Published in 2007 by
Diego Baca
59 Inverness Terrace
Room 10
London, W23JN
Tel: +44 20 7229 8019
e-mail: diegobaca@gmail.com
www.diegobaca.co.uk

VIRAL BLOCKS

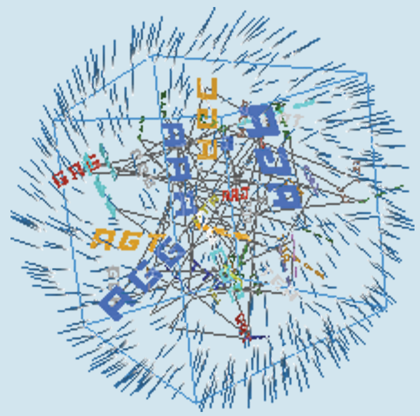
Diego Baca

This major project report is submitted in partial fulfilment of the requirements for the degree of Master of Graphic Design at the London College of Communication.

November 2007

Paul McNeil, Advisor
Subject Lead tutor
School of Graphic Design, London College of Communication, London

Russel Bestley
Course Director
School of Graphic Design, London College of Communication, London



THANK YOU ...

Paul McNeil for your feedback, guidance, and support.

Mariana Baca for your patience, insights, and amazing coding skills

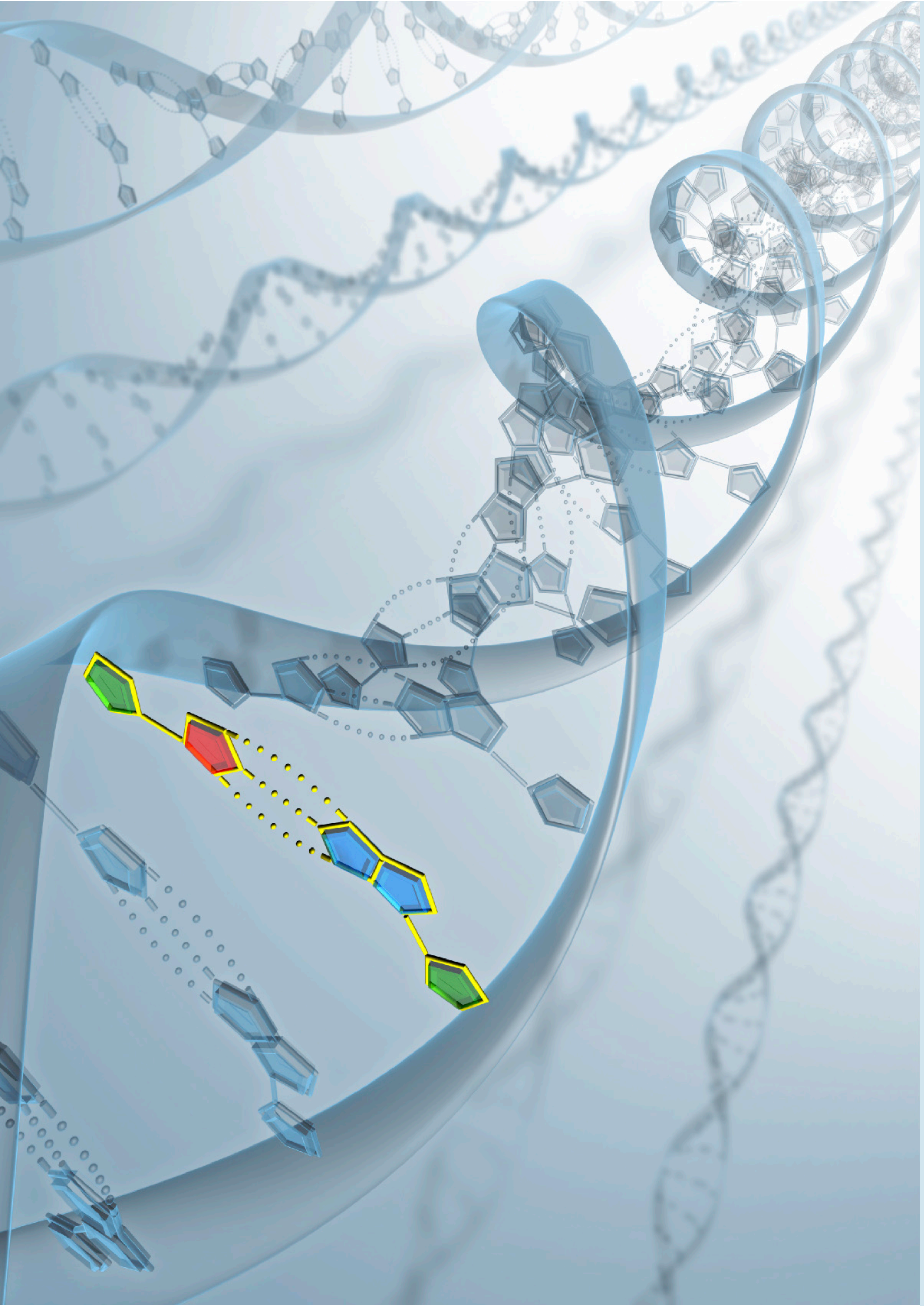
Claire Reimer for your great support and editing skills. You were always there when I needed help.

Charo and Jorge Baca for being my parents. Thank you for giving me this opportunity.

Jorge Cardoso for your collaboration.

MAGD 2007 classmates and tutors for your inspiration and guidance to become a better designer.

The Processing Community you guys rock, keep up the good work.



CONTENTS

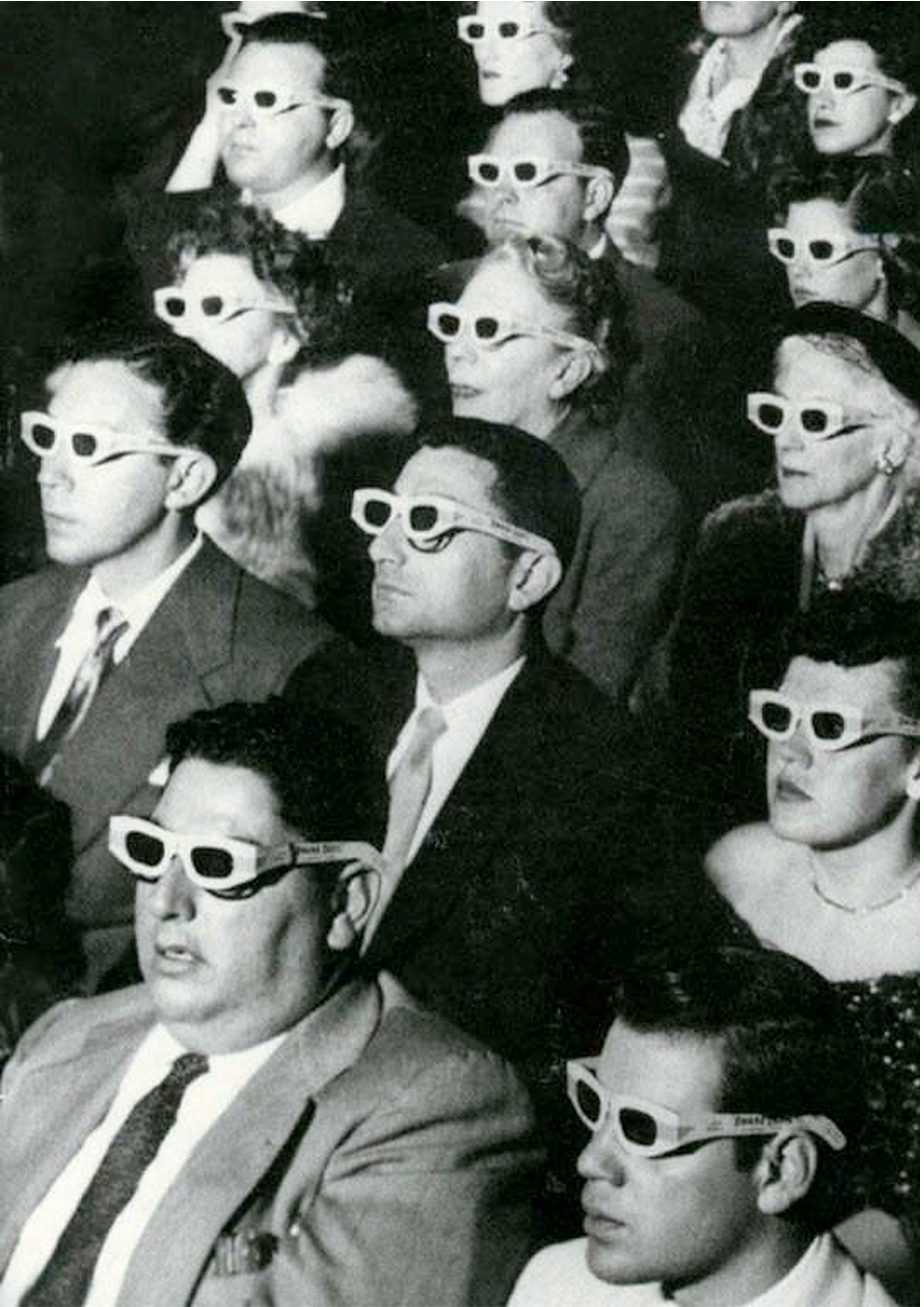
Top: early designer using a Gutenberg press. Below: modern designer using a computerised system. Both have the same goal, but use completely different processes depending on their time period's technology.

1 - INTRODUCTION

New technologies have redefined all aspects of design, beginning at the initial sketch and ending at the final printout. From these new advances there has been a revolution in how users interact with design pieces. The field of interaction design was born as a result of the rapid innovations in computer technology that took place in the early 1980s in labs such as Xerox's PARC and Adobe Labs. One particular field which advancing technology has changed substantially is typography. Typography has evolved from a very physical art, involving heavy wood or metal bits, into a virtual art made up of byes, vectors and Postscript code.

This major project report documents my personal research into interaction design and computational typography and how both are combined with an intention to change or enhance the way we communicate.

The report begins by giving an overview of the field of interaction design and its objectives. It then briefly documents how technology has changed typography and examines several contemporary computational typography works. This is followed by descriptions of my own computational typography exercises, which lead to the eventual creation of Viral Blocks, an interactive program for visualising viral structures. Finally, it ends with conclusions and a critical evaluation of the project.



People are captivated by technologies which augment their senses. Here a crowd is baffled by 3D motion pictures

2 - INTERACTION DESIGN

Fig. 1 & Fig. 2: The computer game Doom (top) and MTV (bottom) are staple cultural phenomena which influenced members of Generation Y.

2.1 ORIGINS OF MULTIMEDIA

The primary focus of my research has been the field of interaction design. Gillian Crampton Smith, director of the Interaction Design Institute (IVREA), sums up interaction design in one sentence saying it's "about shaping our everyday life through digital artefacts – for work, for play, and for entertainment". Crampton Smith's statement holds the crux of this project, because it presents a multi-goal view on design. It isn't enough for something to be successful simply at one level but now we expect, and in fact many of us require, that our everyday objects perform with the efficiency and flexibility of a Swiss Army knife.

The most obvious example of multifunction is cellular phones. They have evolved from big clunky devices, which could barely achieve a local call, to multifunctional marvels, which can be used for work, play and entertainment. This multifunction obsession has invaded every object in our everyday life; multifunction can be found in airplanes, cars, watches, televisions, sound systems even refrigerators.

The primary instigators of this obsession have been the computer and the Internet, due to their sheer power to draw information from various types of media simultaneously. The people who have been most influenced by the computer and the Internet are the ones who grew up during the years which the computer matured from the enthusiast to the consumer phase. Members of this period are known by several names: Generation Y, MTV Generation, Doom Generation, Echo Boomers, and Internet Generation. These multimedia and technology junkies, such as myself, grew up in a world where music was video, games meant the arcade, and social freedom came via the Internet.

It is no surprise that we have become comfortable with being bombarded by information from several sources simultaneously and that when we are not able to access this media conglomerate we feel disconnected or out of the loop. Therefore, it was during this period that interaction design bloomed, drawing from its interdisciplinary roots in industrial design, human factors, graphic design and human computer interaction, in order to make these multifunctional devices "usable, useful, and fun" (Saffer, 2007).

2.2 SENSORY AUGMENTATION

The first task I set to accomplish was to identify the reason people find multimedia and multifunction so appealing. It boiled down to the simple fact that people want the ability to have their capabilities augmented. There are four areas which human capabilities can be augmented by interaction design, which are identified by Doug Engelbart's in his 1962 paper titled "Augmenting the Human Intellect: A Conceptual Framework", they are: artefacts, language, methodology and training.

1. **Artefacts**: are physical objects designed to provide human comfort, the manipulation of things or materials, and the manipulation of symbols.

2. **Language**: is the way in which the individual classifies the picture of his world into the concepts that his mind uses to model that world and the symbols that he attaches to those concepts and uses in consciously manipulating the concepts ("thinking").

3. **Methodology**: the methods, procedures, and strategies with which an individual organises his goal-centred (problem-solving) activity.

4. **Training**: the conditioning needed by the individual to bring his skills in using augmentation means 1, 2 and 3 to the point where they are operationally effective

(Engelbart, 1962).

The ideal scenario according to Engelbart is achieving an improved and augmented state where everyone is trained to efficiently use artefacts, language and methodologies. This phenomenon is happening all over the world as the workforce now relies on multimedia artefacts, language and methodologies created via augmented multifunctional systems. Many members of newer generations like the members of Generation Y are able to benefit from these augmentations with minimal training because they have already been exposed to similar systems during childhood. Modern consumers are increasingly demanding devices that stimulate or enhance more than one sense, providing them with a fully enhanced multi-sensory experience.

This need for enhancing what we currently have has lead interaction designers to ask one fundamental question when designing for interaction: "Why does it have to be like that?" This question is fundamental for augmenting interactions because it allows designers to change their current perspective. The interaction design paradigm is the current pattern for the way people think about something (Bill Verplank, 2001). It is important to challenge these paradigms because it allows interaction designers to create new products and services from the same object. As Brendan Dawes states, "something that was once loved can actually be brought back to life and mutated into a different useful, fun object with a full future ahead".

2.3 INTERACTION DESIGN VARIABLES

In order to understand how to mutate an object effectively, my second task was identifying the variables which interaction designers manipulate. According to Dan Saffer, there are six variables, which are the building blocks of interaction design and are the basic set of resources for manipulation: motion, space, time, appearance, texture and sound.

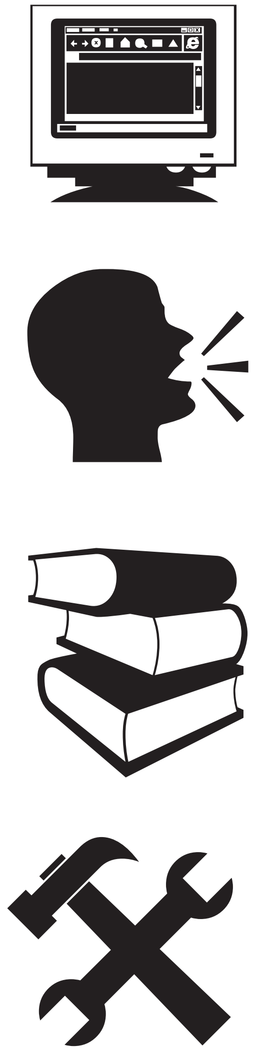


Fig. 3: Doug Engelbart suggests there are four areas where human capabilities can be augmented: artefacts, language, methodology and training.

Motion
First and foremost, interaction design is concerned with communication. Interaction implies communication either with another person or another thing. Through interactions, behaviours are formed which are unique to the type of communication. These behaviours are dictated by motion, which orchestrates the interaction. When we talk, we move our vocal cords and our bodies to communicate, or use our fingers to type when we send an email. "All behaviour is, in fact, motion: motion coloured by attitude, culture, personality, and context... Without motion there can be no interaction" (Saffer, 2007).

Space
"All interactions take place in a space" (Saffer, 2007). The space can be 2D, 3D, physical or virtual. Interaction design often uses a combination of these types of spaces. For example, using a keyboard to type a paragraph on screen is an interaction between the physical, 3D keyboard, and the virtual, 2D type, on the computer screen. Space also provides context for motion (Saffer, 2007), identical interactions can differ significantly in quality, ease and enjoyment depending on the type of space the user occupies.

Time
"Movement through space takes time to accomplish" (Saffer, 2007). Without time interactions would be impossible. Time is important for interaction design because it paces the rhythm and duration of interactions. Some interactions occur in human time while others in digital time (fractions of a millisecond). Regardless of the type, interaction designers must use time appropriately in order to avoid user frustration.

Appearance
Appearance is an important aspect of interaction because it "gives us cues to how [an object] behaves and how we should interact with it" (Saffer, 2007). Appearance, along with texture, are two major sources of what cognitive psychologist James Gibson coined as affordances. Affordances became popular in design in 1988 with the publication of Don Norman's book, The Psychology of Everyday Things. Affordances are inherent visual cues of an object that inform the user how it should be used. For example, the table's flat plane tells the user to set an object on top of its surface. Affordances are contextual and cultural, therefore some object's use might be obvious to some while a mystery to others. Appearances also convey other attributes to the user, such as the value of the object, its safety, its ease of use, etc.

Texture
Texture is a part of appearance. It conveys similar information by the means of how an object feels. Texture communicates affordances, such as whether the object is durable or frail or must be pushed or pulled. It also can convey emotions; a soft fuzzy object has a different emotional response when compared to a hard metallic object. A widespread example of how texture is used to communicate is vibration in mobile phones. Phones usually vibrate indicating an incoming call or voice mail, urging the user to make a behavioural response.

Sound
Sound comprises a small part of interaction design but it is an important part, especially for alerts and ambient devices (Saffer, 2007). Sound has several variables such as, pitch, volume, timbre and tone which can be used to convey information, and increase the quality of interaction.

Having found the reason why people desire augmented interactions, and having understood what variables are available for manipulation, the next step was exploring how interaction designers have started to apply these changes. I choose typography as my area of focus because it is an established art that has been practised through traditional methods for hundreds of years, but has experienced a revolution in the past decades with the appearance of digital media, the web, and other new technologies.

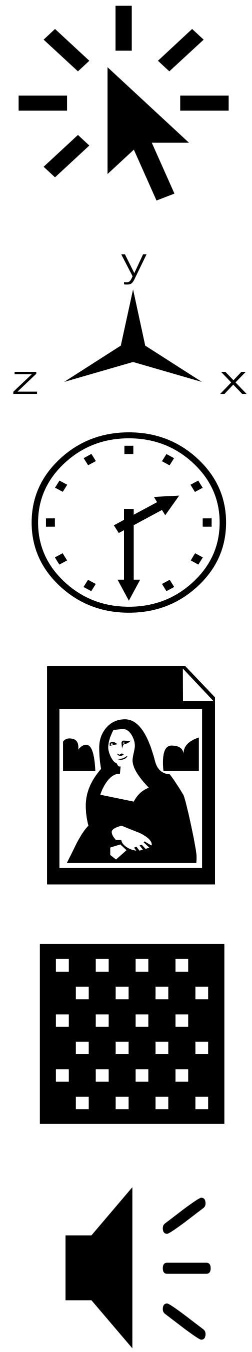


Fig 4: The six building blocks of interaction design: motion, space, time, appearance, texture and sound.

Times New Roman is a popular computer font based on the original Times Roman, san-serif typeface, commissioned by The Times of London newspaper in 1931

3 - NEW TYPOGRAPHY

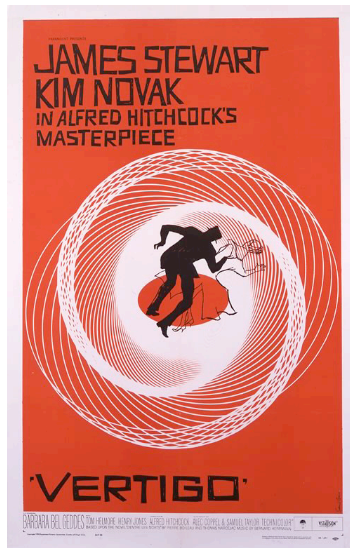


Fig. 5: Poster for the 1958 Hitchcock movie: Vertigo. Saul Bass is renowned for making amazing title sequences for many of Hitchcock's films, adopting kinetic type.

3.1 MANIPULATING TYPOGRAPHY

Typography before the rise of the information age was an art rooted in traditions and constrained by the rules and methods of printing, most which were invented prior the 1900s. Typography was a very physical art. Before the advent of desktop publishing, compositors worked endless hours setting type and managing layouts physically by hand. Unusual layouts which included diagonals or curves, although possible, were extremely time consuming and usually avoided.

The "letter-forms that crack, fly, scream, decay, or otherwise animate as they try to communicate their message" (Cho, 1999) were extremely rare. A breakthrough emerged from Hollywood in the form of film title sequences in the late 1950s with the work of designer Saul Bass. His title sequences were revolutionary and essentially were graphic films within films (Cho, 1999). Saul Bass used type animations in order to set the mood and develop an emotional baseline for the film. His work on the opening sequences of "The Man with the Golden Arm" (1955), "Vertigo" (1958) and "Psycho" (1960) are regarded as some of the best title sequences ever produced.

Similar work has been produced recently by designer Kyle Cooper whose work has been featured in movies such as "Seven" (1995) and "The Island of Dr. Moreau" (1996). Cooper's work mirrors Bass' style by allowing the type in the title sequences to come to life through their motion and interactions with other elements on screen, creating a multi-layered visual narrative. As Peter Cho points out, it is interesting how Cooper uses the theme of genetic mutation present in "The Island of Dr. Moreau" to manipulate the typefaces to a point beyond recognition. Such effect goes hand in hand with the sprit of interaction design, previously mentioned in section 2.2, which tries to create new things by manipulating traditional forms and giving them new life.

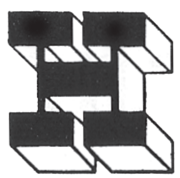
3.2 THE END OF TYPE

The desktop publishing revolution, which rose from the information age, heralded the age of manipulation and customisation of type. Several initiatives in the research labs of Apple, Xerox and Adobe facilitated the revolution by developing new technologies such as the GUI, laser printer and PostScript page description language. Another lesser known, but equally important, initiative was Donald Knuth's work on MetaFont. MetaFont is a programming language used to create vector fonts. Vector fonts are more advanced than bitmap fonts. Because they are based on a set of equations they do not loose quality or resolution when scaled. This system was revolutionary because it introduced parameters to screen-based font creation. Type designers could easily manipulate parameters such as, x-height, width, stroke width, and serifs relatively easy and introduced "the computer science theme of 'abstraction' to type design" (Cho, 1999).

One of the most notable individuals who rode the wave of abstraction was designer and surf aficionado, David Carson. With his trend-setting publication Ray Gun, Carson proclaimed the "end of type" by distorting the type beyond recognition. Carson's chaotic, barely readable and subversive layouts are a perfect example of the extreme level of abstraction that could be obtained with the use of these new technologies. Although Carson promoted a style of rule breaking and of bastard fonts, which was popular in the 1990s, other designers were able to create new guidelines for working with type which apparently had no boundaries.

3.3 DIMENSIONAL TYPE

J. Abbott Miller in his book titled, Dimensional Typography, starts by examining the history of type design and the "traditionally flat and static province of the letter" (Abbott Miller, 1996). Abbott Miller is keen to add that current typographers must acknowledge the fusing of spatial and temporal information with established type traditions converging to create a new approach to typography (Abbott Miller, 1996). Additionally, J. Abbott Miller identifies seven ways which type can be rendered dimensional: extrusion, rotation, tubing, shadowing, sewing, modular construction and bloating. Identifying these techniques is a crucial step for abstracting type successfully and in an informed manner.



The work of designers such as Muriel Cooper and Robert Massin use some of these techniques to present type which is temporal, expressive, and emotional. Cooper's pioneering work at the MIT Media Lab's Visible Language Workshop explored the "information landscape" allowing type to simulate topographical features in the virtual 3D space of the computer screen. While Massin envisioned the page as the stage for type, where it could be stretched, bloated, and rotated to orchestrate a narrative were type is used to visualise characters voice inflections and movements.

Ever since the chaos and rule breaking trend of the 90s, many designers have started to identify new guidelines for typography, as seen on the work of Abbot Miller, Cooper and Massin. As computers become more powerful and versatile, new levels of interaction and abstraction can be used alongside typography. Many designers in the 21st century have incorporated and manipulated the variables of interaction design into computational typography as seen in the following works.

Fig. 6: J. Abbott Miller's proposed seven ways which type can be rendered dimensional, (from left to right): extrusion, rotation, tubing, shadowing, sewing, modular construction and bloating.

Computational typography is not bound to the screen, it can also be used for print. This poster was created using computational methods, by John Maeda, for the mathematics department at MIT.

4 - COMPUTATIONAL TYPOGRAPHY

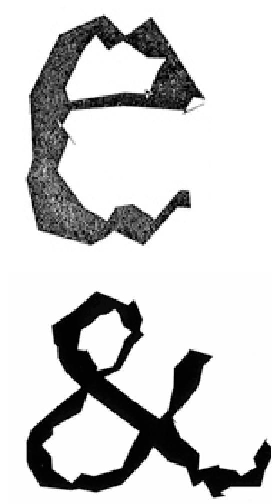


Fig. 7 & Fig. 8: FontFont Beowolf by Letterror, was the first font to take advantage of the randomisation function.

4.1 FONTFONT BEOWOLF BY LETTERROR

Letterpress prints have an intrinsic aesthetic value which is hard to reproduce with modern production methods. Their value as unique physical objects comes from the tiny imperfections on the paper or the ink which are pleasing to the eye. These imperfections do not devalue the final product but make the print special and gives it personality. Computers and digital media due to their precise nature are clean and flawless. How can beautiful imperfections such as letterpress errors be reproduced on the computer without looking overdone or faked?

In 1989 the Dutch type foundry Letterror, founded by Erik van Blokland and Just van Rossum discovered a way which they could simulate printing errors without direct manipulation. Using PostScript to modify their letter-forms, Beowolf was the first font with a built-in randomisation function. This randomisation mechanism works inside the printer allowing the vector points, which give the font its shape, to freely move allowing the font to be flexible. The resulting printout is composed of characters that are unique to each print.

The font Beowolf is an early example of how traditional errors in typography, such as the ones produced by letterpresses, are embraced by computational methods and given new life. The uniqueness in appearance which was valued in traditional techniques was transposed into a modern printing method, allowing computational typography to draw connections to the hundreds of years of aesthetic tradition.

4.2 BIOMORPHIC TYPE BY DIANE GROMALA

One of the primary advantages of having type on screen is the ability to continuously edit and change the appearance of the type, something impossible to do on printed matter. Unlike Beowolf, changes and morphs don't have to be based on random functions but can actually happen as a response to a stimulus (Bolter & Gromala, 2003). Biomorphic Type, such as the typeface Excretia, by Diane Gromala is a continuously morphing type that reacts to the user's physical states measured by a biofeedback device.

The biofeedback device hooked up to the writer measures heart rate, respiration, viscosity of the eye, brain activity, muscle activity and galvanic skin response. The data measured by the biofeedback device is then translated into numbers which control PostScript parameters changing the shape of individual characters.

Gromala's Biomorphic Type allows us to rethink the purpose of typography. In general, typography has been a "window onto the world described by language" (Bolter & Gromala, 2003), which ideally should be transparent as a crystal goblet, as argued by Beatrice Warde. Gromala with Excretia tries to promote a different type of typography in which the purpose is to be noticed by the reader and appreciated (Bolter & Gromala, 2003). It draws from traditions such as Japanese and Islamic calligraphy as well as medieval Illuminated manuscripts where style and image are as important as their context. The physicality of Gromala's typeface draws from calligraphic traditions but its computational qualities allow it to change the static qualities of type and transform the text into a living object which interacts with the writer and the reader.

4.3 LETTERSCAPES BY PETER CHO

Interactivity can be achieved with several different types of devices. Letterscapes, by Peter Cho, explores how users can interact with type using one of the most common interface devices: the mouse. Cho's series of 26 experiments (one for each letter of the Latin alphabet) are interactive games which manipulate several of the interaction design variables, such as motion, space, appearance and texture.

Peter Cho's Letterscapes presents a creative collection of type manipulation. Cho's work stimulates our visual sense by providing entertaining eye candy created from simple typographic forms. It successfully takes traditional forms and applies the power of computation allowing type to express itself beyond form. Several of the experiments presented in Letterscapes make snapshot statements about colour theory, perspective, psychology, physics and even biology.

Although Letterscapes might seem repetitive after exploring it for some time, it is a great introduction to what can be accomplished with type and programming. It successfully stimulated my motivation to reveal, by merging typography and interaction design, concealed meanings and context which might be hidden inside static letters, enhancing and augmenting their communication power.

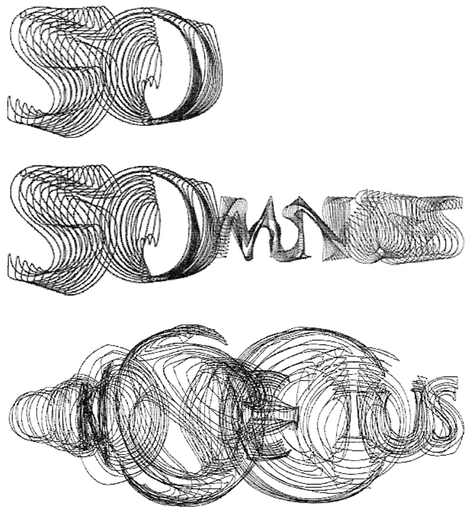


Fig. 9: Diane Gromala's biomorphic type Excretia, distorted by bio-input recorded from the writer.

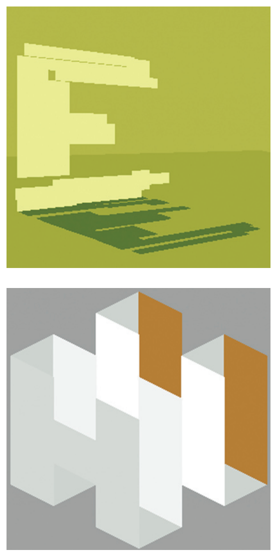


Fig. 10 & Fig. 11: Computational manipulation of letterforms by Peter Cho by his Internet piece titled Letterscapes.

4.4 TEXTARC BY W. BRADFORD PALEY

The computer's power, coupled with the Internet, allows users to analyse and draw information from a seemingly infinite pool of data. The Internet is an enormous database of information which is connected together through hypertext. The invention of hypertext augmented the power of type on screen and allowed users to click text to reveal data relevant to the word being selected. TextArc, by W. Bradford Paley, presents a innovative way to visualise and analyse huge amounts of text data by revealing connections.

TextArc uses the text of a novel such as Alice in Wonderland, and presents the whole text in its entirety on a single computer screen. In fact, the text is presented twice as a series of ellipses, the outer ellipse presents the text line-by-line, and the inner ellipse presents the text word-by-word. Frequently used words such as, Alice in the case of Alice and Wonderland, stand out from the background indicting greater use in the text. Additionally, words that appear more than once are drawn at their average position, relative to the outer word-by-word spiral. W. Bradford Paley clarifies the TextArc mechanism with the following metaphor:

> "Imagine each word attached to where it belongs around the spiral by a tiny rubber band; if the word appears in two places two rubber bands are attached. The net result of this rubber band tug-of-war is that a word will appear closer to places where it is used more. Hence gryphon and caterpillar draw close to their chapters while rabbit stays more central. TextArc was designed with exactly this intent: words draw attention to where they appear in the document."

(Bradford Paley, 2001).

In addition to the features mentioned, the program also "reads" the text by connecting the words on the screen in the order which they appear, using a coloured curved line. This allows the reader to follow the narrative and visualise the word connections simultaneously.

TextArc presents a new method to read and analyse a story, by outlining word frequency and word connections. It removes some of the linearity and static qualities of a book and brings in qualities of hypertext, such as flexibility and the ability to explore related data. TextArc augments the capabilities of the reader, allowing various levels of filtering through different styles of text visualisation.

Fig. 12: A bright yellow line connects words in the TextArc version of the novel Alice in Wonderland. TextArc provides the user an innovative way of reading and visualising texts.

Final version of Randvetica. This simple program allows the user to manipulate the font's randomness in real-time on screen. It produces effects similar to Letterror's font Beowolf.

5 - EXPERIMENTS

Having surveyed a large body of works which manipulate the building blocks of interaction design to create successful works of computational typography, my next step was to create my own computational typography experiments. My experiments started as simple applets manipulating a single variable, and evolved into more complex programs addressing several variables and methods of interaction.

5.1 PROCESSING 0125 BETA

I used Processing 0125 Beta as my primary tool to create computational typography applets. Processing is an increasingly popular open-source programming language developed by Ben Fry and Casey Reas, former graduate students at MIT's Media Lab Aesthetics and Computation Group. Processing is targeted at students, artists, designers, researchers and hobbyists who want to program images, animation and interaction.

Processing is relatively easy to learn. There are several tutorial books which have been published and programming basics are fully documented online. Additionally, there is a strong and active Processing online community helping with and collaborating on each others projects. Having little knowledge of programming prior to this MA course, I was able throughout the year to learn how to code with Processing by reading tutorials and engaging in discussions with many members of the Processing community. My experience is proof that Processing can be a powerful tool used by anyone who dedicates some time in learning its basics.

Fig. 13: Processing is a powerful yet easy to learn open-source programming language from designers.

5.2 RANDOMISATION FUNCTION

I started my experiments by looking at randomisation functions, similar to the ones used by Letterror for their Beowolf typeface. I created two small programs using Processing's "random()" command, the first without motion and the second with motion. The first applet displayed 10,000 static letter "As" with random position, colour, and size. The second applet displayed 200 randomly coloured "As" which moved haphazardly across the screen.

The purpose of these two simple experiments was to explore the power of computation and randomisation of the computer. Creating these visuals is extremely simple when automated by the use of code. On the other hand, if the same experiment was done manually or animated frame-by-frame, it would have taken ages to complete.

Unexpectedly, I also found that when applied to position the randomisation function simulates an organic feel. This is especially true on the 10,000 "As" experiment, where the random patterns created by the "As" against the negative space emulate the patterns created by moss on a rock.

5.3 MOVEMENT IN SPACE

The second set of experiments looked into interactions between type and space. I did a series of four simple applets which explored different types of movement. The first two applets explored movement on the x-axis, the third on the x and y-axis and the fourth explored movement on the z-axis.

These experiments were fairly basic and would later be used as stepping stones to create more complex applications. Their purpose was mainly to learn how to accomplish motion in all three axes with-in the Processing programming language. The most successful of the quartet was the applet manipulating the z-axis. Due to the added dimension, it created a sense of perspective and a feeling of virtual 3d space, which was attractive to the eye.

5.4 REACTION TO SOUND

The third experiment used the lessons learned while manipulating the z-axis, but also added the variable of sound to the interaction. The applet consisted of a 3-dimensional 'F' which reacted to sound volume. As the volume grew louder the letter F's size increased, and if the volume decreased so did the letters' size.

It was entertaining to see the letter react to sound because reaction to the environment is usually a characteristic of living organisms. The letter F seemed to dance in the screen when music was played, or get startled when it heard a loud noise. Changes to the appearance of the type, caused by reactions to stimuli added an extra layer of personality to the letterform. The level of personification obtained, was similar to what Diane Gromala achieved with her Biomorphic Type.

Fig. 14 - 16: 10,000 As (top), Movement in the Z-axis (middle), Dancing F (bottom).

5.5 REACTION TO TILT SWITCH

From the start of this project I wanted to break free from purely screen based interactions, and connect the virtual world of the computer screen with the real physical 3d world. In order to do this I did not want to rely on the conventional methods of interaction such as the mouse and keyboard. In order to create new physical objects to interact with type on screen I relied on the use of sensors and switches connected through a mini computer called Arduino. The Arduino mini computer board is part of a third-party initiative extending the Processing environment to incorporate electronics into the programming.

At this stage of the project I was envisioning that my final piece might involve creating a series of children's toys which aid in learning the alphabet via the use of interactive toys. Therefore, as my fourth experiment I decided to make an interactive equivalent of Alphabet Blocks. I made a prototype using a tilt sensor which changed a letter A displayed on screen from capitalised to lowercase when the tilt sensor was tilted.

I was satisfied by the ability to manipulate type on screen using unconventional devices but after a series of tutorials I became less convinced by the idea of creating children's toys. I veered away from children's toys because I was unfamiliar with my target audience and it would require lots of extra research which was not relevant to my project.

5.6 WII REMOTE INTERFACE

I decided to continue exploring alternative interfaces that could be used to interact with and manipulate type on screen as my fifth experiment. The device I choose to implement was the Wii remote. The Wii remote is a revolutionary wireless device which was first introduced on 2006 with the release of Nintendo's 7th generation video game console. The Wii remote's main feature is its motion sensing capabilities, made possible by accelerometer and optical sensor technology. Ever since the Wii's release many have "hacked" the remote allowing it to interface with other non-Wii devices and programs, such as Processing.

The applet I created uses the Wii remote to steer a letter H around 2-dimensional space, similar to the 3rd motion experiment (section 5.3). In addition, multiple buttons are used to change different characteristics of the letter's appearance, such as scale, colour and shape randomness. The applet also takes advantage of the Wii remote's vibration feedback system causing the remote to vibrate whenever the letter H touches the rectangle positioned in the middle of the screen.

This experiment was successful because it transformed the letter H into an entertaining shape to play with, due to the method of interaction. Using the Wii remote is substantially more entertaining than other input devices, such as the mouse, because the sensation of having your wrist movements translated into an object on screen is satisfying and involving.

Fig. 17 & Fig. 18: Arduino board (top), The Wii Remote (bottom).

5.7 RANDVETICA

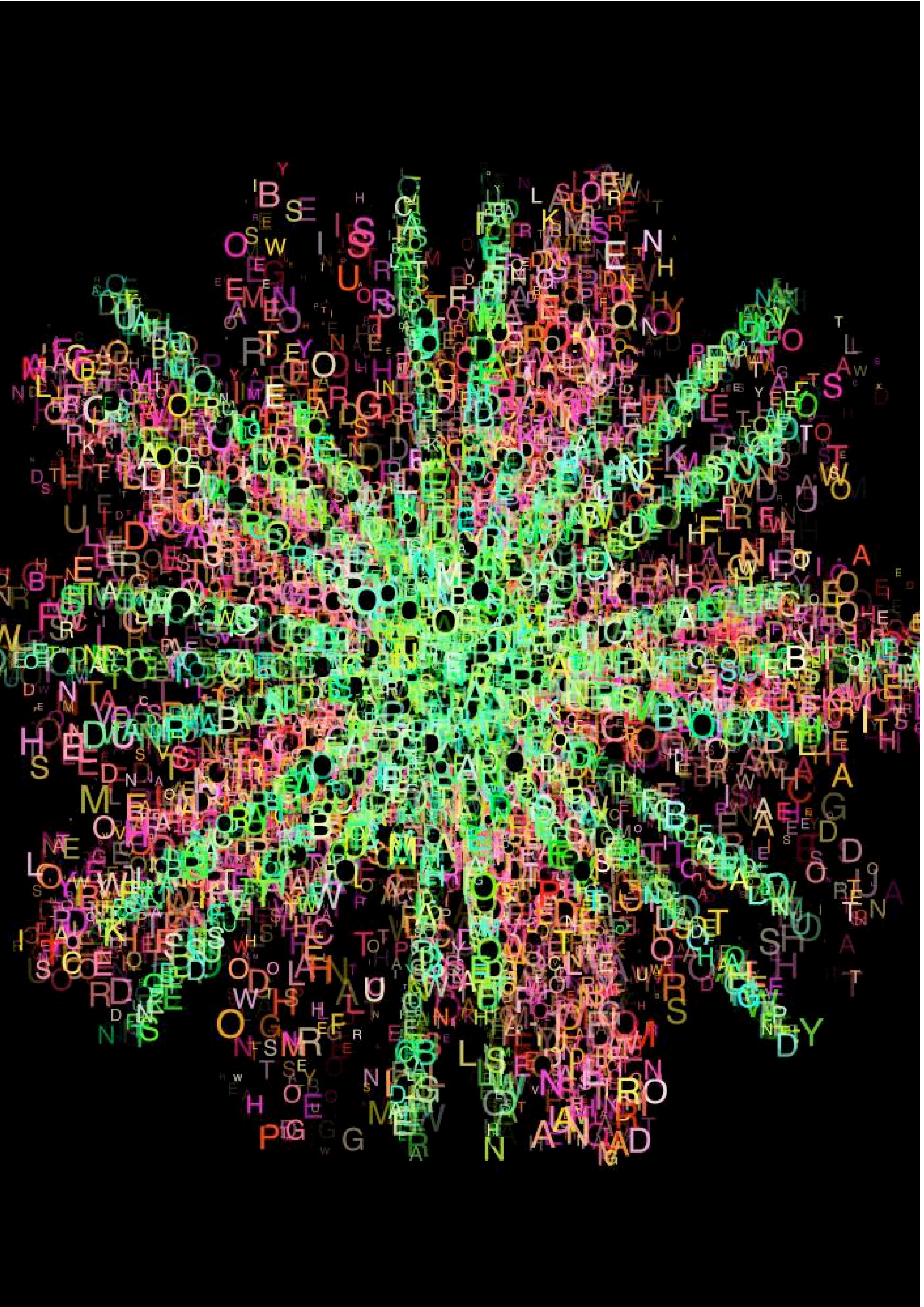
As the sixth and final piece in this series of quick experiments, I decided to revisit randomisation and continue to play with the idea introduced by Letterror's Beowolf typeface.

Randvetica is a simple applet where the user can change the degree of letter shape randomisation on a typeface based on Helvetica. Unlike Beowolf, randomisation does not occur inside the printer, instead the letterforms in Randvetica are in constant flux on the screen. The letters which are in constant mutation can be adjusted in size and randomness strength using two sliders at the bottom of the screen. In addition, when the text is erased using the backspace key, the letters leave an after image of their position. This feature emulates the smudge effect left by words written on chalk blackboards.

Randvetica was my first serious attempt in creating an applet with a specific target audience. I was aiming to create a typeface that could exclusively be used for interaction design, computational artwork or animation. Randvetica, because it is in constant random flux has an organic quality similar to an amoeba. Therefore this typeface could be used by designers who need a typeface which has organic qualities and unpredictable form.

Although I thought it was relatively successful, Randvetica did not meet my goal of achieving a greater level of interaction between the virtual and the physical world as explored previously with alternative input devices such as the Arduino board or the Wii remote. In addition, Randvetica only manipulated form, and lacked any context which interactive design principles could be used to change or enhance communication. The following section outlines my efforts toward choosing a topic of which context could benefit from interaction design and computational typography.

Fig. 19: Quote by Alan Kay, 1971. Text rendered using Randvetica.



Typo Flowers uses the full text of a novel, such as Alice in Wonderland to create flower patterns. The patterns are created using a computational method based on mathematical Rose equations.

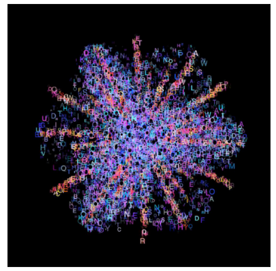
6 - TYPO FLOWERS

Of all the works which I surveyed during the research phase, I found TextArc by W. Bradford Paley to be the most innovative and communicative. TextArc provided users with a new method of reading, which doesn't necessarily have to be linear. In addition, TextArc presented, through a simple glance, several patterns of the novel, many which were not visible in its original printed book form.

Using TextArc as my starting-point, I set to explore other types of information and patterns which could be extracted by presenting a novels' text in its entirety on screen. To achieve this, I wanted to find a new way to visualise text in a manner which was attractive but also achieved a high degree of difference between iterations. While reading a National Geographic Magazine I stumbled across an article which discussed flower taxonomy. In the article the author describes the methods which researchers utilise in order to catalogue flowers based on their physical attributes. This article sparked the idea to make a visual cataloguing system for books utilising an image created using the books' whole text. I decided to emulate the flower system by transforming a novels' text into a computational flower based on Rose equations.

Rose equations are simple sinusoid equations which are plotted in polar coordinates, to create patterns which resemble flower petals. My goal in this experiment, which I named Typo Flowers, was to assign parameters such as, petal count, to code for a wide variety of genres based on the Dewey decimal system. Although this seemed like a plausible idea, the flowers which were created by the Rose equations did not differ enough from each other to achieve an adequate level of complexity as to catalogue books.

I abandoned the idea of cataloguing books and instead decided to focus on a system which could be visualised but also manipulated in order to create forms with meaning. Throughout this project I found that the most interesting experiments were the ones which forms emulated organic systems. Having studied biology and chemistry as part of my undergraduate degree, I decided to explore how others have visualised the genetic code. The genetic code was relevant to this project mainly because it uses a typographical system based on the letters A, C, G and T, therefore presenting a system which could be enhanced with computational typography.



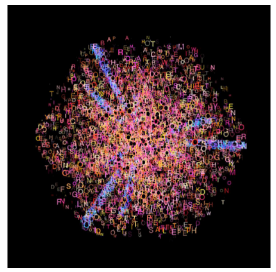


Fig. 20 & Fig. 21: Alternate versions of flowers created using Typo Flowers.

This image represents a segment of chromosome 21. The darker areas stand for coding regions called exons, while lighter areas stand for non-coding areas called introns.

7 - GENETIC WORK REVIEW

Due to the scientific nature of the topic I was exploring, I decided to limit the works I would sample to ones which focused on visualisation methods rather than raw genetic data, in order to avoid dwelling on purely scientific studies

The works which ultimately were most relevant to my exploration were the genomic cartography experiments by Ben Fry. Designer Ben Fry has worked extensively with different ways to visually represent the human genome, which is composed of 3.1 billion characters. Ben Fry's work approaches each of the 21 human chromosomes with a distinct visual treatment. While some of his experiments are just static printouts of the genomic sequence identifying introns (non-coding segments) and exons (segments coding for genes), others are complex interactive pieces displaying differences in haplotypes (small differences between the genetic codes of each individual). Fry's experiments were inspiring, and pushed me to apply what I had learned from my previous experiments into genetics.

Before I could start my own experiments I decided to narrow the complexity of the genetic codes I would be manipulating, by choosing to focus on viral genomes. Ben Fry in his work tackled an immense amount of data because he was analysing human chromosomes which are several million characters in length. On the other hand, the genetic codes of viruses are more manageable with lengths starting at only a few thousand characters.

I quickly surveyed existing methods of visualising viruses, but most were highly complex scientific systems, which required intricate equations and advanced hardware to render usable images. In addition, I looked at existing taxonomical systems for viruses, and found that the graphics used by researchers and educators are fairly basic and non-interactive, such as the Virosphere diagram.

Having surveyed the field, I identified an opportunity to create a visualisation system for viruses, ideally to be used by educators and targeted mainly at biology students (ages 15-18). This system would involve visualising genetic and structural viral information using principles of interaction design and computational typography.

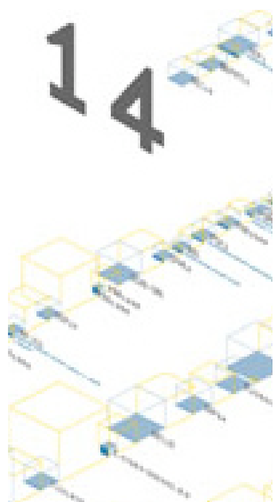
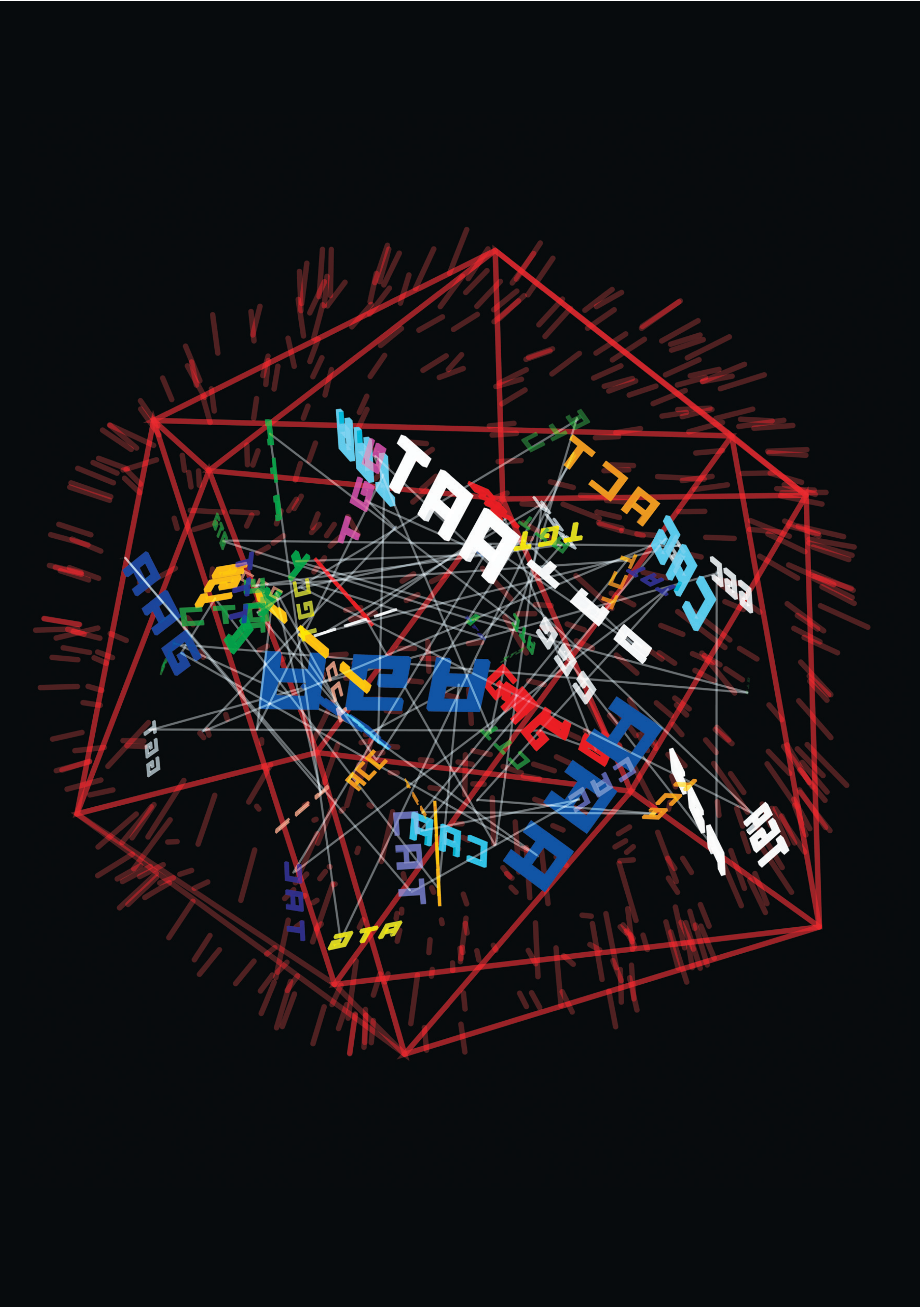


Fig. 22 & Fig 23: Haplotypes visualisation (top), Chromosome 14 visualisation (bottom).



Visualisation of the HIV virus using Viral Blocks. The letters, colours, and geometric patters stand for genomic and structural viral information.

8 - VIRAL BLOCKS

The first step in creating a visualisation system for the genetic and structural information of viruses was to identify which parts I wanted to include as part of my visualisation. I settled on seven characteristics: morphology, viral envelope presence, viral host, taxonomical group, codons present, codon frequency, and genome size. Having identified which variables I wanted to manipulate I started developing the alpha version of Viral Blocks.

The alpha version of Viral Blocks, named Genomic Flowers, used the same visualisation principles as Typo Flowers because I had become attached to the aesthetic look of the flower shapes produced by the Rose equations. While Typo Flowers used the text of a novel, Genomic Flowers used the genomic sequence of a virus, in the form of a text file, to produce its visuals. While Typo Flowers displayed each character individually, Genomic Flowers differed by displaying characters in trios. The three character combination represented codons, which are the basic grouping of information in the genetic code. There are a total of 64 codon combinations coding for 20 amino acids (basic building blocks of proteins). Because there are more codon combinations than amino acids, some codon combinations code for the same amino acids, which make the genetic code redundant. This redundancy in the genetic code helps organisms prevent copying errors when reading their genetic material. In addition to the central flower shape created by the genetic code characters, I added other visual layers which help visualise other viral structures, summarised on table 8.1.

Table 8.1 Genomic Flowers Structure List

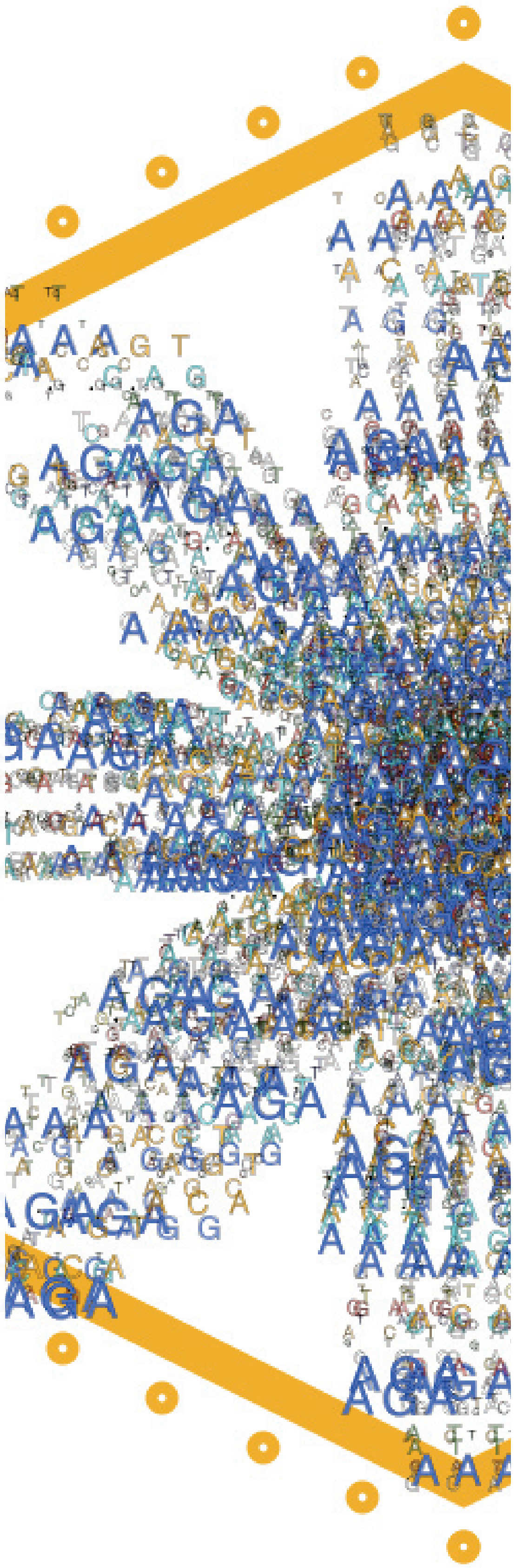
Variable	Visual Representation
Morphology	Outer geometric shape (square, icosahedron or circle)
Viral envelope presence	Circles surrounding the geometric shape (only if envelope is present in a virus)
Host	Color of geometric shape and/or circles representing envelope
Taxonomical group	Petal shape of flower created by genetic information
Codons	Colors representing the 20 amino acids
Codon frequency	Opacity and size of the characters
Genome size	Relative diameter of the flower created

Genomic Flowers was not very successful mainly because it wasn't interactive and engaging. After seeing one or two of these flowers being generated, exploration was over, and there wasn't much else for the user to see or explore. This was mainly because the program lacked any movement, therefore there was no visual interaction taking place. In addition, I felt that using the analogy of flowers into viruses was confusing. It would have made sense if I was using flower genomes but coupling flowers with viruses made the concept unclear.

I decided to start anew, so I could abandon my fixation with the flower shapes. After surveying the Processing website for ideas, I concluded that a successful visualisation would have to include motion and preferably 3d graphics. I had experienced problems in previous experiments dealing with 3d graphics because they required an insane amount of computational power. In order to make the transition into 3d, I had to drop the idea of presenting all the characters of the genetic code on screen simultaneously, because it was impossible for the computer to calculate the movement in 3d space of thousands of characters in real time. When I forced the computer to achieve those calculations the resulting motion was extremely sluggish and very frustrating. Instead I focused on presenting only the 64 codon combinations allowing each trio to be a separate 3d object moving in space. Having overcome that technical hurdle, I also decided not to include taxonomical group as one of the variables in this second version. This decision was largely because that variable relied on flower petal shape, which had been abandoned at the start of the second version. Table 8.2 summarises how each of the six variables are visualised in the second version, renamed Viral Blocks.

Table 8.2 Viral Blocks Structure List

Variable	Visual Representation
Morphology	Outer geometric shape (cube, 3d-icosahedron or cylinder)
Viral envelope presence	Hair-like structures surrounding the geometric shape (only if envelope is present in a virus)
Host	Color of geometric shape and/or hair-like structures representing envelope
Codons	Colors representing the 20 amino acids
Codon frequency	Opacity and size of the 3d-characters
Genome size	Relative diameter of the geometric shape



Satisfied with the visualisation strategy, I turned my efforts to another aspect of interaction which I had been trying to address: alternative input devices. Using the concept that DNA and amino acids are the blue-prints and building blocks of life I wanted to create a physical object which used blocks to manipulate the screen. I was able to achieve this using the Lego Mindstorms NXT programmable brick. The Lego NXT was similar to the Arduino board because it allowed me to connect Lego motors and sensors which interact with Processing. Processing and the NXT did not interface out-of-the-box, therefore I had to use a third-party plug-in developed by Jorge Cardoso, professor at the Portuguese Catholic University at Oporto. Colaborating with Jorge Cardoso was extremely helpful as I was able to identify bugs and suggest features for his plug-in while I used the NXT on the Viral Blocks project. The machine I created using the NXT was a simple scanner which recognises Lego blocks of different heights, labelled with the letters A, C, G and T. The user places these blocks in different combinations in four separate bins, in a similar manner as a typesetter sets type, then the scanner is able to read the arbitrary height combinations. Changing the Lego block combinations changes the parameters of visualisation, essentially mutating the virus. Having worked extensively on making the machine work, it became a substantial part of the project, therefore I decided to adopt a Lego identity for the program. The Lego identity proved to be well suited and was used in more depth on the graphical user interface and tutorial videos.

The GUI and the tutorial videos of Viral Blocks work together to create an educational narrative. The program was designed to prevent users from accessing all its options from the start, therefore users have to watch tutorials and explore each viral structure individually before getting the big picture. This strategy was very successful because it brought about a fun exploratory pace to the program.

The final version of Viral Blocks was very successful, because I was able to accomplish several of my goals such as manipulating interaction design variables, using computational typography and successfully implementing alternative input devices. Additionally looking at the project from an educational standpoint, I believe the program entices users to explore the structures and genetics of a wide variety of viruses, in a fun and informative manner.

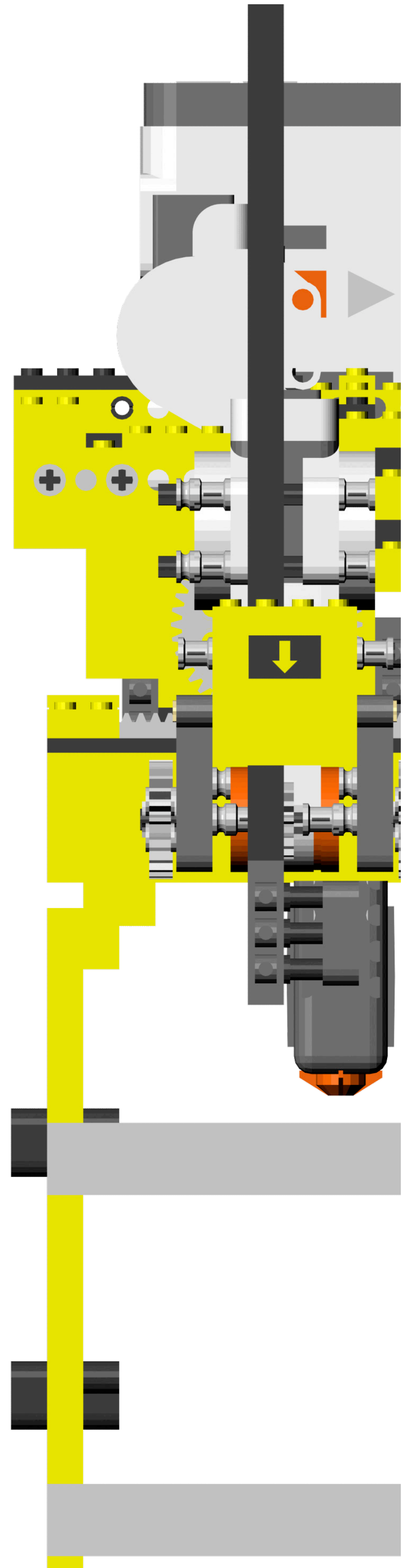


Fig. 24 & Fig. 25: Visualisation of HIV virus using Genomic Flowers (left). Sequencer machine created with the Lego Mindstorms NXT for Viral Blocks (right).

Biological structures, due to their miniscule size and complexity, benefit greatly from visualisation methods. These visualisation methods can be useful in promoting an interest for science and educating future generations of biologists and geneticists.

9 - CONCLUSION

Interaction design, besides providing a new language and methodology for developing better interactions, also promotes an alternative angle for tackling design problems. By analysing the building blocks of interaction design one can effectively identify aspects of an artefact, service or system which can be improved. By asking oneself the fundamental question "Why does it have to be like this?" one gets a new perspective and the artefacts, services or systems which once were obsolete or antiquated can acquire new life. Looking back at the project I feel that I accomplished several of the goals which I originally laid out on my proposal, because I was able to successfully change and enhance the way viral structures and genetics are communicated. In doing so, I took existing systems such as typography and the genetic code, and applied computational methods to their construction in order to enhance and augment them.

Ultimately, I wish I had had more time to test Viral Blocks on my target audience. Having once been a biology student, I can provide some perspective on what could be useful for students, but my evaluation of the success of the project would have been biased. If I had had the adequate contacts, I would have taken the project to classrooms in order to get feedback from educators and students alike. I believe Viral Blocks could be an attractive tool for classrooms or science museums because it uses striking graphics to introduce abstract concepts such as genetics.

As an end note, the Viral Blocks project has been an amazing journey of exploration. The project originally began with my passion for technology and my admiration of the letterform and quickly became a thorough examination of the field of interaction design and computational typography. I hope that by providing the Viral Blocks project online (http://www.diegobaca.co.uk/viralblocks/main.html), others can benefit from my research and be informed of what can be accomplished using computational methods and understanding interaction.

10 - GLOSSARY

Allele: any of the alternative forms of a gene that may occur at a given position on a chromosome.

Amino Acid: a simple organic compound containing a carboxyl (-COOH) and an amino (-NH2) group. Amino acids are the basic components of proteins.

Arduino: a physical computing platform based on a simple input/output board that implements the Processing language.

Codon: a sequence of three nucleotides, which together form a unit of genetic code in a DNA or RNA molecule.

Exon: a segment of a DNA or RNA molecule containing information coding for a protein sequence.

GUI: Graphical User Interface, a way of interacting with a computer using items such as windows, icons, and menus, used by modern operating systems.

Haplotype: a group of alleles of different genes on a single chromosome that are closely enough linked to be inherited, usually as a unit.

Intron: a segment of a DNA or RNA molecule which does not code for a protein sequence and interrupts the sequence of genes.

Lego Mindstorms NXT: programmable robotics kit released by Lego in late July 2006. The main component in the kit is a brick-shaped computer called the NXT brick. It can take input from up to four sensors and control up to three motors, via cables very similar to but incompatible with phone cords.

Nintendo Wii: Nintendo's seventh generation video game system, released in 2006.

Nucleotide: the basic structural unit of nucleic acids such as DNA.

PostScript: is a page description language and programming language used primarily in the electronic and desktop publishing areas.

Processing: an open-souce JAVA based programming language developed by Ben Fry and Casey Reas at the MIT Media Lab.

Protein: a complex molecule composed of one or more chains of amino acids; are an essential part of all living organisms.

Viral Capsid: a protein layer which protects the genetic material of a virus. It determines the viruses' morphology. The majority of viruses have capsids with either a helical, or an icosahedral shape.

Viral Envelope: a coat surrounding the viral capsid, present in some viruses. The primary use of the envelope is help the virus enter host cells.

Virus: a sub-microscopic particle, composed of genetic material, such as DNA, or RNA, surrounded by an outer protective protein layer, called a capsid.

11 - IMAGE REFERENCE

DNA
iStockphoto, file: 3974661, Photographer: ktsimage.

Gutenberg Press
http://personalpublisher.files.wordpress.com/2006/01/Gutenberg-1.jpg

Modern Computerised Press
iStockphoto, file: 2979448, Photographer: narvikk.

Multimedia: 3D Glasses Crowd
http://www.oracleofottawa.org/OOO_multimedia.jpg

Doom Logo
http://www.rome.ro/images/doom/Doom.png

MTV Logo
http://en.wikipedia.org/wiki/Image:MTV_Logo.svg

Vertigo Poster
http://www.unc.edu/~huiliu55/222vertigo.html

Dimensional Typography
Abbott Miller, J. Dimensional Typography: Case Studies on The Shape of Letters in Virtual Environments. Princeton Architectural Press. New York. 1996.

Maedastudio: Math Butterflies
http://www.maedastudio.com/1998/mathbutterflies/index.php?category=static&next=1999/oneline&prev=2005/golfball&this=math_butterflies

Beowolf E
http://www.publiclettering.org.uk/Images/ecimages/02.jpg

Beowolf &
http://www.nwe.ufl.edu/~tharpold/resources/beowolf/

Excretia
Bolter, Jay and Diane Gromala. Windows and Mirrors: Interaction Design, Digital Art, and the Myth of Transparency. The MIT Press. Cambridge. 2003.

Letterscapes
http://www.typotopo.com/letterscapes/

TextArc
http://textarc.org/images/alice4.gif

Processing Logo
http://processing.org

Arduino
http://www.mitchellpage.com.au/research/wp-content/uploads/2006/04/arduino_extreme_480.jpg

Ben Fry Genetics
http://acg.media.mit.edu/people/fry/chromosomes/22/

Haplotypes
http://acg.media.mit.edu/people/fry/haplotypes/isometricblocks.png

Chromosome 14
http://acg.media.mit.edu/people/fry/chromosomes/14/

BioCurious Banner
http://biocurious.com/

12 - BIBLIOGRAPHY

BOOKS

Abbott Miller, J. Dimensional Typography: Case Studies on The Shape of Letters in Virtual Environments. Princeton Architectural Press. New York. 1996.

Baines, Phil and Andre Haslam. Type & Typography. Laurence King Publishing. London. 2005.

Bolter, Jay and Diane Gromala. Windows and Mirrors: Interaction Design, Digital Art, and the Myth of Transparency. The MIT Press. Cambridge. 2003.

Branczyk, Alexander, Jutta Nachtwey, Heike Nehl, Sibylle Schlaich, and Jürgen Siebert. Emotional Digital: A Sourcebook of Contemporary Typographics. Thames & Hudson. 1999.

Dawes, Brendan. Analog in, Digital out: Brendan Dawes on Interaction Design. New Riders. Berkeley. 2007.

Ellison, Andy. The Complete guide to Digital Type: Creative use of Typography in the Digital Arts. Laurence King Publishing. London. 2006.

Greenberg, Ira. Processing: Creative Coding and Computational Art. Friends of ED. Berkley. 2007.

Hefland, Jessica. Screen Essays on Graphic Design, New Media, and Visual Culture. Princeton Architectural Press. New York. 2001.

International Designers Network. IdN Special 04: The art of Experimental Interaction Design. Systems Design Limited. Hong Kong. 2004.

Maeda, John. Creative Code: Aesthetics + Computation. Thames & Hudson. 2004.

Maeda, John. Maeda @ Media. Universe Publishing. 2001.

Maeda, John. The Laws Simplicity. The MIT Press. Cambridge. 2006.

Moggridge, Bill. Designing Interactions. The MIT Press. Cambridge. 2007.

Parker, Lauren. Interplay Interactive Design. V&A Publications. London. 2004.

Paul, Christiane. Digital Art. Thames & Hudson. London. 2003.

Read, Casey and Ben Fry. Processing: A Programming Handbook for Visual Designers and Artists. The MIT Press. Cambridge. 2007.

Saffer. Dan. Designing for Interaction: Creating Smart Applications and Clever Devices. New Riders. Berkeley. 2007.

Sassoon, Rosemary. Computers and Typography 2. Intellect Books. Bristol. 2002.

Woolman, Matt. And Jeff Bellantoni. Moving Type: Designing for Time and Space. RotoVision. East Sussex. 2000.

Woolman, Matt. And Jeff Bellantoni. Type in Motion: Innovaions in Digital Graphics. Thames & Hudson. Lonodn. 1999.

CODE COLLABORATORS

Baca, Mariana. MIT Media Lab. Research Assistant. 2007.

Cardoso , Jorge. http://jorgecardoso.org/processing/NXTComm/. (September, 2007).

Peña, David. http://david.laveneno.org/. Onine. (September, 2007).

DVDS

Verplank, Bill. Interview by Bill Moggridge (Feb. 2001) Designing Interactions DVD. The MIT Press. Cambridge. 2007.

INTERNET

(For a more detailed view of internet sources visit: http://diegobaca.co.uk/Works_Compilation/works.html).

A Digital Remake. http://www.progetto-exp.org/?page_id=97. Online. (August, 2007).

Alphabot. http://processing.org/exhibition/works/004/index_link.html. Online. (August, 2007).

Apartment. http://turbulence.org/Works/apartment/#. Online. (August, 2007).
Biomorphic Type. http://www.lcc.gatech.edu/~gromala/excretia/. Online. (August, 2007).

Brendandawes.com. http://www.brendandawes.com/. Online. (June, 2007).

C71123. http://www.c71123.com/category/typography/. Online. (August, 2007).

Caligraft. http://www.caligraft.com/. Online. (August, 2007).

Calliscope. http://www.transphormetic.com/t4/_calliscope03.htm. Online. (August, 2007).

Chronotext. http://www.chronotext.org/. Online. (June, 2007).

Daka-daka. http://www.flong.com/dakadaka/index.html. Online. (August, 2007).

Dryade. http://igor.grafitron.com/dryade/index.html. Online. (August, 2007).

Dynamic Typography Sketches. http://www.aalab.net/projects/ActiveText/DynTyp/. (August, 2007).

dynTypo. http://www.dyntypo.com/. Online (June, 2007).

FontDont- Beowolf. http://www.letterror.com/foundry/beowolf/. Online. (August, 2007).

Leogeo. http://www.leogeo.com/. Online. (June, 2007).

Letterscapes. http://www.typotopo.com/letterscapes/. Online. (June, 2007).

Maeda Studio. http://www.maedastudio.com/index.php. Online. (June, 2007).

Messa di voce. http://tmema.org/messa/. Online. (August, 2007).

Nutexts. http://typotopo.com/projects.php?id=nutexts. Online. (August, 2007).

One Hundred Million Poems. http://www.uncontrol.com/deploy/content/EXPERIMENTS/massin/massin_big.html. Online. (August, 2007).

Ortho-type. http://www.ortho-type.it/beta033/index.html.
Pcho.net. http://www.pcho.net/. Online. (June, 2007).

Processing. http://processing.org/. Online. (June, 2007).

quiresiste.com. http://quiresiste.com. Online. (June, 2007).

Responsive Type. http://www.responsivetype.com/index.php. Online. (August, 2007).

RoboType. http://robotype.net/. Online. (June, 2007).

State of the Union. http://stateoftheunion.onetwothree.net/index.shtml. Online. (August, 2007).

Tagcloud. http://www.metaportaldermedienpolemik.net/blog/Blog/2007-08-10/text-comparison-using-relational-tagclouds. Online. (August, 2007).

Takeluma. http://www.pcho.net/takeluma/index.htm. Online. (August, 2007).

Text Arc. http://textarc.org/. Online. (August, 2007).

The Alphabet Zoo. http://typotopo.com/projects.php?id=zoo. Online. (August, 2007).

The Legible City. http://www.jeffrey-shaw.net/html_main/frameset-works.php3. Online. (August, 2007).

Type is Art. http://www.typeisart.com/. Online. (June, 2007).

Type me, Type me not. http://acg.media.mit.edu/people/pcho/portfolio/typemenot.html. Online. (June, 2007).

Typophile. http://typophile.com/. Online. (June, 2007).

Typotheque. http://www.typotheque.com/site/index.php. Online. (June, 2007).

Universe. http://universe.daylife.com/index.html. Online. (August, 2007).

Virus Taxonomy Online. http://www.virustaxonomyonline.com/. Online. (August, 2007).

Voroni Type. http://typotopo.com/projects.php?id=voronoi. Online. (August, 2007).

MAGAZINES

Computer Arts Projects. Interactive Design. Issue 93. January, 2007.

Quammen, David. A Passion for Order. National Geographic Magazine. June, 2007.

THESES & PAPERS

Bradford Paley, W. TextArc: Revealing Word Associations, Distribution and Frequency. Digital Image Design Incorporated. 2001.

Cho, Peter. Computational Models for Expressive Dimensional Typography. Massachusetts Institute of Technology. Media Arts and Sciences. 1999.

Ditterline, Patrick, Blacke Engle and Brian Yeung. The Effects of Kinetic Typography on Redability. Carnegie Mellon University. 2000.

Engelbart, Doug C. Augmenting Human Intellect: A Conceptual Framework. Stanford Research Institute. 1962.

Fry, Benjamin. Computational Information Design. Massachusetts Institute of Technology. Media Arts and Sciences. 2004.

Fry, Benjamin. Genomic Cartography. Massachusetts Institute of Technology. Media Arts and Sciences. 2002.

Fry, Benjamin. Organic Information Design. Massachusetts Institute of Technology. Media Arts and Sciences. 2000.

Lewis, Jason E. Dynamic Poetry: Introductory Remarks to a Digital Medium. Royal College of Art. 1996.

Small, David L. Rethinking the Book. Massachusetts Institute of Technology. Media Arts and Sciences. 1999.

Wong. Yin Yin. Temporal Typography: Characterization of time-varying typographic forms. Massachusetts Institute of Technology. Media Arts and Sciences. 1995.