathai Restaurant during London Design Week. A team of visual artists were commissioned to produce three design collections for ceramic tableware, menus and placemats. These were used as part of a hypothetical service structure and collected into a pattern book.

The designers were introduced to the software through a hands-on demonstration workshop. This allowed them to experiment with the rules and constraints of the system. It should be noted here that this workshop cemented the use of the newly developed lexicon, with the new terminology being adopted in order to effectively teach and translate the systems limitations and parameters. One striking finding was the diversity of their approaches to the interactive image construction. For example, some preferred to draw an image and then populate with blobs, whilst others constructed the image in line with the topographical nature of the parameters.

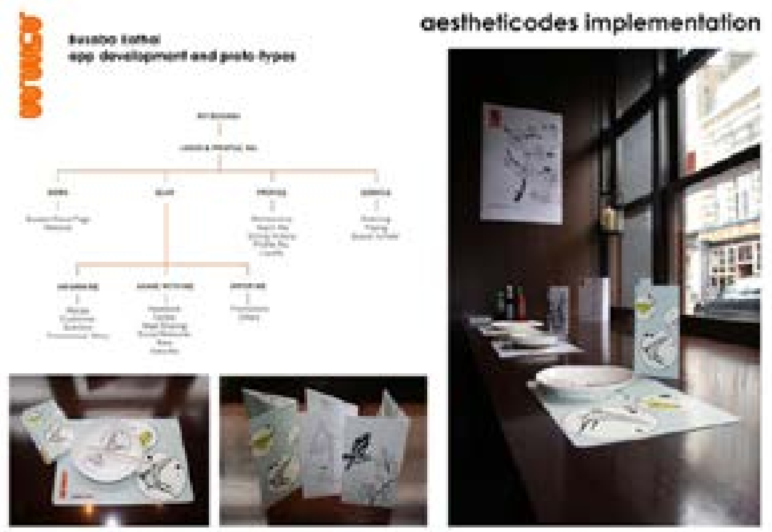

Figure 11. Busaba app tree diagram and photos of menu, placemats and plates

As the pattern book began to take shape, the team began producing prototypes to test whether the images would be successfully read by the software once transferred to physical objects. Ceramic tableware was created by printing the designs as enamel transfers which were then applied to the plate before firing. The firing effectively fused the design to the surface of the plate and only in this state could it be known whether the camera could see the image.

The transformative nature of the ceramic printing process brought complications to testing. As ceramic colours do not fully mature prior to the firing cycle, they often differ in colour after firing. The printing uses a cover coat, which is a plastic film that lifts from paper to the ceramic object and retains an oily sheen until fired. This has the effect of creating a reflective coating over the image.

The successful process of firing and testing resulted in an empirical approach of resolving design ideas and material limitations. Through trial and error, designers and developers gained a working knowledge of the idiosyncrasies of the relationship between what the camera sees and what the system requires, specifically in this case for the process of producing fired ceramics. Designers could anticipate and solve some of the problems before firing.

A similar process of working with material knowledge is discussed by Benford and colleagues [Benford et al. 2016] when producing another artefact, the Carolan Guitar (see Figure 12), a complex design that needed to balance visual aesthetics with the sound and acoustic integrity of a musical instrument. This resulted in extended discussions and alterations in both the aesthetics and techniques to overcome these obstacles.

Story

Arguably one of the key benefits to the Artcodes system is the ability to design interactive motifs and images that are visually relative to the intended interaction, context, provenance and/or narrative of the object they adorn. The possibilities and potential of the relationship between designer and programmer often provides the design with direction for development. Two very different fields, computer programming and visual design, engaged positively together to tackle the challenges of imbuing meaning and visual cues into an interaction experience.

The potential for this depth of interaction became apparent during the Busaba collaboration. The first iteration of interactive objects designed in collaboration with Busaba had the intention of road testing the technology and its live interactive capacity. The objects were successful to test the parameters, speed and user experience, but lacked the playful and meaningful interaction that the system afforded the designer and user. Following a reflection upon this in the initial test, a further challenge was conceived in the Busaba setting to wed the interaction to the Busaba service structure concept, which tells the story of how the restaurant is blessed by Buddhist monks once a year and adheres to principles of Feng Shui.

The designers and programmers began to explore the layers of complexity around building visual interactions that were aesthetically informed by the narrative and meaning of an object and its context. This new layer provided the missing link, to answer the question ‘why bother?’.

This became increasingly evident in development meetings.

Interestingly, the challenge of designing imagery relating to context and narrative provided the motivation to push the visual complexity in terms of design ambiguity and gestalt clues within the design.

It could be argued that this event provided the DNA for how the research would develop. It has evolved exponentially since this point with a series of narrative-inspired research probes that challenge the user to explore the meaning of an interaction, additionally learning more about its context as an enticing object in the environment to interact with.

Conclusion

A Research through Design method and an understanding of terminology resulted in a fully open system that allowed the designer to explore and understand how both the human and system sees. This allowed for the development, exploitation and production of contextually rich artefacts and objects where the story matters. The designer had a key role to play, and as discussed, became the key driver of various processes.

Discussed in this paper are three of the artefacts produced to support developments in the Artcodes software. Since then, the abundance of objects that were produced continues to grow, from ceramic tableware, clocking-in factory cards and guitars to birthday cards, wallpaper and wearable pieces. These showcase a diverse use of the software, allowing it to be further developed by artists and designers in and on a range of materials, techniques and artistic styles.

By allowing the designer ‘behind the scenes’ of the software and the subsequent expansion of an open system, our process enabled greater developments than might have been available to a less design-led approach. It was the designer who needed to balance all of these facets to produce an aesthetically beautiful image that not only worked with the software, but also functioned within the context and the surroundings.

It was important to understand that some logical and mathematical processes of the technology hindered the practical realities of a design-led approach. Through a collaborative approach the artcodes software was realised. This was key in moving away from simplistic ‘domino’ aesthetics into a world where beautiful and intricate designs can be adopted and embedded by the technology. The ultimate aim is to ensure the engagement with the artefacts because of the aesthetic value and interaction interface in equal measures.

Figure 12. Carolan Guitar. Image shows the user scanning a section of the guitar to reveal the link to information about the guitar and videos of artists playing it.

Figure 13. Wallpaper installed in a family home inside a bedroom

Figure 14. Busaba menu, placemat and bowl installed in Busaba Eathai restaurant

References

Blippar. (2018). Augmented Reality (AR) & Computer Vision Company | Blippar. [online] Available at: http://blippar.com/

Benford, S., Hazzard, A., Chamberlain,A., Glover, K., Greenhalgh, C., Xu, L., Hoare, M., and Darzentas. D., (2016). Accountable Artefacts: The Case of the Carolan Guitar. *In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI '16*

Costanza, E. and Huang, J. (2009). Designable visual markers. *Proceedings of the 27th international conference on Human factors in computing systems - CHI 09.*

Gaver, W. (2012). What Should We Expect From Research Through Design? In: *Proceedings of ACM Conference on Human Factors in Computing Systems (CHI 2012)*.

Google.co.uk. (2018). Google Goggles. [online] Available at: http://www.google.co.uk/mobile/goggles/

Morrison, C., Smyth, N., Corish, R., O'Hara, K., and Sellen.A., 2014. Collaborating with Computer Vision Systems: An Exploration of Audio Feedback. *In Proceedings of the 2014 Conference on Designing Interactive Systems (DIS '14).*

Meese, R., Ali, S., Thorn, E., Benford, S., Quinn, A., Mortier, R., Koleva, B., Pridmore, T., and Baurley. S., 2013. From Codes to Patterns: Designing Interactive Decoration for Tableware. *In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13).*

Otsu, N. (1979). A Threshold Selection Method from Gray-Level Histograms. IEEE Transactions on Systems, Man, and Cybernetics, 9(1), pp.62-66.

Poweredbystring.com. (2018). String®. [online] Available at: http://poweredbystring.com

Zimmerman, J., Forlizzi, J. and Evenson, S. (2007). Research through design as a method for interaction design research in HCI. *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '07*.

Figure 15. Carolan Guitar. Detaied with artcodes inlay. Multiple artcodes allows for various points of interaction.