Declaration

I declare that the work described in this research Paper is, except where otherwise stated, entirely my own work and has not been submitted as an exercise for a degree at this or any other university.

Signed: _____

Aishling Erin Costello

26th February 2014

Permission to lend and/or copy

I agree that Trinity College Library may lend or copy this research Paper upon request.

Signed: _____

Aishling Erin Costello

26th February 2014

Disclaimers

The information contained in this document is intended to provide a guide to those seeking admission to the
programme, and to the students on the course. Trinity College Dublin reserves the right to update or change
syllabi, timetables, or other aspects of the programme at any time. Changes will be notified to current
students by email.

Introduction

> *"Code is the 21st century literacy and the need for people to speak the ABC of Programming is imminent. Our world is increasingly run by software and we need more diversity in the people who are building it."* [Liukas, 2014a]

Initially, this research paper was interested in exploring Irish public design in the form of digital services, systems and products. Its core intention was to explore what digital Irish design is in style, execution and how it fairs against world-wide standards.

As the case studies were delved into more and more, it was evident that Ireland was at times, suffering from accessibility issues, as well as poor design execution and dilapidation of the long-term implementation of design products in public spaces.

Inefficient real-time bus stop signs throughout Dublin, the abandonment of a nationwide way-finding system for heritage sites, the embarrassment that was the Irish Revenue Tax website of 2012 – 2013; were but a few of the examples of ubiquitous Irish design that is failing, but no-one seems to know why.

Eventually, as the public's apathy towards solving any of these problems sank in, it became evident that the only way anyone can propose solutions for the 'bad design' one encounters, is to look at the root cause.

Design has a deep and rich history in Irish culture, in the form of crafts such as fashion and textiles, ceramics, glass, print making, book making, millinery, publishing, letter press. We are a nation known for producing internationally renowned products that are coveted the world over. Galway crystal is popular in America, just as Orla Kiely fashion is in Japan. We, as an Irish people, know what 'good design' when it comes to craft. So why do we accept digital design that is found wanting as national standard? Should there be a contemporaneous zeitgeist for higher expectations of digital design in Ireland today?

Ireland has experienced many reformations in design culture over the centuries, and it is evident that we are facing one now. Post 2007 recession, we are turning into a culture of hackers, DIY-ers and self-producers. Colloquial online resources for voicing opinions about 'the state of the nation' are becoming more and more popular.

Broadsheet.ie[1] is an exemplary standard of the average outspoken Irish person. Commenting on current affairs at home and abroad, the site offers a safe public space to say what you feel is right or wrong about culture.

We as a nation are finally speaking out and acting to change political agendas. We are seeing major reform in the rights of women, fathers and homosexuals. 2014 has already encountered the Ms Panti and RTE Scandal and the IONA Institute's response and *Operation Transformation* has kicked off as a nationwide reform to beat back obesity. Post 2007 Ireland is a changing social landscape, and public services that cater to these changes should evolve in parallel with said landscape.

It is therefore evident that it is an imperative to look at the root cause of the failures in Irish digital design. Treating the symptoms, i.e. the already existing faulty design, is important but it will not prevent future problems. Therefore, we must facilitate a change in the designer's process, ethics, skillset and culture, in order to produce lasting products and systems that, rather than crumble and decay in the world, adapt and change it for the better.

The overarching question driving the methodological approach and content of material of this thesis, is whether or not it is important to become a computationally literate society, and what changes that will impose on the future of humanity.

This question proposes the research of computer science education, design and basic literacy skill of children. It is hoped that by focusing in on these topics, that a set of milestones for building a new pedagogical approach that combines the fields of computer science, software studes, computational logic, literacy, primary level education, design theory and practise will become evident.

Situating these milestones in today's reality, as well as a speculative system to put in place in schools is also important, as it will help to flesh out the theory of the established milestones.

The three main examples of learning techniques that this thesis will focus on include motivation, 'the flow' theory, and literary features such as metaphor and simile. They will be investigated from a theoretical approach as well as located and situated within existing practical applications.

[1] Broadsheet is available at www.broadsheet.ie

LITERATURE REVIEW

Lev Manovich published his long-anticipated book on the fundamentals of the role of computers in society, *Software Takes Command* in 2013. In this publication, researchers and enthusiasts who are interested in the study of digital humanity are confronted by a wealth of information pertaining to the role software plays throughout all strata of society.

Manovich advocates the promotion of computational logic in education and practice across a range of digital based professions, that would otherwise not be traditionally familiar with computer-based logic and language. His argument for a mass immersion of persons into educating themselves about how computers and technology function, let alone become proficient at using these various forms of technologies, begins with the grandiouse statement:

> *"Software has become our interface to the world, to others, to our memory and our imagination – a universal language which, the world speaks, and a universal engine on which the world runs."*
> [Manovich, 2013, pg 2]

Such a strident opening prepares the reader for the onslaught of well prepared and researched arguments, as to why humanity should take more care to better educate themselves in the subject *software studies*; formally known as *media studies*.

This publication is the pivotal source of foundational information which stabilise the the overall question proposed by this research paper; *can humanity benefit from a design generation that is computationally as well as computer literate?*

With this angle in mind, creating a bibliography around the question of 'how' this statement would come to fruition became a priority.

Locating a source of criticism that counterbalances Manovich's enthusiasm on the topic, came forth in the form of Victor Papanek's 1988 classic paper appropriately titled, *"The Future Isn't What It Used to Be".*

In his opening introduction, Papanek criticises the approach designers of the time are taking to the adaptation of the design process, to the new digital age arising out of the 1980s. He argues that by systametising the design process to become more scientific,

to better suit digital technologies, designers are creating ill-fitting products within society. He states this quite clearly;

> *"…their approach stands for reason, logic, and intellect, but such a method leads to reductionism and frequently results in sterility and the sort of high-tech functionalism that disregards human psychic needs at the expense of clarity."* [Papanek, 1988, pg 4]

It is therefore important to recognise that technology is merely a tool for the designer, and while it is in society's best interest to become computer literate – which means to know the fundamentals of how hardware and software function, as well as computational logic, language and syntax and the history of technologies – it is an imperative to never forget that technology exists to mediate information from user to user, not act as a source for this information.

Jeannette Wing agrees with Victor Papanek's enthusiasm for humanity to not allow the *"tool to get in the way".* [Wing, 2008, pg 3721] Here *"the tool*" refers to the technology at hand, the hardware and accompanying software, be that on a personal computer, laptop, tablet or smart phone. *"In the way"* refers to her argument, which is within the context of creating a new pedagogical approach to computational logic in formal primary school level education.

By adapting Papanek's argument, Wing has evolved the original problem proposed by him, into a challenge that is helping to shape a system to teach computer literacy en masse to future generations from a young age.

Her research paper both agrees and disagrees with Papanek and Manovich on this topic. By triangulating these core texts, the overarching question becomes fleshed out and ready to be dissected within specific spheres of humanity; early literacy education in children, the education and industry of graphic design, and how computational literacy affects society as a whole. As she eloquently phrases her reasoning as to *why* computational pedagogy needs reform in the school system;

> *"If computational thinking will be used everywhere, then it will touch everyone directly or indirectly."* [Wing, 2008, pg 3720]

And that this focus of direction will be a motivational driving force for future generations of society;

> *"the next generation will have to be able to think in order to succeed in modern society."*
> [Wing, 2008, pg 3720]

The core text for the first chapter of this thesis, bridges a gap between design and science. The graphic designer knows the fundamentals of typography, and how best to create legability and readability for the user. Science, in this instance specifically neuroscience, breaks down the current understand of the neuroscientifical approach to cognitive skills used when reading text.
Reading in the Brain is a *"highly accessibly description of the brain circuitry of work behind reading"*. [Dehaene, 2014] Dehaene presentes us with a comprehendible and comprehensive breakdown of both the cognitive and neurological processes involved in language acquisition, interpretation and pedagogy.

Dehaene, in his chapter on *learning to read*, invites the theory of linguist and neuroscientist Uta Firth when explaining the psychological breakdown of the three stages a child encounteres when learning to read.

Firth's three stages create a structure on which to base 'milestones' in a new pedagogical approach to teaching computational skills to children and adults.

These milestones spring forth from a variety of sources. Wing directly references a specific set of milestones in her approach to building a new pedagogy. John Naughton, a journalist writing for *The Guardian Online*, makes reference to a more generalised set of milestones for this new pedagogy. By combining their efforts, a new system of teaching is getting closer to the foreground of becoming implemented within education systems across the globe.

Due to the context of the main argument of this thesis esixting in a digital realm, it would be folly not to search for theorists and researchers who are publishing their work in the form of online materials; be they individual blog sites [an example being the blog

of Adrian Short[2]], corportation's offshoot blogs [such as *The Guardian Online's, Developer Blog.* A site dedicates to reporting current affairs in the realm of global developers[3]], specialists within their specific field [Uta Frith's personal site[4]] and websites dedicated to the archiving and reporting of digital humanities and technology [*WIRED* website is an excellent source of information in this area[5]].

Methodology

The type of research conducted for the entirety of this dissertation is in the form of a pragmatic approach to the analysis of qualitative texts. This approach to research was favoured over a biased qualitative or quantitative approach due to the subject matter of the thesis as well as the types of information available about the topics discussed.

The research began with analysing the information in the book *Reading in the Brain* by Stanislave Dehaene, with the intention of better understanding how people translate the semiotics of visual language into meaning and why people with learning difficulties have difficulty with this process.

Originally, the main focus for this thesis was to investigate whether or not an understanding of the technology in use for design would help the designer avoid accessibility issues.

Reading *Software Takes Command* by Lev Manovich heavily influenced the aforementioned original argument and altered it to be less concerned with 'bad design' products and instead focus on the root cause, the designers themselves.

At this point in the early stages of research, Manovich's absolute dedication for everyone to become computationally competent was in absolute agreement with the main argument. Discovering the texts of Papanek and Wing gradually shifted the angle from absolute agreement to hypothetical absolution on the topic. That is to say, that there is certainly leeway when it comes to demanding an entire generation of humanity to be computer programmers, technology literature AND not require these skills on a daily basis.

It was when reading Manovich that the angle of looking at the two topics of 'design' and 'programming' began to shift. The angle quite dramatically changed from

[2]Adrian Short's site is available at http://adrianshort.org/2014/02/06/year-of-code-neoliberal-agenda/
[3] The *Guardian Developer Blog* is available at http://www.theguardian.com/info/developer-blog
[4] Uta Frith's personal site is available at https://sites.google.com/site/utafrith/
[5] *WIRED* is available at http://www.wired.com/

looking at programming as a problem designers need to wrestle with in order to create accessible design for a wide spectrum of users, to teaming ‘design’ with ‘programming’ to possibly eliminate inaccessible design in the future. ‘Design’ and ‘programming’ began to meld into a single discipline, and investigating this seemed more integral to answering the original question; *why is bad design, bad?*

Hunting and locating case studies that encompass the changes outlined by Uta Firth in her stages of reading, helped ground the theoretical framework she developed for literacy pedagogy in the venn diagram of ‘design’, ‘education’ and ‘programming’. The case-studies of children encountering technology during the Logographic, Phonological and Orthographic stages outline the growing ubiquitous nature of technology, and its impacts on these stages of early learning development.

It seemed only natural to triangulate the cited information presented by Dehaene and Manovich in their publications, with first-hand case studies and examples of the kinds of theoretical approaches they mention.

Investigating blog sites that specialise in writing about technology affects culture, the economy and politics. WIRED magazine is a thorough source of information pertaining to technology. They claim to be the ‘first word’ in technology reporting, and their claim is backed by the awards they have won for their reportage. [WIRED 2014]

Thinking of the impacts technology is having on children today, and the initiatives being put into place to teach them how to learn computational skills is creating an interesting speculation of future ubiquitous technologies. Looking to science fiction in the forms of film, television and the novel, it is interesting to speculate as to the possibilities of what the future might bring in terms of computational ubiquitous standards of living for the future.

Chapter One

Literacy Education from Early Childhood to Adulthood

Introduction

"I wish you to gasp not only at what you read
but at the miracle of its being readable.

Valdamir Nabokov, Pale Fire"

[Dehaene, 2009, pg 196]

It is with wonder that we watch an infant so readily and quickly learn in turn to recognise, speak, read and write their first language.

It is important to analyse the development of language acquisition so that one can better understand the breakdown of how humans become literate. Understanding this process will help with the construction of a new pedagogy to introduce computational theory and computer programming skills to future generations.

Throughout the coming chapter, we will investigate the early development stages of language acquisition according to Uta Frith's three stages. We will then pit this process against the encounters of children with technologies, to see if there are any apparent parallels across the two topics. Finally we will speculate on how technologically literate children might change the face of user interfaces in the future.

Stages in learning how to read

Uta Firth is a German born Neuroscientist and Linguist working in London. Her main topics of study include breakthrough discoveries surrounding the treatment of autism, as well as the pedagogical literacy program for children who suffer from learning disorders such as dyslexia. Her studies conducted in the mid 1980s of child literacy development conclude with three definitive stages in the process of learning to read and write in English. [Messer, 2011]

The first stage, the *"logographic stage"* otherwise known as "*pictoral*", involves the child's recognition of word shapes based on remembering certain key characteristics of visual language. Typographic brand names, familiar first names, daily recognised signage e.g. child's house number; become familiar through habitual encounters. The child in this stage has not yet grasped the concept or logic of writing, but they are developing an acute awareness of semiotics and logotypes identities. In other terms, they are associating the landscape of word formations with specific products, processes or information. As Dehaene phrases it;

> *"The child's brain is attempting to map the general shape of words directly onto meanings, without paying attention to individual letters and their pronunciation."* [Dehane, 2007, pg 200]

An example of this is the child's recognition of the *"coca-cola"* brand mark to represent a cola flavoured soda drink.

The second stage is called the *"phonological"* or *"alphabetic stage".* In this stage, the child is converting graphemes to phonemes. It is important to understand the difference between grapheme and phenome to better understand how and why the child differentiates between these two forms of syntax. As the Collins Online Dictionary defines them;

> *"A phoneme is the name given to the basic unit of pronunciation for each language. They form the compound grapheme structures of each word (morpheme), and vary in their pronunciation and meaning from language to language."*
> [1. Collins English Dictionary, 2014]

> *"A grapheme is the atomic structure on which phonemes are built. They are the smallest indivisible letterforms/sounds that begin building up a language."*
> [2. Collins English Dictionary, 2014]

This conversion involves the child learning how to associate sounds and meanings with letterforms, and their role within words. For example, a child might read "globe" as "g-

lob-e", with three phonemes, as they are unused to encountering graphemes in this order.

It is not uncommon for students to start writing during the *"phonological"* stage. Through the process of writing, the child is,

1. Developing a wider vocabulary in their personal logographical lexicon
2. Remembering the spelling and pronunciation of common usage words in a particular language.
3. Creating a fluid neuron pathway from the visual centres of the brain to the language centers.

Learning how to decipher graphemes and phonemes as a whole unit in a single fluid motion, not only allows the child written and oral fluency of language, but the ability to *"decipher an alphabetic script".* [Dehaene, 2009, pg 202] Cognitively, this involves the child correctly deciphering the difference between graphemes in context, as well as the translation of letter-to-sound. As this orality technique develops, the brain must correctly process the verbal-visual information. [Dehaene, 2009, pg 202]

This stage in the literacy development process is vital in understanding the logic of visual letterforms. Building this understanding is important for the development of language fluency, in aural, oral and visual terms.

Understanding this stage in the development of language acquisition could play a large part in the development of a pedagogical approach to learning how to use programming languages and logic.

The third and final stage is called the *"orthographic stage".* In this stage, the emphasis is on the length of time it takes a reader to parse through literature, as opposed to the development of parsing skills. The reader is no longer slowed by word length or grapheme complexity. They are recalling familiar words from a built up vocabulary developed in the first stage and refined in the second.

This is the stage in which fluency is developed. It is also the stage in which dyslexia and other learning difficulty problems will become evident, and in some cases, treatable.

While the root cause of dyslexia remains unknown to researchers, it is understood that deficits in the neuron pathways of the brain can lead to the

misinterpretation of information, and this can lead to difficulties in the literary development of children. [Dehaene, 2009, pg 238] [6]

The accuracy of *"recognising written words as a whole"* depends on *parallelism*. Parallelism[7] is the ability to recall the correct use of graphemes and word meanings from a lexicon created during the *"phonological stage"*. The idea of parallelism in learning the logic of software programming will be explored later in this chapter. These three stages are not structured in a rigid chronological order, but revisited cyclically throughout the child's early development stages. [Messer, 2011]

This insight may be the saving grace of many mono-lingual students who then take on the task of learning languages after the plasticity of early teenage phase of cognitive development as solidified. The task of learning a language gets continuously more difficult as we age, as the neuropath ways reaching between the visual and linguistic centres of the brain harden over time. [Dehaene, 2009, pg 220]

Often, the *"orthographic stage"* follows the *"phonological stage"*, but it is important to note that ongoing research into a broad scope of test subjects of early-learning readers suggests that this three level model of literacy development isn't necessarily indicative of learning how to read. Other models exist, especially in the development of non-latin languages. For the sake of brevity, this thesis will focus on Frith's three stages, as they are a globally recognized standard, although challenged.

Early Signs of Technology Literacy

"Technology codes our minds, changes our OS.
Apple products have done this extensively.
The video shows how magazines are now useless
and impossible to understand, for digital natives.
It shows real life clip of a 1-year old,
growing among touch screens and print.
And how the latter becomes irrelevant.

[6] In the interest of brevity, this thesis will recognize the importance of knowing how learning difficulties work, as well as their role in constructing a rounded argument.

[7] Parallelism is the repetition of a syntactic construction in successive sentences for rhetorical effect. [3. Collins English Dictionary, 2014]

Medium is message.
Humble tribute to Steve Jobs,
by the most important person : a baby."

– Jean Louis

[UserExperiencesWorks, 2011]

On the 6th of October 2011, Internet activist Jean Louis posted a video of his one-year-old girl using an iPad with gesture fluency, and then a clip of her trying to operate a print magazine as if it were a tablet. [UserExperiencesWorks, 2011]

In the video, the child can be seen 'pinching' the pages, as well as 'swiping' to change the page/image or scroll. In the video, Louis promotes the idea that for his daughter to be born into an age post Steve Jobs, *"a magazine"* will forever be *"an iPad that does not work".* [UserExperiencesWorks, 2011]

Daniel Donahoo published an article in Wired magazine, condemning the statement as fraudulent, as Louis wasn't taking into account the natural development of his daughter's motor skills.

Donahoo, a 'tech-dad' father of two and software developer/writer, criticized Louis as being unaware of how user experience [UX] design works as well as oblivious to the motor skills his daughter is developing away from the technology, during analogue playtime. Donahoo elegantly phrases this,

> *"So, the cleverness of the user experience design is not that the iPad changes the girl, but that it leverages off fine motor skills that are so integral to our ability to engage with the world that a one year old can begin to engage with a touch screen"* [Donahoo, 2011]

Early childhood development preceding literacy education requires babies to learn a wide range of motor skills, some of which include 'pinching', 'waving' and 'rubbing'. These motor skills are developed through a wide variety of 'play' scenarios that include the very physical tasks of molding dough, using pegs, tearing/scrunching paper, turning cards, shaking dice, using scissors and so on. All of these play tasks

require the infant to execute and develop a number of hand-eye co-ordinate dexterity, that refine into fluid and controlled motions over time.

Even Stanislas Dehaene confirms that from a young age, babies and toddlers *"discriminate among objects, using their contours, their texture, and their internal organization."* [Dehaene, 2009, pg 198]

Improving the hand and finger dexterity of children will strengthen their grip. This will encourage the correct use of pencil grip and forearm to upper arm mobility when writing. [Buxamusa & Mahoney, 2010]

These motor skills are remarkably similar to the motor skills necessary to navigate tablet/gesture based technology. It is interesting how the development of learning the functionality of a tablet in isolation of learning how to write, will stop once the first stage of mobility is reached. That is to say, if the child was indeed one of Louis' *"digital natives",* and society did not require it to write with a pen, would it cease to require its hand motor skills to develop in the way necessary for it to operate a pen? What would this mean for hand based crafts, occupations etc.? It is no wonder that children are picking up tablets and know how to work the basic navigation of the OS and individual apps.

Future ubiquitous technology

Ender's Game is an Orson Scott-Card classic Sci-Fi novel from 1985, with a 2013 movie release of the same name. The narrative requires young children — approximately 3 years up to 11 years — to work with futuristic UI gesture based technologies.

If we look at the movie adaptation as an example of using ubiquitious computing as an everyday device, we can see the postulated motorskills required to operate the complex data structures of these computers and technologies. [FIG 1]

Ash Thorp is the creator of the *Ender's Game* UI. What is most interesting about the design is the way in which the users are seen to be typing. In FIG 2 we can see not a keyboard, but a set of 'rings' surrounding each finger of the user, and a 'bar' for the thumb. Is Thorp postulating that in the future we will do away with glyphs on a qwerty board and restrict the typography to a non-editable entity on our screens? This playful approach to how user interfaces will look and feel as a part of future technologies is an

example of the lyrical approach science fiction takes on speculative ubiquitious technology of future ages. As Egan quite elegantly describes Thorp's work;

> *"Nested spirals of cosmic calculation spool out at the whim of a young boy-turned-war-commander."* [Egan, 2013]

What does this mean for the development of reading, and further still, the development of reading in a non-writing world?

Ubiquitous technology is comprised of the ambient low resolution interfaces that surround humanity. Car dashes, ATM machines, household appliances, computers at work; ambient technology surrounds us. If we were to adapt some of the conceptual UI designs created by Ash Thorp and his collaborators, what would change in the pedagogical approach to literacy in the children of the future?

How we learn to Programme

Introduction

> *"Computational thinking will influence everyone in every field of endeavour. This vision poses a new educational challenge for our society, especially for our children. In thinking about computing, we need to be attuned to the three drivers of our field: science, technology and society. Accelerating technological advances and monumental societal demands force us to revisit the most basic scientific questions of computing."* [Wing, 2008, pg 3717]

There are many different kinds of teaching practices that exist when constructing a syllabus for the education of primary level children. One such example of a widely used technique is the constructivist approach. [Dehaene, 2009, pg 197]

This approach requires the student to obtain a level of independence from the teacher at an early stage in the education system, in order to develop the skills to 'teach oneself' while receiving guidance from the system via text books, online tutorials, classroom assistants and teachers. [Prawat, 1992, pg 356]

Under the constructivist method of learning, the teacher obtains a level of independence from the student, which can benefit the student as the teacher has room to act as a motivational agent and is not required to be constricted by guiding the student through a rigid syllabus.

Dramatically changing the focal point of education from from a) guiding the student through a strict set of educational goals, to b) teaching the student to guide themselves benefits the student in the long term. Ultimately, the desired results of each practice are identical; the student is to reach a level of understanding of a subject at a particular point in the educational system.

The constructivist approach to learning allows the student to educate themselves to a similar level of understanding as if they were guided step-by-step by a trained professional through the material. This skill of self-teaching is important when it comes to teaching oneself skills outside of the classroom. For example, a designer working in a studio is more likely to teach themselves a new software program or a new programming language, if they have been successful at teaching themselves a similar skill before. As Prawat phrases it;

> *"Being provided with a new set of theoretical or conceptual "lenses" can be empowering for teachers, but it also complicates their lives. This is particularly true of constructivist theory. While there are several interpretations of what this theory means, most agree that it involves a dramatic change in the focus of teaching, putting the students' own efforts to understand at the center of the educational enterprise."* [Prawat, 1992, pg 356]

Dehaene rjects the constructivist approach to learning, citing that is it too *"simplistic".* His argument that as the brain *"remains plastic"* as the child grows into adolescence, the window of opportunity for learning and education stays open for longer, thus allowing for the constructivist approach to be drawn out for longer than is it. That is to say, he believes that dismissal of the groundwork laid out during the early stages of self-development should not be dismissed so easily by the educational system. By allowing children and young adolescents the opportunity to continue self-learning in formal education via the constructivist approach, will allow them to self-teach when they are adults. [Prawat, 1992, pg 357]

Programming Learning Tools

> *"Accomplished teachers create, enrich, maintain, and alter instructional settings, materials, and strategies to capture and sustain the interest of their students and to make the most effective use of time."* [Vygotsky, 1978]

According to *InTime* (Integrating New Technologies Into the Methods of Education), a good teacher is one who can enrich the lives of their students through an intellectually entertaining engagement with coursework material. They know the material they are teaching intimately, and they are aware of the class standards through the observation of students during play, practice, study and their vocal input.

While they are no substitute for this nurturing environment, online tutorials are becoming more and more popular due to their ease of access and adaptability to suit a wide variety of students. Trying to replace the sensation of the curatorial teacher-classroom effect with an online tutorial is not a solution, but rather a different way of learning. It is important to focus more so on the pedagogy present in online learning material, as the internet is more readily available than classroom learning.

It is therefore in the interest of this thesis to examine and scrutinise the pedagogy that exists within online communities that are learning to program, rather than classroom techniques, to better understand which method is more useful to the graphic designer – who is presumably beyond learning in a classroom. Samurcay notes this in his research;

According to Renan Samurcay, it is important to establish a hierarchy of the concepts and procedures used in programming languages. This is to ensure that the novice programmer can start with simple expressions and gradually build up an internal lexicon of logic, syntax and language specific terminology. His testing of beginner programming students in the late 1980's provided researchers with empirical evidence that there is a clearly defined structure the beginner programmer adapts when learning how to code. This structure is not a linear pathway, but rather evidence of trying to apply logic learned prior to learning coding logic. That is to say, the logic one learns through literature.

Designers – graphic, product, UI etc. – are the interpreters of information and act as a kind of mediator between user and information. One way in which information can be interpreted is through the gathering, structuring, deconstruction and re-construction of data. This process is identical to the process the coder must endure to achieve a finished program.

> *"Data structuring – one of the most important concepts of programming – requires not only the concepts of variable, looping etc. but also some concepts directly related to the modelisation of the given problem."* [Samurcay, 1985, pg 37]

Designers undergo this type of information processing with the use of a brief or instruction. Often the contents of the brief will give an outline of the kind of information to be used; the desired outcomes, the method of process to be applied to the information and the desired product result. These considerations are also to be made by the programmer, as they have to deconstruct a set of guidelines to favour a desired result. This cross-over is important when we consider combining computer programming with design education.

The three main examples of learning techniques that this thesis will focus on include motivation, 'the flow' theory, and literary features such as metaphor and simile.

Motivation as a Learning Tool

There are two forms of motivation that spur the student on to learn a new skill. The first approach is the needs that inspire the student and the second is the flow of information encouraging the user. In this case, we can break down motivational needs and motivational encouragement using Maslow's Hierarchy of Needs as well as Csikszentmihalyi's flow theory.

Maslow's *Hierarcy of Needs* is a set of linear motivational needs that are often visualized as a pyramid. This pyramid can contain between five [developed in the 1940s] and eight [developed in the 1960s-1970s] levels of motivational needs. As we

progress up the pyramid, the need for the levels decrease overtime. For example, the base of the pyramid details the fundamental needs of humans (which are not dissimilar to the basic human rights laid out by the UN governing body) while the upper parts of the pyramid are desired only by those who live in a society that provides the bottom strata regularly.

In explaining the motivation for a user to want to learn a new skill, we can account for this desire to fit into one of the top five levels of the eight level pyramid.[8] These levels are as follows;

Level 3. Social Needs – Belongingness and Love – work, family, love, affection.

Level 4. Esteem Needs – self-esteem, achievement, mastery, independence, status, dominance, prestige, managerial responsibility etc.

Level 5. Cognitive Needs – knowledge, meaning etc.

Level 6. Aesthetic Needs – appreciation and search for beauty, balance etc.

Level 7. Self-actualisation Needs – realizing personal potential, self-fulfilment, seeking growth and peak experiences.

Level 8. Transcendence Needs – helping others to achieve self-actualisation.

[McLeod, 2007]

Progressive challenges and 'The Flow' as a Learning Tool

"We consider the problem-solving situations function not only as a criterion of what is learned, but also as a situation in which the students encounter some new concepts. So each of the given problems

[8] It is assumed that if the user is both attempting to learn a new skill and is of a sound mental and physical state, then they will be fulfilling the first two levels of Maslow's Pyramid. These two levels encompass the need for food, shelter and safety for the user and their family.

constitutes for us a didactical situation in which we have introduced some new concept." [Samurcay, 1985, pg 40]

That is to say, each new problem should present to the student, a previously un-encountered piece of logic and/or syntax. This is to challenge the user and present an achievable – albeit sometimes difficult to finish – goal at the end stage of each tutorial.

This method of work and reward is called *The Flow*, an idea engineered by psychologist Mihaly Csikszentmihalyi. [MikeW, 2009] *The Flow*, explains a user's immersion into a project. That project might be problem solving, gaming, a learning tutorial, watching a film [in this instance it is the narrative of the film that is drawing the user in] etc. An *optimal state flow* may be achieved if the challenges encountered in the project match the ability of the user, but only at a stretch. That is to say, if the user finds the work load to be too easy, they fall out of the flow into boredom. If the work is too hard, the user falls into a state of anxiety. Both of these results, boredom and anxiety, create obstacles that may prevent the user from advancing through the tasks.

As FIG 3 suggests, when the user is not engaging with the task at hand, they are in a state of relaxation, they are not improving their skills. To engage the user once again, they simply have to choose a more difficult task to enter into the flow stream. By gradually increasing the difficulty of the tasks to suit the ability of the user and keep them focused on slowly and steadily increasing their skillset, the user will improve their ability to perform that task with enthusiasm and a sense of self-achievement.

The Flow method also demonstrates that the user is improving their skills more when they are aroused. This method may be adapted by tutorial sites when structuring their learning programs. If you can keep the user entertained while teaching them the skills necessary to progress through the coursework, they will be more likely to stay engaged with the work. [MikeW, 2009]

Metaphor as a Learning Tool

Seven Things You Should Know If You're Starting Out Programming, is an article from *The Guardian Developer Blog* by Jonathan Richards. It details a brief account of his immersion into the world of programming while heading up the Data Journalism unit at the Times. [Richards, 2011]

He references the *catch a shooting star* principle, which he encountered when taking an online tutorial. Coined by *Hackity Hack* Ruby tutorial expert *_why*, the username of *"the world's most unusual and beloved computer programmer"* Jonathan Gillette. [Lowrey, 2012]

Levin explains that if you don't *"do something"* with the information you've created on a computer – that is, the program file you've created, or the string or integer etc. – then it will vanish. He calls the process of *"holding onto"* this information as if one is catching a shooting star.

Using metaphor to explain the process of saving values as variables is of use to the design profession, as visual communication employs visual metaphor as a method of information delivery. Therefore, using language and literary techniques that are common to both practises could be an effective method of explaining the logic of coding.[9] Certainly, this presumes that the designer – or indeed, any student – is literate enough to grasp the meaning of the metaphors. This variable is of course at the discretion of the student.

Author of *Hello Ruby*, Linda Liukas is determined that *"stories are the most formative parts of our child hood."* For her, as a teacher of programming and having grown up under the influence of Finnish storytelling, narratives are a useful way to learn about the world around us. Her storytelling heritage inspired her to write her debut book *Hello Ruby*.

Goal Setting as a Learning Tool

> *"Programming is just like any other art, say like painting. Learning the initial constructs is easy but getting good at it takes a lot of practice. So keep coming up with cool new projects to code, and in a matter of time programming will become very natural to you."*
> [Weber, 2012]

[9] In fact, the inventors of the Python programming language, named the language after the British Television series, *Monty Python's Flying Circus*. This is because they wanted to create a playful approach to the teaching and comprehension of the language, and by incorporating many of the television series sketches into tutorials and the syntax to make it more fun for users. [Taft, 2010]

In an interview with journalist Harrison Webber, *Programr* Founder Rajesh Moorjani explained that his main motive when he was learning how to code was to set goal projects, which helped to improve his coding logic and syntax. He created *Programmr* based on his desires to create working projects that he could share with his friends and family.

By completing a project from start to finish, Moorjani was fulfilling his desire to reach Maslow's level 7 of the Motivational Needs Pyramid; a feeling of self-fulfilment and personal pride in one's abilities.

Teaching children; Literature and the Art of Storytelling

> *"Teaching has become more than an activity that conserves valued knowledge and skills by transmitting them to succeeding generations. Therefore, teachers also have the responsibility to challenge existing structures, practices, and definitions of knowledge; to invent and test new approaches."* [Vygotsky, 1978]

This is exactly what Linda Liukas is doing. Founder of *Rail Girls*, she works for *CodeAcademy* and is the current Finnish representative on the *Digital Champions* European Union group.[10] [Liukas, 2014b] Unlike the 'in classroom' teachers Vygotsky is talking about, Liukas is predominantly an online teacher, and as such, she adopts a more literary approach to imparting knowledge. Through storytelling, Liukas creates a personable experience for the online student.

> *"Ruby's world is an extension of the way I've learned to see technology. It goes far beyond the bits and bytes inside the computer. This is the story of what happens between the ones and zeros, before the arrays and the if/else statements."* [Liukas, 2014a]

According to Maria Alterio's 2003 research paper on using storytelling to enhance student learning, the act itself can transcend cultures and communities. Just as Uta

[10] The Digital champions are responsible for the promotion and upkeep of the Digital Agenda for Europe. This initiative is to boost European revenue through the promotion of digital technologies in each Country. [European Commission, 2014]

Firth's *"Logographic Stage"* sees children applying meaning to familiar typographic logos, Alterio states that before the ability to articulate ourselves through coherent reflexive descriptive language was possible, we *"learned to make sense of our world through stories."* [Alterio, 2003, pg 1]

In her promotional video, Liukas talks of the stories and protagonists she grew up with. *Madeline, Pippy Longstocking,* and *Little My* from *The Moomins*. She recognised the importance of using fiction to explain the logic and principles of programming, how powerful a tool it can be to simplify a principle and how to use that principle in context. She discovered her character *Ruby* while studying, as she would draw her in the margins of her programming lecture notes. Creating a character to exist solely to explain programming is a creative way to simplify logic.

> *"... writing software is about expression, creativity - and practical application. Our kids should learn to bend, join, break and combine code in a way it wasn't designed to, just as they would with crayons and paper or wood and tools."* [Liukas, 2014]

Supporting this argument on using story as a learning tool, is Jenette Wing. She states that;

"Through effective visualization and animation, even at early grades we can viscerally show the difference between a polynomial-time algorithm and an exponential-time one or show that a tree is a special kind of graph." [Wing, 2008, pg 3721]

Formal Education of Children in National Schools Across Britain

John Naughton is a United Kingdom based Professor at the Open University, where he teaches *Public Understanding of Technology*. He also writes for the Guardian Online, on *the Developer Blog* section.

On the 31st of March, 2012, Naughton published an article which calls for the reform of the Computer Science curriculum in British primary schools.

Naughton references the philosopher Gilbert Ryle when he says that Computer Science has been taught, up to and including the present time, as a "*category mistake*". That is to say, that an error has occurred in teaching things of one kind as if they are of another kind. Teaching children about the technology of today through using older

machines, datedOS systems, and through using various obsolete software products does not equip them for a future in technological literacy. That they are simply not prepared to interact with newly release software, OS systems and computer hardware, if they have not yet encountered them. In his article he phrases his request of the UK education system in the form of a manifesto, which could be adapted by the board of education as they reform their pedagogy on the subject;

> *"Starting in primary school, children from all backgrounds and every part of the UK should have the opportunity to: learn some of the key ideas of computer science; understand computational thinking; learn to program; and have the opportunity to progress to the next level of excellence in these activities."* [Naughton, 2012]

According to Naughton, teaching an effective system of Information Technology in Primary schools requires effectively teaching the two distinct areas of computer science;

1. A set of key concepts that govern the make up of a networked society. These concepts include the historical groundwork to understand how we came to live in a networked world e.g. to learn about Alan Turing, Xerox Park etc.
2. To teach computational thinking as one way of problem-solving, conceptualising and troubleshooting information.

Teaching these two concepts

Wing outlines two problems to avoid when teaching children a revised syllabus and course outline for Computer Science in her 2008 paper on reviewing current pedagogical approached to Education;

1. *"We do not want to let the tool get in the way of the concepts"*
2. *"We also do not want people to just be able to use the tool but not have learned the concepts"*
3. *"We do not want people to come away thinking they understand the concepts because they are adept at using the tool."* [Wing, 2008, pg 3721]

Naughton and Wing are in agreement over similar details that criticise what needs to be changed about the current standards in education for computational science in primary schools, as well as aware of how to go about implementing these changes. By combining their research, a set of vague milestones can be outlined for what is to be expected of future generations in terms of capability when dealing with technology in the future.

Milestones

It is important to remember that these milestones are purely based on observation and not empirical data research.
Before beginning to implement any of these guildelines, it would be necessary to conduct trials and tests based in classrooms, with teachers at the primary learning source, as well as the computer as the primary learning source for children and adults. Wing cleverly references a maturity system that corresponds with material learned in school. We can only speculate as to what the milestones might be, but without directly running tests, it is almost impossible to know when exactly in the education system is best to encouter and effect these milestones;

> *"What would be an effective ordering of concepts in teaching children as their learning ability progresses over the years? By analogy, we teach numbers to children in kindergarten (when they are 5 years old), algebra in junior high (12 years old) and calculus in senior high (18 years old). There may be many possible ways to structure the progression of computational thinking concepts; which is the most effective for which kind of learner?"* [Wing, 2008, pg 3721]

Milestone 1.

Introduction of the 'tool' to the student. This stage requires the student to understand that the tool, the hardware in use, is not unique and they will encounter many like it in their lifetime. If possible, exposure to multiple kinds of technology will benefit this understanding.

Milestone 2.

Learning the functionalities of computers, the 'tools', available to humankind and understanding the importance of becoming fluent in the use of many different tools. That is to say, to eliminate the bias one had for one operating system over another. This can be achieved by properly explaining the pros and cons of Mac machines versus Windows and Linux machines, and requiring the student to be exposed to running similar tasks on all three OS systems. This is with the hope that the student will better understand that different structures exist across platforms and OS systems, and they will gain a fluency when deciding which system is best for which task.

Milestone 3.

Learning how to problem-solve using computational logic. This should be taught alongside the learning of mathematical thinking, engineering thinking and prose logic. Teaching this to young students can help eliminate any ambiguity when they come of age to start coding and having to make choice about the logical structure of code for a program.

Milestone 4.

Learning the history of machines, technology, logic, computational languages as well as ethics in the age of technology. This stage could also begin when the student is at a very young age, to allow for the ethical implications of technology to sink in over time and help form opinions of code structure and functionality when they are older.

It is an imperative to teach about the latest trends in technology, as well as a historical grounding in past inventions. This is to familiarise students with the idea that as society's demands change, so too does technology change.

This stage could also encompass elements from the *digital humanities*, as the student will be exposed to many of the theorists, historians, heros, practitioners and critiques of this field of study.

Milestone 5.

Learning computational theory, its function in the world of technology, its logic and its syntax. This stage begins as soon as the child is old enough to have progressed from *logographic* to *phonological* stage comfortably in their literacy education. Introducing the logic of variables, loops, functions, and specific syntax in the form of

code can be broken down and digested by those who recognise the importance of phenomes and graphemes in language acquisition.

It is important to expose the child throughout their education to a variety of computer languages, but to also ensure proficiency in at least one of these languages.

The teaching of these languages can beging with the teacher as the primary source of content explanation and motivation for the child to retain the information presented. It is interesting to speculate how changing this teaching structure to a more constructivist pedagogical[11] approach might encourage the child to beging grasping the theories of computational logic for themselves, and lay the groundwork for the child to teach themselves new languages as they grow older.

Establishing this independence early on, and coupling it with an emphasis on the rapid changes in technology capabilities, might allow for more flexible developers and designers in the future, who can not only think conceptually about speculative technology, but perform coding to match their speculated functionality.

Milestone 6.

The student as developer, designer, and encourging them to create their own programs. This stage is of course the desired outcome of teaching computational logic to any level of student. It is interesting to speculate as to what age this independence and capability might take place. If with the effective implementation of computer science in schools this stage began to occur in younger students, how would it change the face of ubiquitious computing in the future?

[11] Constructivist pedagogy "refers to the idea that learners construct knowledge for themselves---each learner individually (and socially) constructs meaning---as he or she learns." [Hein, 1991] For more information on the Constructivist approach to teaching programming to children, see the Introduction section of *How we Learn to Programme* in chapter one.

Chapter Two

Design Education and Practise

What is graphic design?

> *"If we want to reinvent and re-construct graphic design, we have first of all to create linguistic distinctions capable of grasping a new reality that otherwise would not be understood if we remain bound to standard terminology."* [Bonsiepe, 1994, pg 47]

In his 1994 report, Guy Bonsiepe called for the reformation of graphic design to be inclusive of user interface and user experience design. He recognised the new roles of designers during the early and mid stages of the digital revolution and dawn of the Internet. He elegantly and succinctly expresses this ideal in his writing that *"...graphic design"* should function *"as information management and link it to the notion of information..."* [Bonsiepe, 1994, pg 48]

He further adds that the shifting sands of design should include allowing the designer to have an authorial hand on their work. That is to say, to allow the styles of designers to come through their solutions to briefs and allow for a sense of authorship for the individual designers themselves;

> *"I propose to shift the role of the graphic designer from translation of information from a non-visual state into a visual state, to the authorial organization of information."*
> [Bonsiepe, 1994, pg 48]

Another misconception of graphic design is that it is intrinsically linked with advertising. And while it is true that graphic designers play a primay role in the creation of advertising campaigns, it is important to shatter the idea that all graphic designers are interested in advertising. It is more true to say that some graphic designers do not work in advertising at all. Bonsiepe

recognises this rejection in the mid-nineties and calls for the *"...liberation of graphic design from its ancillary status in the domain of advertising and promotion."* [Bonsiepe, 1994, pg 49]

The professional approach of the info-designer, as Bonsiepe now calls the graphic designer, who is free of advertising and has authorship over their work, is a three stage process at its most basic.

Stage one – the concerns of the brief are taken into account, and addressed under a particular perspective, within a particular context. For example, if the brief were to design a wayfinding system for the Himalayan mountains, one would take the Himilayan ecology, politics, sociology and history into account, rather than research unrelated topics.

Stage two – the designer will use the objects of their trade to create a remedy to the brief. These objects include cognitive tools of the trade; such are logical reasoning, objective and subjective viewpoints, programming skills, understanding of design fundamentals ; as well as analogue/digital tools at their disposal e.g. 3D printers, specialised software, photography, social media and so on.

Stage three – implementation of the designer's answer to the brief. It is prior to the release of this stage that the designer should encounter the ethical implications of their solution, and worked to create a balanced and fair answer to the brief.

This three stage process is not indicative of all designers and their individual processes, but it is a good starting point when breaking down the process of design to understand how and why it occurs in society.

It is important to know the functionality of design in today's world because it shows the important of the designer's role as mediator of information in an information society. The designer is therefore responsible for providing a method of understanding information for the user to *"navigate in a highly complex information universe."* [Bonsiepe, 1994, 49]

A list of competencies necessary for the info-deisgner to have, are listed on page 50 of Bonsiepe's report. According to the researcher, the info-designer should be able

"*to use computer programs for scripting, illustration, image-editing, animation, and video."* This is to ensure that the designer can execute what he cites to be a growing demand of the designer; "*interface design for computer programs".* Twenty years later, Bonsiepe has been proved right, the growing demand for UI (User Interface) and UX (User Experience) design is growing as fast as people are buying the products that demand such experiences.[12]

The *"socio-ecological impact of the work of the info-designer"* [Bonsiepe, 1994, pg 50] is also important. This is to ensure the designer, who is operating outside of a standardised hippocratic oath equilivant, is maintaining high standards and releasing accessible and quality products into the world. Without these standards, there would be no stopping designers from creating unnecessary, dangerous and/or inefficient products/systems in the world.

It is therefore important for design pedagogy to include a contemporary as well as historical analysis of *"the cognitive domain of the design process."* [Bonsiepe, 1994, pg 51] We can take the Irish examples of the National College of Art and Design (NCAD) and Institute of Art, Design and Technology (IADT). Faculty members of each design course maintain one foot in the industry as well as one in the realm of academia. For example, in NCAD, Francis Halsall is a visual culture lecturer who maintains connections with the Irish Museum of Modern Art as a fellow collaborator on a range of podcasts. In IADT, David Smith lectures in Typography & Joint Programme Co-Ordinator[13], as well as a senior designer, art director and managing director for a design firm in the centre of Dublin. Similarly, McCarthy and De Almeida from their 2002 research conclude that the graphic designer is responsible for the interface that exists between content and user, product and consumer, government and citizen. They elaborate on the roles of the designer and suggest that they almost have a kind of control over the actions of the public; *"designers mediate, translate, and amplify the visualized environment, giving tangible form to the objects and experiences that inform, persuade and entertain us."* [McCarthy & De Almeida, 2002, pg 103]

[12] Irishjobs.ie, on the 19th of February 2014 has over 200 UI jobs going in Ireland. Visit http://www.irishjobs.ie/Ui-Designer-Jobs for more details.
[13] Availabe at http://www.iadt.ie/en/design/studiolecturers.html

Reasons designers should learn to program

While the primary concern of this thesis is to investigate the role of a computer literatre designer, it is also necessary to consider the changes society would face if a range of other disciplines became as computationally literate as those who interact with computers on a daily basis. To reiterate the over-arching question proposed within the introduction of this thesis; *can humanity benefit from a design generation that is computationally as well as computer literate*, begs a secondary question; *how will computational and computer fluency benefit the designer?*

By placing the designer as a mediator in society, we create an interesting power struggle for the designer to ethically face when designing for public use;

> *"... this discourse not only acknowledges the unique qualities of the medium,*
> *but also critically examines the social and cultural role of the designer*
> *in reinforcing or legitimizing power structures."* [McCarthy & De Almeida, 2002, pg 104]

Wing recognises the impacts of scientific discovery on society as a driving force for computation literacy. Considering previous statements made by Bonsiepe to point out that designers mediate the world society lives in, it is believable that they exist to evolve and adapt alongside these scientific discoveries. In fact, Wing suggests that this transient existence be "*celebrated",*

> *"... scientific discovery feeds technological innovation, which feeds new societal applications; in the reverse direction, new technology inspires new creative societal uses, which may demand new scientific discovery."* [Wing, 2008, pg 3722]

This pin-points a past failure of graphic design history. With the invention and popularisation of personal computers, analogue based graphic designers faced a decision which dramatically changed their roles in society.

Do they upskill and learn how to use a computer as a part of their design process, or do they remain analogue and eventually lose their studios to digital based ones?

There is a lesson to be learned from the past for graphic designers and visual communicators – it is better to adapt to the tools available today, in order to adapt to the tools available tomorrow. By upskilling, the designer is not only stabalising their career, they are evolving with the changing times and can teach what they have learned to a new generation.

Putting the role of graphic designer into the position of mediating information between government and citizen creates a power struggle that relies on the designer's sense of social, political and ethical responsibility. Stewardship becomes a trait of the designer, and they are forced to reason with the ethical implications of their actions and choices when answering a brief. AIGA (American Instuitute for Graphic Arts) is a governing body of graphic designers that exist to mediate, translate and protect graphic designers and the practitioners of this discipline.

On the AIGA website, is a statement regarding the importance of *transparency* in the creation of information between citizen and agency. This trust distinction is important as without it, the ethical driving force of the graphic designer can be called into question by either recipient or source of the information in question.

This statement echoes Wing's recognition of the changes scientific discovery impinges on society.

> *"Design can strengthen democracy by building trust in the communication between government and the governed. Trust emerges from understanding; design is a critical intermediary in making the complex clear and enhancing understanding. In most of the interactions between government and its citizens, there is an exchange of information. Communication design can simplify and clarify this process; designers can also design service experiences that are more effective, more comprehensible and more efficient."*
> [AIGA, 2014]

Relating to this, is the ethical habits of corporations on society. In his 2007 article on the ethical habits of corporations, John L Campbell questioned under what conditions corporations are likely to act in a socially responsible manner. [Campbell, 2007, pg 947]

He discovered that if the members of an organization are exposed to peer pressure that encourages an ethical business environment, the corporation will behave in an socially ethical and responsible manner.

There are several dimensions that dictate what corporate responsibility is. The most relevant for this dissertation is the topic of *"how it treats the government with respect to operating within the law and not trying to subvert it."* [Campbell, 2007, pg 947]

If we compare this with a major challenge facing today's design generation, the inaccessibility of some teachnologies on some minorities within society, we can see a distinct rationale as to why it is important to understand corporation's ethical habits in a digital society.

An example of corporations subverting state-regulations is Amazon.co.uk.

If you run the index page of the United Kingdom based Amazon site through the WAVE website accessibility generator, there are a number of accessibility errors encountered. [14] This is because of clashes between stylistic choices made by the company and the current standards of accessibility outlined by the W3C – the World Wide Web Consortium.

Campbell suggests that this irresponsible behaviour will be rectified if state regulations are put in place for this practise, as well as incentivising corporations to abide by them. As Campbell phrases these two points,

"Corporations will be more likely to act in socially responsible ways if there are stong and well-enforced state regulations in place to ensure such behavior, particularly if the process by which these regulations and enforcement capacities were developed was based on negotiation and consensus building among corporations, government, and the other relevant stakeholders." [Campbell, 2007, pg 955]

[14] On the 23rd of February 2014, there were 8 errors, 16 alerts, 37 features, 47 structural elements and 26 contrast errors amiss. This information is available at the following site address, http://wave.webaim.org/report#/http%3A%2F%2Fwww.amazon.co.uk%2F
Due to the changing nature of the Amazon homepage, it is important to note that these figures fluctuate.

"... if there is a system of well-organised and effective industrial self-regulation in place to ensure such behavior, particularly if it is based on ther perceived threat of state intervention or broader industrial crisis and if the state provides support for this form of industrial governance." [Campbell, 2007, pg 956]

The design decision to evolve the website beyond today's accessibility standards creates the problem of inaccessibility while pointing out the constrictions of these accessibility standards. Campbell calls for someone to step up to monitor this kind of behaviour in design, to both repremand companies who are not creating a balanced accessible website, but also to take note of the aging accessibility standards that may need to be reconstructed and rewritten in future.

"...if there are private, independent organisations, social movement organisations, institutional investors, and the press, in their environment who monitor ther behavior and, when necessary, mobilise to change it." [Campbell, 2007, pg 958]

This role could fall to the designer, to act as a conscientious participant in the rectification of these accessibility standards. This type of mediation is an integral part of computational theory and digital humanities, that could be addressed in *Milestone 4* of the new pedagogy.

Chapter Three
Social and Cultural influences of Computer Literate Groups

Introduction

> *"The success of our information technology, including computers and communications, has raised society's expectations of us. People now demand availability, 24 hours per day, every day, 100 per cent reliability, 100 per cent connectivity, instantaneous response, the ability to store anything and everything forever, and the ability for anyone to access anything from anywhere at any time."* [Wing, 2008, pg 3723]

Considering the raised expectations of technology within society, it is necessary to consider how not only designers are educating themselves in computer science. This chapter will look at a number of case studies involving a constructivist approach to teaching oneself computational languages in a variety of settings. Examples of peer-to-peer groups, both online and in person, solo online tutorials, solo online workshops and in-person group workshops for children will be formally introduced, with the intention of creating a brief outline of the types of learning approaches availabel to today's computer owner en mass.

Informal group work, through camps and workshops

a. Programr

Programr is an online resource for students learning how to program and who want to develop web and local hosted applications. The aim of the site is not to provide tutorials but rather the practical applications of programming as explained examples. They do this by creating *"labs, practice tests, assesments and contests."* [Weber, 2012]

Rajesh Moorjani mentions in this article that the intention of *Programr* was to create an atmosphere where programming students can create real projects to show their friends and families the progress they're making. He stresses that this social interaction with learning the basics of coding is integral to the learning process as *"it provides the*

much-needed motivation and pats-on-the-back to keep on creating cool stuff." [Weber, 2012]

Moorjani's comparison between art and programming is interesting. In this interview he states that becoming a proficient programmer is similar to the art of painting. In his words, *"Learning the initial constructs is easy but getting good at it takes a lot of practice. So keep coming up with cool new projects to code, and in a matter of time programming will become very natural to you."* This is excellent advice for the beginner programmer!

b. Coder Dojo

The term *'Dojo'* means *"a place of the way"*, and in Japanese it means the space in which you train in something that eventually leads to the way of enlightenment. [Kendo for Life LLC, 2011]

Applying the Eastern philosophy of enlightenment to the process of obtaining mastery over a skill, in this instance the art of learning programming, reiterates the use of metaphor in Educational models. Creating a tangible

CoderDojo is a global community of programmers teaching their craft to students from ages five to seventeen.

The company was founded by Irish programmer James Whelton and American Bill Liao, when James was just a secondary school student in Cork. He founded the practise after hacking into an iPad Nano and impressing some younger students who also wanted to know how to hack into computers. His first step was to set up a computer club in his secondary school where he began teaching html and css to younger students. Bill Liao, an entrepreneur and Philanthropist saw potential in James' skills as a teacher, programmer and businessman and offered to help fund what is now *CoderDojo.*

Owing to it's popularity in Cork, the group set up a Dublin cell group Google's Montevetro building. It wasn't until the group was set up as an open source project that the company finally saw international groups spring up in cities across the world. [CoderDojo, 2014]

c. Pyladies

"We are a group of women developers worldwide who love the Python programming language. We write code by day or night. Some of us hack on Python projects on the side, while others work full-time on Python development. But it doesn't matter. We all just like writing Python code, and that's what brings us together." [Lee, 2014]

PyLadies is a global group set up to act as a support network for female programmers who use the versatile language Python. Their mission statement is to create a nurturing environment for the female Python programmer, be they using the language in their profession, as a hacker on the side, or just using it for fun.

The group is split into location based cell groups, which operate on the same mission statement and desired outcomes as the global group. This division allows for an on-the-ground presence in every major city, to allow for a physically present support group. This system of support is not unknown to practitioners of religious faiths. Major world religions operate under the same system set up as cell groups, churches and communities of particular faiths.

The Irish division of the group resides prominently in Dublin, with monthly meet-ups in the IFSC buildings along the quays in the city centre. Set up on the 25th of October 2013, the group is open to members with extreme experience in Python to those just starting out as beginner programmers.

To apply for a position at these meetings, one simply has to create an account with the online site *MeetUp*, and 'follow' the PyLadies Dublin group. [Lee, 2014]

This form of social-media based information is not uncommon to the digitally literate. As we can see from the research gathered for the support and education sections of this thesis, one can potentially learn everything there is to know about computer programming, languages, logic, networking etc. from the Internet. Social groups, like the PyLadies and other online cell groups, create an encouraging support network for people to self-teach themselves subjects. Having a group that is potentially available 24 hours [the online website] as well as a physical meet up[the monthly meetings] provide a safety network and encouraging atmosphere to learn and work in.

d. Year of Code

Adrian Short's criticism of the *Year of Code's* intentions as a Neo-Liberalist agenda, throw some light on some intentions as to why we should teach young children to be coders and developers. In his article *"The Year of Code's Neo Liberal Agenda",* from February 6th 2012, we can see that while he is in favour of encouraging everyone to learn how to program, it should not be for blind reasons.

As an example of the neglect *Year of Code* promotes, he cites a Newsnight report in which one of the Newsnight team journalists interviews some of the children as they are working on building programs in the classroom.

The interviewer asks the child *"Why do you need to know all that stuff do you think?"* (*"that stuff"* is referring to the code on screen). The child responds with a hesitant answer that suggests a vague description of the future, *"…because when you're older you're going to think you might need coding for your work, like say if you were a banker you needed coding to do the banks."* [Short, 2012]

What Short is trying to stress is that it is not enough just to educate children in the logic of programming language, the fundamental breakdown of how computers and other technology works, and the syntactic structure of code. He is concerned that initiatives like 'Year of Code' are forgetting the most important reason for learning how to program, and that is 'why' you should learn how to program.

e. Young Rewired State

Young Rewired State, was set up in 2009 by Google executives to introduce government data to young, UK based programmers. Created for kids and teenagers aged 18 and under who are self-taught coders, developers and designers, the initiative was established to provide a network between young programmers in the UK. The aim of the network is to proved government data and information to these young programmers, so that they may create apps, websites and software to solve real world challenges.

> *"Throughout the rest of the year we invite all of the young people to attend a variety of events, often ones where they can contribute to solving civic problems through code, or where they can mentor and teach other young people or adults."* [Rewired State Ltd, 2014]

One of the founders and CEO's of *YRS* is Emma Mulqueeny. On the 8th of February 2014, Mulqueeny publically announced her resignation from the *Year of Code* board of directors in a scathing blog post on her personal site.

Her main objection was that the *Year of Code* project was not properly thought out and it excluded the people already working to improve the conditions of young programmers in the UK. She states in her blog post that

> *"I get cross emails, dms, tweets, calls from people saying why did I not include them… not me,*
> *I just called Rohan out after I caught wind of this, under a week before it happened.*
> *I cannot do anything."* [Mulqueeny, 2014a]

In her Guardian Online article from the 12th of February 2014, Georgina Voss states quite clearly the underpinning flaw in the *Year of Code* initiative:

> *"Critics of the Year of Code scheme noted that, despite George Osborne's entreaty of the personal and social good of education, the program seemed targeted more towards industry needs and a rhetoric of economic growth as highlighted by the fact that the majority of the advisory board came from the private sector and, like Lottie Dexter, many had no technical backgrounds themselves."*
> [Voss, 2014]

An initiative created and run by industry professionals, who themselves do not know how to programme or the basics of computational logic, are trying to encourage kids to programme for the sake of getting jobs in their industry. This reeks of the perpetuated neo-liberalist agenda that Adrian Short detailed in his blog post from the 2012 *Year of Code.*

Voss uses the comparison of mining children from the industrial period in Britain to compare with the intentions of the CEOs and PR people from *Year of Code.* This dark comparison throws light on why the intentions

of the industry may not have the best interests of the children and students in mind.

It is evident that there is a counterbalancing force affecting the intentions the industry is using to drive kids to learn computational skills. Whether or not this is an important factor in creating a new pedagogy for computational programming in primary school is unclear. It is however true to say that the fundamental politics and socio-politics surrounding computers, technology and computational literacy should be openly discussed in schools (see *Milestone 4*, in chapter one part two *How We Learn to Programme*) to allow the student the right to choose their own intentions for learning this skill.

Conclusion

> *"The history of computer media so far has been not about arriving at some standardized language – but rather about the gradual expansion of uses, techniques and possibilities."* [Manovich, 2013, pg 93]

In the introductory chapter of this thesis, we met with Uta Frith's three stages of early-learning literacy development. Through the careful analysis of these stages, followed by a breakdown of the current practises of online based computational logic and coding tutorials, we deduced a set of milestones.

These milestones are a deconstruction of the different reference points a new pedagogical system for teaching computational logic that should be addressed.

They are articulated as basic units of a course structure that should beging to be tested and implimented within schools in the coming years, via a wide range of teaching techniques.

Using a method of pedagogy to implement these milestones like the Constructivist approach, may lead to children adapting to the already personal pedagogy of computer programming today – the solo work of learning from online tutorials and teaching oneself how to code by building websites, apps, software and algorithms in one's spare time. A self-motivated learning curve that is encouraged and improved by formal education could create a competitive and productive drive in a generation.

The importance of teaching children to be computer literate as coders, theoriticians, as well as users will have an impact of the thought process of many professions in the future. While it is still only speculation at this time, we can imagine what a future of a totally computer literate generation would looke like, through the lens of Science Fiction films, books and television shows.

As the aforementioned Milestones suggest, 'computer literacy' is not just about beging able to use a computer via a graphic user interface (GUI). It is about being able to build, repair, expand and manipulate data in the form of software and hardware as one wishes.

If we speculate for a moment the impacts this will have on all future socio-political, environmental, medical and scientific endeavours, we can only begin to comprehend the hugh changes it would make to society, humanity and nature.

> *"The ability to read a medium means you can access materials and tools generated by others.*
> *The ability to write in a medium means you can generate materials and tools for others.*
> *You must have both to be literate."* [Manovich, 2013, pg 103]

In conclusion, this dissertation began with *what* is to be done about this inefficient design work that surrounds us, and evolved through *why* should we learn to program into *how* to go about learning and teaching programming.
Ending this process with *when* should humanity finally become computer literate as a generation and face into some of the most challenging problems we now face as an over-populated, sickly, socially divided, weather beaten and war-savaged planet.

Appendix

Illustrations / Images

FIG 1

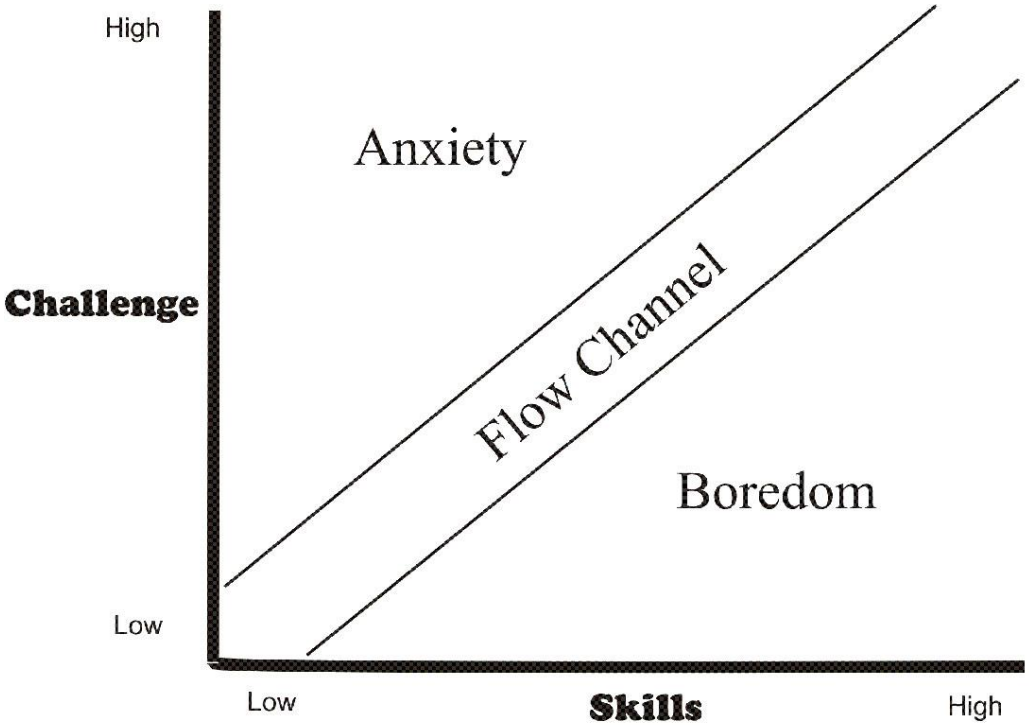
Title: ENDERS_GAME_ASH_THORP_WEB_128_906

Available at: http://ashthorp.com/ENDER-S-GAME

FIG 2

Title: fakeui.tumblr.com_Ender's Game Typing

Available at: fakeui.tumblr.com



The Flow. After Mihaly Csikszentmihalyi, *The Flow (1990)*, p. 74

FIG 3

Title: 081014_whyplay_flow

Available at:

http://www.gamecareerguide.com/db_area/images/item_images/081014/081014_whyplay_flow.jpg