the CRITICAL complexes of COMPUTING

(an ANIMATED DOCUMENTARY of the INTERNET)

*The Penguin Dictionary of Critical Theory* defines the goal of critical theory as follows: “to preclude the emergence of...an 'administered' modern industrial society which has such ideological control over the deepest desires and feelings of its subjects that they are quite literally unaware of their exploitation, frustration and unhappiness... by demonstrating that a transition to a freer and more fulfilling society is objectively or theoretically possible, and then by demonstrating that the existing state of society is so unsatisfatory and frustrating that it *ought* to be transformed.”[[1]](#footnote-1) The term was born with the Institute for Social Research at Frankfurt in the early 20th century, idiomatically 'the Frankfurt School,'[[2]](#footnote-2) but the terminology has since extended its significance, and now refers to a broader range of schools of thought.[[3]](#footnote-3)

What constitutes the 'critical' aspect of critical theory is contentiously debated by terminological fiendsters across various humanities departments. The definition of 'critical discourse' employed in this thesis references the 'critical' aspect of 'critical theory'. Critical discourse can be qualified by three distinct dimensions:

1. The discourse **details and reveals the technical operations of some socio-political apparatus** for a technically non-specialized, but otherwise interested and democratically consistent audience.[[4]](#footnote-4) These details would otherwise be opaque to all but a specialized constituency.
2. The discourse **further details and reveals what is at stake socio-politcally in these operations**. What impact do they bear on political subjectivity and agency? Which bodies are inflected, violated, and/or erased through them?
3. The discourse **generates a space for further examination and critique of the operations in question**. In other words, the discourse is a discourse: not a hermetically foreclosed matter-of-fact.

Critical discourse makes visible, politicizes, and enables critique of the operations that simultaneously conceptualize and ratify the society in which we live. As can be seen in the increasingly evident muscle of computing systems in societal anatomy, computing substance in the world--algorithms, software, hardware, and so forth--is replete with operations that demand critical scrutiny.

One institutional reflex in response to the growing strength, impact, and tangible abuse of society's technological muscle is the genesis of (inter)disciplinary formations such as computer ethics and technology policy. Princeton's version of this is the Center for Information and Technology Policy (CITP), which sits at the intersection of Computer Science, Economics, Politics, Engineering, Sociology and the Woodrow Wilson School of Public and International Affairs.[[5]](#footnote-5) CITP houses important research and space for discussion, on topics such as national and subnational surveillance, the ethics of computer science research, and the policy and implementation of artificially intelligent systems.[[6]](#footnote-6) This is undoubtedly a space that fosters critical discourse on computing in society, and this project’s initiative is conceptually aligned with CITP's ideological imperatives. A significant portion of it, in fact, is to indicate a few perceived oversights in the Princeton Computer Science department’s pedagogy of computing—and to make some suggestion towards reform about a more critical pedagogy. Ethics in computer science pedagogy at Princeton is currently treated at best as an afterthought, rather than as a systemically incorporated aspect of teaching computer science. This is a pedagogical deficiency of which many CITP and CS faculty and students are aware, and there is an active effort to incorporate ethics into course syllabi in the computer science department.[[7]](#footnote-7)

This project is not a curriculum of ethics for the CS department, but it is designed in the same space as CITP and like-minded initiatives. It looks to delimit the space of technical consideration by providing an interface between computer scientists and critical citizens and subjects of other denominations. This interface works both ways. On the one hand, it opens a space where non-specialists can understand, and thus more shrewdly critique, the computer-technical operations at work in societal infrastructure. On the other, it exposes computer scientists and other specialists to this critique: it opens a discursive space where constructive inter- and trans-disciplinary conversation can take place.[[8]](#footnote-8) Part II of this thesis will more clearly articulate the advantage of a poly-disciplinary discursive space, by example, with respect to the architecture of the Internet.

Part I of this thesis is dedicated to examining in closer detail the ethical and, I argue, sometimes *un*ethical complexes that constitute computing and its pedagogy in the academy. This analysis draws heavily on my experience as an undergraduate majoring in Computer Science at Princeton, and in this respect it is quite particular (and perhaps even ethnographic). Though this study has no pretensions as a comprehensive critique of computing pedagogy in the U.S., I do believe that much of the ideological architecture at a place like Princeton might be symptomatic of the computing pedagogy and culture that manifests in many other American institutional contexts. Though this thesis is being submitted for an undergraduate degree in Computer Science, I somewhat paradoxically present this work not as Computer Science per se, but as a window or interface to the discipline. With it, I am interested both in making some of the technical specificities with which CS grapples available to a poly-disciplinary discursive space; and in doing so, I hope, beginning to articulate something of what a methodology rigidly bounded as ‘Computer Science’ might miss.

In Part II, I theorize narrative 3-D animation as a documentary mode for computing infrastructure that proposes to make visible, politicize, and enable critique of the architecture of digital infrastructures in a poly-disciplinary forum. This methodology uses Ivor Goodson's and Scherto Gill's theoretical framework in their 2014 book, *Critical Narrative as Pedagogy* to theorize narrative as a presentational mode that generates critical discourse with respect to technical operations in computing for a non-specialized audience. To demonstrate how this methodology might work, I present a 3-minute film that presents the architecture of the Internet, the world's most ubiquitous and pre-eminent technological infrastructure.

More specifically, the film narrates the journey of a data packet sent from a cellphone in Madrid, Spain to a Facebook server in North Carolina, United States from the 'first-person' perspective of the data packet. The film's critical intention is to detail that the Internet, though often thought as 'the cloud' through Mircosoft's commercial rhetoric in the early 2000s that gave rise to that conception, is actually enabled by a global, physical infrastructure of cables, routers, and other hardware that are subject to material and geopolitical exigencies. I intend the film to represent a concrete step towards a computing culture in which more parties can participate, and thus ultimately one that is more democratic. My hope is that this thesis, which is self-consciously an experiment in computing pedagogy, offers a contribution towards how we might foster and refine the discourse regarding what is at stake in the Internet's socio-political operations—for us as political subjects, as agents, and as bodies.

PART I

PART II

The representation of computer science concepts in 3-D narrative is, to the best of my knowledge, a new kind of project. There are many video explanations of the Internet, many with descriptive animation, but all of those that I have seen are from the ‘classroom’ perspective: the video explains how the Internet works as if a teacher telling a student. 3-D representation is increasingly being used to represent complex architectures in interactive presentation. Forward-thinking real estate firms now offer 3-D models of homes online[[9]](#footnote-9), and 3-D representation is used in all walks of scientific display, from child-friendly museum exhibits to cutting-edge engineering initiatives.[[10]](#footnote-10) With the advent of advanced graphics technologies such as virtual and augmented reality, 3-D has become a sort of representational currency of the future.

Interactive 3-D models can make for compelling pedagogical tools, as they describe an object from many perspectives, and often give a much clearer sense of the object than a 2-D equivalent. A 3-D model can also, however, be a overwhelming place to start. Which annotations should I read first? How can I best understand this object in a given lesson time, or in the two minutes I am choosing to look at this exhibit?

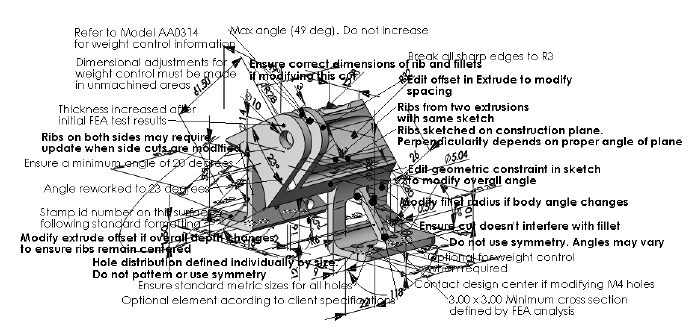


Figure 1. *Visual Clutter*.

What the interactive 3-D model often lacks as a pedagogical experience is *narrative*. A wealth of data is on display, but there is very little sense of how a newcomer might interpret it. The narrative method of 3-D representation that I am proposing, therefore, is something closer to ‘documentary’ than ‘museum exhibit’. The object of study is represented not as a plain archive of facts, but rather as a factual world that is psychologically cohered through the constructed figure of a story.

Computer scientists become uncomfortable when one begins to talk about narrative, as there is a preference for more ‘objective’ representative modes in the labor of depicting data, a bias inherited from its scientific pretexts. As Barbara Klinger notes, however: “Scientists, it turns out, are also storytellers.”[[11]](#footnote-11) There is no use case for a fact unless it is contextualized in the broader narratives of social life. At the beginning of each and every paper, scientists tell a story to justify their research as interesting and important. Scientists deal with data, and from these data they suppose to extract facts: but facts are simply dollars in the broader currency of narrative. Narrative presentation and context is what gives the facts value.

Humanistic disciplines such as history and anthropology are often juxtaposed against so-called ‘hard’ sciences such as biology and chemistry in their treatment of fact, as the former are know to favor narrative as an epistemology over scientific positivism. Since the publications of critiques of scientific credibility from theorists such as Bruno Latour and Steven Shapin, it has been clear that a factical hierarchy between science and the humanities is misguided. What is published as ‘science’ is never a pure function of data, as the data always at least suffers observation and interpretation. Science is produced through a socio-cultural apparatus, and involves many subjectives despite its pretension to objectivity. As Shapin has argued, it is important to scrutinize the apparatus through which science is presented, as what comes to be considered ‘truth’ is not hermetically sealed: it has an architecture.[[12]](#footnote-12)

The architecture of social truth has a history as long as society, and much of it is characterized by narrative as a dominant mode of representation. As Edward Said, a respected anthropological theorist, notes:

Stories are the heart of what explorers and novelists say about strange regions of the world; they also become the method colonized people use to assert their own identity and the existence of their own history. The main battle in imperialism is over land, of course; but when it came to who owned the land, who had the right to settle and work on it, who kept it going, who won it back, and now who plans its future—these issues were reflected, contested and even for a time decided in narrative.[[13]](#footnote-13)

The structural role that narrative plays in the architecture of truth is increasingly evident in contemporary politics and society. Trace some examples, politics as theatre, stories as the currency of fact, proliferation of fake news.

The scientific reflex to the epistemological instability of contemporary politics is to assert that scientific facts are the superior kind of fact, and that the problem is that so many heretically contravene this fact. When this reflex is transposed to a politic, however, it turns out to be broadly ineffective, as it does not take into account any notion of social truth as an architecture. A building’s renovation is a reconstruction, not an instant replacement. In looking to re-architect social truth, narrative is an extremely powerful design methodology: it is an *essential* mechanism of information transfer in the social and political sphere.

To be more concrete, what I am suggesting is that, like science, computer science has a discourse problem. None would deny that its research has far-reaching implications for societal infrastructure, and as such it is becoming increasingly clear that its mechanisms, both technical and methodological, need to be scrutinizable beyond the hermetic bubble of technologists. The way we will make these mechanisms available—though it risks sounding like a Silicon Valley entrepreneur at a TED talk—is by telling better stories about them.

In computer science, one could say that stories are everywhere. They are employed at the surface of every application to make complex operations available to the user. The way that stories function in the way that we understand and use software is profiled succinctly by a popular agile[[14]](#footnote-14) software development methodology, called *story-driven modeling*. From the Wikipedia page: “Instead of abstract static structures, story-driven modeling focuses on concrete example scenarios and on how the steps of the example scenarios may be represented as object diagrams, and how these object diagrams evolve during scenario execution.”[[15]](#footnote-15) The technique employs hypothetical scenarios in which code will be executed, and uses these scenarios as a specification for the program. In other words, the software designer uses a collection of sub-narratives as units of design to compose what could be considered the ‘grand narrative’ of the software at large, the set of ways in which it is designed to be used. As with narrative in books, there must exist threads of continuity between sub-narratives in order to construct a coherent grand narrative.

The technique of using stories to think computational logic permeates developers’ thinking about software at the level of code. Even though code is a brand of interface that is often discussed as if it is cold, hard logic, story-driven modeling is a predominant currency of understanding when (good) developers think computational operations. More generally, it would not be absurd to claim that *every* technical understanding is a little fiction spun about the operations being performed at the hardware level of 0s and 1s. An operating system is one big narrative that is composed of many sub-narratives to make possible higher level concepts such as ‘data and ‘functions’ at the level of software applications. A software program such as Microsoft Word is a grand falsification of what is actually going on under the hood so as to make meaningful operations—such as writing a book in a document—possible. Any coherent description of these operations is a necessary ‘narrativization’ of its intricacies. And this is a feature rather than a bug: the stories we tell at the different levels of computer and programming language architecture are what enable the many complex applications we see in computing. Narrative (often called ‘abstraction’ in Computer Science) is what makes 0s and 1s useful and meaningful.

This is a familiar gesture at the level of commercial software applications. A common framework in marketing for instructing about how a software ought to be used is by way of example, through what people in user experience (UX) call *user narratives*. These are stories about how a given application is used in the ebb and flow of a user’s daily life, which signal to other potential users why they might want to use the application.[[16]](#footnote-16) User narratives are simultaneously used in the development lifecycle of an application: they work both as a marketing figure, and as the stories in story-driven development.

This points to a general use-case of stories in software: they pedagogically compose the interface between executable computational logic and user understanding. Indeed, if your philosophical inclinations allow it, narrative might be said to exist as the sole currency of understanding computational logic. (I think there is a case for this; but it is not crucial to my argument here, so I leave it open-ended.)

It is clear from recent literature is computer science pedagogy that narrative has *some* currency. Narrative as CS pedagogy. The proliferation of stories as a pedagogical figure in teaching CS.

The argument that narrative does effective pedagogical work for computational logic in software concurs with the way that narrative operates across a range of other factical domains. Indeed, the act of effectively representing factical matter through narrative constructs is formalized as a genre, namely *documentary*. Brief paragraph introducing/connecting documentary. Points from below, narrative’s implication in political/legal structures.

Computer Science needs representation beyond code; it needs documentary of its processes and methodologies.

This project’s intention is to compellingly present Internet architecture at a high level via narrative’s abstraction; not to comprehensively detail its complexities. A fetishized take on the Internet is a useful building block of a more holistic understanding.

**WHY FILM?**

The question is rightfully put: why not some other documentary mode of representation? Though it is hopefully by now clear that computational logic and computer science need some ulterior documentary representation to be brought into political and social discourse, there is no obvious predilection for film as its carrier.

**WHY NARRATIVE**

Storytelling is acknowledged as an effective methodology in computing culture.

* Storytelling through Agile development, ‘user stories’.
* Increasing call for ‘narrative’ pedagogies in teaching software. Gamified resources for learning how to code, etc. CS literature here, could review more.
* Anthropological literature regarding narrative (and Goodson&Gill)—stories help us to think the human in situations. Indeed, this is the same impulse that both agile stories and narrative pedagogies for teaching software respond to.
* Documentary film-making, the cultural history that looks to animate data through narrative frameworks.
  + Edward Said, *Culture and Imperialism* 1994: “stories are the heart of what explorers and novelists say about strange regions of the world; they also become the method colonized people use to assert their own identity and the existence of their own history. The main battle in imperialism is over land, of course; but when it came to who owned the land, who had the right to settle and work on it, who kept it going, who won it back, and now who plans its future—these issues were reflected, contested and even for a time decided in narrative.” pp.xiii
  + In fact, they are still decided by narrative: c.f. legal narrativity. The construction of the national subject in a court of law is still formed and defended around narratives qualitatively comparable to those persuasionist tactics used in Greek courts to appeal to the sympathies of the democratic ballots (c.f. high school Classics texts). The narrative status of the legal subject is highlighted in Forensic Architecture’s work: c.f. Keenan’s essay, and the role that racialized narratives play in the performance of legal persecution, white supremacist shooting. *Narrative’s legal legibility in democratic governmentality*.
* Which leads us (ish) to Pixar’s and Disney’s prerogative: to literally animate data through narrative frameworks.
  + Pixar literature, Herhuth’s **aesthetic storytelling**.

**WHY FILM**

Many experimental interactive teaching resources fall through, as they ride a wave of kitsch that permeates all new forms of technology.

* There is a media-cultural legitimization that takes place with new technologies
  + Theatre: still very little legibility (through Plato’s anti-theatrical bias)
  + Film is an established form of representation, particularly pertinent to documentary.
* *Documentary*: this genre contains a media-specificity in our cultural appropriation; it is filmic. When thinking about making a documentary of the Internet, the question becomes: how to legitimately portray the Internet when its fundamental concepts have no materiality; at least, not in the same way that we can show the materiality of a phenomenon (c.f. *Fuoco Ammare*).
  + Doc as a genre has looked to challenge the status of narrative, and in particular the character’s constitutive role in narrative—but we have come to a place where documentary often takes place through a character, the logic being that a human fact to the facts humanizes them. c.f. Tret’iakov, *Biography of the Object*: and the Soviet factographers more generally.
  + Anthropomorphization disfigures, but it also subjectivizes the reality of digital infrastructures that is so often presented as static and objective.
  + Doc film theory? Sheila Curran Bernard, *Documentary Storytelling: Making Stronger and More Dramatic Nonfiction Films*, 2nd edition (Focal Press, 2007) – “Documentaries bring viewers into new worlds and experiences through the presentation of factual information about real people, places, and events…. But factuality alone does not define documentary films; it’s what the filmmaker does with those factual elements, weaving them into an overall narrative structure that strives to be as compelling as it is truthful…. Story is the device that enables this arrangement.” (from the introduction, 1st para).
  + **Stressing the importance of story --** “How to tell a story” is the first chapter of Peter Gabriel’s *Practical Guide for Activists*. Part One of Bernard’s book is titled “Story Design.”
  + Animation has been used since Disney commissioned in 1942-45 for Franc Capra’s *Why We Fight*. Now full animated documentaries (2008: *Waltz with Bashir*).
    - **This in combination with Herhuth, for animated film’s reality disfigurement/critical potential**.
* To recap and summarize: animated film, an emerging political legitimacy.
  + Allows us to look at the architecture (and consequently politics) of something that is not actually visible.
  + Particularly relevant to software complexes, as software architecture is *spatial*.
  + Is particularly punctual in keeping from code’s frightening syntax.
    - In CS classes, you are taught architectural paradigms, not code. The code is the implementation of the architectural paradigms (and utterly important; the building processes)—but as an intellectually engaged observer, you don’t need to know everything about the building to appreciate the political figurations in which the architecture is implicated, or implicates its residents.
    - In this sense too, can be both a tool for young viewers, but also theoretically interesting and engaging across many generations and areas of expertise.

**WHY 3D**

More room for complex representation.

Barbara Klinger in *Cave of Forgotten Dreams*, "Scientists, it turns out, are also storytellers." pp.993. Also, “3D explicitly reminds us that there is no naturalism without spectacle and that this spectacle and its pleasures have long derived from the techniques and technologies on display.” pp.992

“3D films tend to expose the relation between technology/science and the art that subtends all fimmaking” pp.993 – she is talking about 3D-viewing, but the commentary extends to 3D animated space. There is no way this could have been done without a certain cadre of technology.

*Blurring the boundary between binaries, 3D’s effect*.

* Spatial represenation in software, and the way this can be made available through 3-D. Explicitly putting architectural paradigms in softfware rhetoric in touch with architecture as a physical consideration of design, etc.
* Ability to represent complex structures in a diagrammatic paradigm, and combine it with narrative.
* Reusable and open source models, iterative corrections, re-animations possible.
* *Forensic Architecture*, shows the critical potential of 3D? (maybe unnecessary)

Also a fun potential of reusing assets in VR, AR. 3-D is becoming hip.

**ANIMATED DOCUMENTARY OF COMPUTING INFRASTRUCTURE**

A dialect of critical computing: making complex infrastructural(ist, c.f. Durham Peters) concepts available to an audience that is not code nor algorithmically literate.

PART III

process

Creating a 3-D animated short, it turns out, is no trivial project. There are multiple steep learning curves to scale. To name a few: the multi-faceted interface of a 3-D animation suite, story construction, animation technique, character realism, lighting theory, rendering softwares and techniques, 2-D post-production, other contingencies of film production such as audio—and on top of this, practicing and refining a data-intensive workflow and pipeline that requires a close attention to folder organization, file types, storage capacity, version control, and keeping backups in case of computer meltdown. The process of learning how to make a 3-D animated film, for me, has been distinctly comparable to learning all the technologies in a software product—front end, back end, devops, marketing, agile development—all of which are in distinctly unfamiliar languages.

In order to make progress on the task at hand, it has been imperative to schedule, prioritize the task at hand—and most importantly, to take shortcuts wherever they are available. It is all too easy to get lost down a rabbit hole reading about technical domains that are not at all relevant to the short one is trying to make. The world of 3-D animation, like so many other software parishes, is vast and pregnant with marvelous fissures of techno-religious reverie that will quickly forget a project’s progress in favor of these temporary happinesses.

At places like Pixar, 3-D animated films and shorts are produced by a creative-industrial pipeline. Before any animation is done, a film’s concept is worked, painted, and pictured in depth. Similar to the way an application is often storyboarded and designed in a visual suite such as *Photoshop* or *Adobe XD* before any specifications are handed to a software developer, the majority of the film’s contours—the plot, the look, color and feel of characters, the color palette (called a ‘color script’[[17]](#footnote-17))—is refined by a team of artists.[[18]](#footnote-18) A plot and script goes through many stages of revision, a lengthy creative process that is non-linear with technical animation; though a bulk of artistic work is done up front to set the scene, artistic refinement continues in conversation with the technical animation process. Designing a compelling story is a deeply subjective task, and Pixar is famous for the importance it attributes to this aspect of its animation practice. John Lasseter is the pioneer of this art at Pixar, and for many it is Pixar’s commitment to story that keeps them at the forefront of creative-industrial animation.[[19]](#footnote-19)

What makes Pixar an animation studio, however, is its animation. Ed Catmull, the technical mastermind who founded Pixar along with Lasseter (and famously Steve Jobs) is considered by many the father of creative animation, as many of the technical breakthroughs that made 3-D animated film possible are attributable to him.[[20]](#footnote-20) 3-D animation science has made extraordinary bounds since its genesis in the late 20th century; but the process of creating an animated short is still an extraordinarily labor-intensive one.[[21]](#footnote-21)

The approach I took to creating *The Internet: A Spatial Story* is detailed below. Much of the process was ad hoc, as Princeton has very little in the way of faculty who can advise in both the practicalities of 3-D animation and/or creative animation methodologies.[[22]](#footnote-22) The backbone of this process was derived from Kenny Roy’s excellent book, *Finish Your Film: Tips and Tricks for Making an Animated Short in Maya*.I would, however, like to thank Tim Szetela, Adam Finkelstein, and Nora Willet, whose helpful introductory pointers steered me in the right research directions early on. I would also like to thank Andrew Gordon and James Cunningham.

LEARNING MAYA

In addition to *Finish Your Film,* my working knowledge of *Maya*, *Photoshop*, *After Effects*, *Substance Painter*, *Substance Designer*, and the various other applications I used to render the short is derived from several other books, and countless sources on the Internet: *Lynda.com* tutorials, YouTube videos, Maya documentation, blog posts, and many other sources.

It would be futile to attempt to cite every video or blog that I read in the process of creating this short; but some of the more comprehensive books, blogs, and videos that I found consistently useful are listed in the bibliography. A lot of learning was done up front in order to understand the principles of making a 3-D animated short, lighting, and other germane technics. Though I started the project in August 2018, I didn’t start working on assets for the film per se until January, 2018. Much more of the learning, however, was done dynamically. As I was attempting to produce I scene, I would realize that I did not know how to do something, and so I would take a few days to research theory and tools, before returning to the task at hand with a greater understanding of Maya’s toolkit. In a later section, I detail some of the more interesting technical difficulties I encountered during the short, and the path taken at these crossroads.

STORYBOARD

In 3-D animation especially, the storyboard of a film is where almost all of the narrative innovation, and much of the aesthetic innovation occurs. Drawing a parallel to an application development software workflow is again instructive here. In order to make an application that solves a real problem and has relevant design, it is important for a designer to storyboard in quite specific detail that functionality the app before handing the specifications to a developer. This is because what is good design is consistently not the same thing as what is easy to implement technically. If a developer tries to dynamically design an application concurrently with its implementation in code, good design will often be comprimised in favor of facile implementation on the development side.

This is a truism that holds not just in software engineering, but also across many other physical engineering and building practices. The practical development of anything is a tradeoff between what is ideal and what is practical. If there is no ideal (i.e., no designed specifications for the project), then what is practical in stages of construction will dominate the architecture of the system. There are countless cases in engineering history, software and otherwise, in which a lack of emphasis on design in favor of practicality has led to faulty architecture—sometimes with dire, even fatal, consequences.

The storyboard of a 3-D animated short is its design specification. The preproduction and production phases of the process (detailed below) are the equivalent of the engineering phase in the analogy above: they are the implementation of a design specification through technical means.

ANIMATIC

3-D PREPRODUCTION

3-D PRODUCTION

ANIMATION

RENDERING

POST-PRODUCTION

ASSET PREPARATION

To come:

* The remainder or Part I, as proposed in the first four pages above.
* The detailed and theorized outline of 3-D animation as critical computing pedagogy. This will discuss why animation is compelling pedagogically, why narrative is the preferred methodology, and why 3-D space is a fertile medium for representing complex technological architectures that has not yet been adequately practiced or explored.
* A once-over on 3-D animation in industry structures such as Pixar and Disney, and a comparative framing of my approach.
* A reading of my own short with respect to its critical potential (as located in the three dimensions outlined in the first section of this draft).
* A ‘further work’ section that explains what the specific fruits of this thesis’ labour are, how they can be extended and/or reworked, and what other forms of critical computing might arise.

BACKGROUND

It has been my intention to construct a project that investigates computing infrastructure with a critical interface since at least January 2017. While taking pause from school back home in New Zealand, I started collecting notes towards a critical history of the web browser: starting with Tim Berners Lee’s inaugural *WorldWideWeb* in 1989, the browser that was coevally released with the Internet itself, through the browser’s commercialization by Netscape in the early 90s, the browser wars at the turn of the century, and the highly corporatized ‘free’ nature of the web browser today.

After some research, it was somewhat astonshing to me that, though there exist many histories of the Internet,[[23]](#footnote-23) there seems to be no book-length history of the web browser available. The history exists only as a Wikipedia page, and is also collected in part as chapters of these histories of the Internet, between software development manuals, corporate retrospectives and other technical documentation such as Internet RFCs; but has not yet been readily collected in one text for the critically engaged reader who is not necessarily a computer scientist.[[24]](#footnote-24)

During a PIIRS Research Fellowship in Berlin in the summer later that same year, I worked to design a project for my thesis that would present Internet architecture in a form that might prove interpretable to my peers and colleagues in humanities departments, and to a more general public. It was clear from conversations that there was a general desire to learn more about how the Internet worked; but, not knowing how to code, they considered much of the widely available technical documentation (such as Stanford’s whitepaper[[25]](#footnote-25)) somewhat inaccessible (see pp.xxx in Part I regarding the hyper-disciplinary gap to Computer Science.)

I iteratively researched (and in some cases began implementing) a range of different projects with this in mind. Some of these false starts are outlined here:

* A ‘wayback’ browser emulator through which to explore the development of Internet aesthetics from the 90s to the present.
* A stand-alone browser that would explain parts of the Internet as it happened, instead of displaying HTML and CSS in their conventional attire.
* A proxy web application that uses a headless browser under the hood to retrieve websites and ‘annotate’ them to explain how parts of them work.

While these projects remain interesting to me, I came to realize that their ability to reach a non-technical audience might be limited. More likely, I would create another poorly maintained web application that doesn’t do quite do what it promises, and is of dubious use to anyone.

In the course of keeping up with web development while working on these projects, I was reading Lin Clark’s ‘code cartoons’ on WebAssembly and Mozilla’s new *Quantum* browser engine.[[26]](#footnote-26) Clark’s blog posts deep-dive into technical intricacies of parts of the Firefox browser, illustratively using cartoons to clearly demonstrate what is happening in an algorithm, or some other programming construct. As concepts in computer science tend to be spatial (there is a reason we talk about computer and software *architecture* rather than computer *literature*), cartoons and diagrams are a principal pedagogical tool.[[27]](#footnote-27) Even if one doesn’t understand the semantic convolutions of a given language, it is almost always possible to understand what is happening conceptually by looking at a diagrammatic representation of the algorithm. Thinking these false starts and code cartoon as pedagogy together led me to thinking that the best way to detail the architecture of the Internet was not necessarily through an interactive structure, but rather first and foremost through an analogue of this diagrammatic pedagogical paradigm.

1. David Macey, *The Penguin Dictionary of Critical Theory.* Penguin Books, 2002. pp.75-76. [↑](#footnote-ref-1)
2. See—Jay, Martin. *The dialectical imagination: a history of the Frankfurt School and the Institute of Social Research 1923-1950*. Berkeley: University of California Press, 2008—for a sound intellectual history. [↑](#footnote-ref-2)
3. "critical, adj.". Definition 3c. OED Online. January 2018. Oxford University Press. http://www.oed.com/view/Entry/44592?redirectedFrom=critical (accessed March 02, 2018). [↑](#footnote-ref-3)
4. 'Democratically consistent' here indicates that there should be no anti-egalitarian qualifications for this audience; economic, racial, gendered, or otherwise. The Brazilian theorist Paulo Freire articulates the egalitarian labour that is contingent to critical thinking in his eminent 1970 work, *The Pedagogy of the Oppressed* (Freire, Paulo. *Pedagogy of the oppressed*. Penguin Group, 1996). As Scherto Gill notes in his 2014 essay *Reframing the Critical*: Friere's conception of the critical investigation "in turn helps men and women develop a critical form of thinking about their world, through dialgoue. The dialogic approach to education is not to impose or convey any existing views, but rather to explore shared understanding about the world. Part of the dialogue is to enable the oppressed to identify that the oppressor has been 'housed' in the oppressed; consequently, the oppressed is a dual being, as he/she continues to carry the oppressor within himself/herself." (Goodson, Ivor, and Scherto Gill. *Critical Narrative as Pedagogy*. “Reframing the Critical.” New York, NY: Bloomsbury Academic, 2014.) [↑](#footnote-ref-4)
5. <https://citp.princeton.edu/about/>, accessed 3/2/18.. [↑](#footnote-ref-5)
6. <https://citp.princeton.edu/events>, accessed 3/2/18. [↑](#footnote-ref-6)
7. c.f. Jasmine Peled's thesis workshop on ethics, Jonathan Lu's seminar last semester, *Society and Ethics of Computer Science*, Princeton AI4ALL, <http://ai4all.princeton.edu>. [↑](#footnote-ref-7)
8. need citation? surely interdisciplinarity is established enough now... [↑](#footnote-ref-8)
9. Tiny Home models. [↑](#footnote-ref-9)
10. Rolls Royce Submarines uses a 3-D model of a nuclear reactor, etc [↑](#footnote-ref-10)
11. Barbara Klinger, *Cave of Forgotten Dreams*. pp.993. [↑](#footnote-ref-11)
12. Steven Shapin, *A Social History of Truth*. See also *The Architecture of Public Truth*, ed. Forensic Architecture. [↑](#footnote-ref-12)
13. Edward Said, *Culture and Imperialism*. 1994. pp.xiii [↑](#footnote-ref-13)
14. ‘Agile’ software development refers to a set of practices for collaboratively coding applications that solve user problems. Aspects of this methodology are spread across much coding practice. See <https://softwareengineering.stackexchange.com/questions/11512/are-there-any-major-alternatives-to-waterfall-and-agile> for alternatives. [↑](#footnote-ref-14)
15. <https://en.wikipedia.org/wiki/Story-driven_modeling>, accessed 4/5/18. [↑](#footnote-ref-15)
16. <https://uxmag.com/articles/designing-around-the-whole-story-with-user-narratives>, accessed 4/5/18. [↑](#footnote-ref-16)
17. <http://pixar-animation.weebly.com/colour-script.html>, accessed 03/25/2018. [↑](#footnote-ref-17)
18. See <http://pixar-animation.weebly.com/pixars-animation-process.html>, accessed 03/25/2018. This practice, and others referenced subsequently, was confirmed during a private tour I received of *Pixar* Animation Studios in Emeryville, California in late January, 2018. [↑](#footnote-ref-18)
19. See BIG GREEN BOOK, and ED CATMULL’S BIO. [↑](#footnote-ref-19)
20. For more on Catmull’s involvement in founding 3D animation science, see ED CATMULL’S BIO. [↑](#footnote-ref-20)
21. The release of Toy Story in 1995 marked the beginning of 3-D creative animation. See ED CATMULL’S BIO for a sketch of the history leading to this release. [↑](#footnote-ref-21)
22. Though I am critical of computing pedagogy elsewhere, the fact that animation is not taught at Princeton is not a specific criticism of mine. Though I think it would be great if Princeton taught more in the way of 2- and 3-D animation, I do not think it should necessarily be Computer Science’s purview to teach these skills (much in the same way it should not necessarily be the responsibility of Computer Science to teach the practices and methodologies of commerical software development). [↑](#footnote-ref-22)
23. For a good example, see Johnny Ryan, *A History of the Internet and the Digital Future*. London: Reaktion Books, 2013. For many others, just type ‘history of internet’ into Google. [↑](#footnote-ref-23)
24. This paper is currently incomplete, titled at present *Towards a Critical History of the Web Browser*. I am meaning to re-write and eventually publish it somewhere. [↑](#footnote-ref-24)
25. <https://web.stanford.edu/class/msande91si/www-spr04/readings/week1/InternetWhitepaper.htm>, accessed 2/3/18. [↑](#footnote-ref-25)
26. <https://hacks.mozilla.org/author/lclarkmozilla-com/>, accessed 3/2/18. [↑](#footnote-ref-26)
27. Another notable pedagogue who is known for her code cartoons is Julia Evans. See <https://jvns.ca/teach-tech-with-cartoons/>, accessed 2/3/18. [↑](#footnote-ref-27)