Ontario College of Art and Design
University

screenPerfect: the importance of accessible technology for artistic use.

Author:
Alex Leitch

Supervisor:
Dr. Emma Westecott

A thesis submitted in fulfilment of the requirements
for the degree of Masters of Design

in

Digital Futures

in the Faculty of Design

26th November 2013

Declaration of Authorship

I, Alex Leitch, declare that this thesis titled, 'screenPerfect: the importance of accessible technology for artistic use.' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a master's degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

Acknowledgements

I wish to express my appreciation of my thesis advisors, Emma Westecott and Simone Jones, without whose generous contribution of time and sensible advice this would be a much weaker paper. I would also like to express gratitude to the Site 3 coLaboratory community, who provided me space to work and a community to work with, and Bento Miso - Dann Toliver, Jennie and Henry Faber expressly - for their technical advice and their help in hosting No Jam 2. I would also like to extend my thanks to Evan of the Digital Futures tech desk for a key tip when I was just starting my research. Hannah Epstein deserves my thanks as well for being a fantastic collaborator.

To my personal support community, many thanks for your patience with me. It's okay, it's done now. Now we can go back to conquering the world.

For Adina, and her unflagging belief in the value of the work...

Abbreviations

CYOA **C**hoose **Y**our **O**wn **A**dventure
A popular format for game narrative, describing a branched, choice-based structure, as versus a linear story.

FiG **F**eminists **i**n **G**ames
A SSHRC-funded association of digital researchers interested in disrupting gender bias in game development.

DMG **D**ames **M**aking **G**ames
A non-profit community organization based in Toronto dedicated to supporting dames interested in making, playing, and changing games.

JS **J**ava**S**cript
A scripting language, traditionally used for client-side programming on the web.

Indie **I**ndependent Developer
Indie developers are game developers not associated with tradtional, well-funded development houses.

1

Introduction: Researching A Better Engine

This is a thesis about art and technology.

The arts and the humanities are the technical name for the fields of work that produce both North America's culture and its record of its culture. We use computers to do the work, the same as people use computers to do most other kinds of well-paid brain work in 2013. Computers are about as far from a straightforward tool as it is possible to be: in 2013, a computer is not so much a hammer as it is an ongoing negotiation between software that works pretty well most of the time and hardware that is generally reliable as long as it is not presently wet.

The use of computers as tools is a specific skill set, which is as unique as the skill set of using a paintbrush. The central component of this skill set is a comfort with curiosity: the tools change all the time, and many of them are not well written. Almost all take a vast period of time to allow skill acquisition before content - art - can be produced, and this time is expensive. To require time over money is to restrict people who have little time from using contemporary tools. Within these restrictions, the number of people for whom computer use in and of itself will be a delight is a limited population. This is problematic, because art production is difficult even without the boundaries raised by software challenges. The more limited and specific the skill set required to use contemporary software tools, the more difficult it is to include a diversity of voices in the cultural production of genuinely contemporary work.

When artists are excluded from technology, culture splits on lines of privilege. There are artists who make art, and technologists, who make technology, but do not see themselves as particularly responsible for the ideas encoded in their work. Technology is not neutral. It is authored, and where there is authorship, there is a responsibility for ideas. When large groups are left out of communication media, particularly those tasked with

producing the language with which culture speaks to itself, there comes a disconnect in the public representation of our sense of self.

Video games are the newest of the cultural production engines. A good game, as described by the MDA Hunicke, LeBlanc and Zubek, 2004, first involves engaging mechanics – the loop systems of reward and scoring – then graphics to create a world, then sound to fill out that world. The controls are weighted this way and that, but the most profitable games - referred to as AAA or triple-A properties - are presently power fantasies. There are other fantasies out there than the ones at the end of a gun. The tools have been created to design gun-centric worlds, and therefore, gun-centric worlds are the ones that get built. People who are not interested in games about violence do not have ready access to other metaphors, and this restricts the voices we have allowed to produce new cultural content.

The value of a broad range of voices in any cultural practice should be self-evident, but video games are presently unique in their high-level uniformity. Unlike any other discipline, the basic tools for games tend to be expensive, and the toolsets that exist to generate games support specific mechanics in similar-seeming worlds. This means that art games are unfairly compared to commercial games produced with the same media. The commercial games set the ground of the conversation, which denies the subjectivity of the producer of the second-stage tools. These producers - developers and game designers - are disappeared into the system, their work only revealed by which elements an artist can or cannot subvert. Without cooperative tools, it is challenging to make new things.

For many years, the most important thing about new games was their graphics. We are now topping out the graphics field. New pictures are available, and will always have their audience, but as Galloway has asserted, games are not pictures. They are not films. Games are software that exists when action is taken within their rulesets, and therefore, the newest and most interesting research in games is not how to shoot prettier things, but to find new ways to shoot them. It is not the best graphics that are most important, but the finest actions, the most interesting experience for the investment.

The problem of interesting experiences is not trivial. Video games offer an economically advantageous distraction engine, a way to enact an artificial life during a period of declining general wealth. Allowing a diverse range of voices easy access to portray their own games, their own alternate or idealized modes of being, is a way of making those voices more real, of offering an alternate human experience to the “asshole simulator” Bissell, 2013 genres manufactured at much higher budgets.

1.1 Research Question

My research question is to ask which conditions make software tools more accessible, as versus more limited, to people who are new to making new media art. I am specifically limiting my research to the idea of a game engine. A game engine is a software system that allows content producers to place assets in context with programmed scripts to generate different systems of interaction. My research is in how the use of a radically simple game engine - a piece of software which is fundamentally limited - can enable a polyphony of creative expression by reducing the friction of learning new software tools.

In this paper, I ask questions of how collaborative work affects artistic production, how much influence a tool can have on an artist's practice, what sort of assumptions are made apparent to producers via the tool, and what the position of a technologist is in engendering this work.

1.2 Initial Approach

1.2.1 Collaborative Practice

Collaborative practice is somewhat difficult, as code practices around collaboration rely on open-source techniques as well as pair programming. Pair programming is a coding practice that involves partnering a more experienced and less experienced coder in order to communicate code experience and improve the legibility of the completed software.

This thesis assumes that the definition of gaming as an art form dependent on action is accurate, and therefore, after developing an initial round of software, the main body of feedback and development in this software package has been via user collaboration. To that end, I have organized a game jam in concert with a local coworking facility, to host an event where volunteers can use the tool to develop their video works. The volunteers can then give feedback to both the software developers and to each other on what is possible with the tool. At that point, I will incorporate their feedback into future versions of the tool, hopefully leading to a stable platform for interactive experience development. This is a dynamic type of user-centered design, relying heavily on version tracking software and rapid updates, in line with Agile development ideals.

The initial software underlying screenPerfect has been developed in concert with an artist who laid out an idea for how a video interaction might work, which has then been created and refined, released to more artists, and then revised again. The hope is that

each new version of the tool will generate a useful echo chamber, amplifying new ideas even as it makes advanced technology easily accessible to content producers.

The toolset can then be released and left for artists to use and analyse.

1.2.2 Code and Theory

The initial software of this project was developed as a response to the lack of privacy and control of various shared media sources online. Rather than developing for a mass audience, I began with the idea that this should be developed privately, for small audiences using disconnected technology.

The code of screenPerfect was written in Javascript on the server and client side, using Node.JS. Node is a server environment library and framework that is intended to permit web developers familiar with JS to write their code to the server. During the process of the thesis, the idea of screenPerfect as a game engine was taken up by my industry partners, Bento Miso, who are interested in the idea of new game engines as a use case for their language, Daimio **daimio** In the case of ScreenPerfect, I decided to write the initial application in Node.JS and javascript to take advantage of the speed of the Google V8 code engine. When that application was done, I turned it over to Miso, who refactored the code into something that could be attached to the Daimio language system.

The critical theory that underlies this practice is a combination of French poststructuralism - Helene Cixous in particular - and contemporary writing on video games and the history of women in technology. By producing the software and content with the input of a local feminist collective, Dames Making Games, I have grounded the work in a social justice driven practice which encourages women to take part in their own lives by learning how to interact with machines and communicate with the broader world.

1.2.3 Game Research

The Twine engine is a branching narrative engine for authoring text narratives. I have taken the idea of Twine and transformed it to use video and still images to expand the possibilities of an author experience.Twine is affordable and requires very little training to access, but it is also limiting, in that it undermines skillsets in imagemaking which may already be highly realized. Cinematographers making Twine games may find it lacking.

In addition, Twine does not take advantage of the new multi-monitor systems, which are both important and difficult to work with. ScreenPerfect is useful to both people who wish to make straightforward games, and to installers who would prefer to use multiple clients to explore the possibilities of multiple-projection screen surfaces.

1.2.4 Cyborgs, Women, Game Design

My initial approach and research method to this work is based on the Agile Manifesto, a software development methodology that opposes siloed, top-down software development. I have paired Agile development with the work of Helene Cixous, whose "Laugh of the Medusa" provided a template for ecriture feminine, an argument for women writing of their own experience in order to be made visible. I have also examined works such as Vera Frenkel's *String Games* and some of the history of conceptual video art within Canada.

This work is related to various texts of feminist and woman-oriented cybertheory that have appeared in the years since: Haraway's Cyborg Manifesto, TIQQUN's Preliminary Materials Towards A Theory of the Young-Girl. All of these are academic constructions of femininity as it is seen in relation to technology: they are feminist in the formal sense of the word. There are other senses of the term, which I will not be examining within the paper. Rather than expressing this work in context with Cixous as *écriture feminine*, I will be using Cixous as a reference for the idea of the alien perspective as a perspective of resistance within a means of expression controlled by a neutral-to-hostile majority perspective.

This approach addresses women as an alien construct to the more conventional world of technology, which has been recently associated with a masculinist performance that is unnecessary for the pure structure of good rules and the development, through that, of good software.

My initial approach is to pair with an artist who had a game idea, take that idea, and then make it reproducible. Reproduction is, after all, the province of cyborgs: we control our biological systems, and in doing so, we have conquered what was once a hard-built destiny. By making the consequences of biological sex into a more pliant construct of gender, we have transformed what we must be to what we might be.

This work has been difficult, but I believe it to be important, not so much for the software itself - a proof of concept - but because it is important to provide software that permits people to access new technology. I am not alone in thinking this. The Arduino project, a microcontroller designed to make electronics more accessible to artists, and

the Processing project, a simplified version of Java intended to improve the experience of scripting visual effects for artists, are both dedicated to the same ideals. Underlying both projects, as well as the broader Maker movement, is an ideal of participation in one's own work.

Put simply, people who write code, particularly using the Agile methodology, are engaging in creative practice themselves, and they then display that creative practice through the artists who repurpose their work. This is a different design pattern than technology conventionally pursues, where the work is designed in isolation and released. I am using the Agile method to develop software because it does not require that one knows what the end shape of the software will be in order to pursue the end goal. Agile requires instead that developers pursue goals in sight, always keeping their development loose enough that they can repurpose their work without much effort, and it is ideal for working with artists.

What might be is a developmental model for software that permits transition in scope from the singular, minimum-viable-product model, to a model that builds on itself until complete. A small, perfect thing that does one thing very well, which permits artists to pair their own practice with a software built expressly to make their lives easier, for not too much money. This is important, because my initial approach assumes artists to be undercompensated for their work. Although artists are the central agents of production of all the cultural capital - the invisible value - of the culture industry, they are not the prime beneficiaries of the financial system that backs, stores, and distributes that capital. Therefore, software for artists needs to be inexpensive, and set up to be almost trivially easy to use. This reserves the value of scarcity to the ability of the artist, rather than applying the majority value to the role of the engineer. This kind of invisibility is the invisibility of good management, of any type of good administration. Like housekeeping, code recedes until something goes wrong.

To test this idea, I have approached people to produce video—games with the screenPerfect software in the context of a voluntary game jam – a type of collaborative space where participants work with digital tools to generate new, raw games in a limited window – and then compiling the results into an arcade machine for presentation.

1.3 Structure of the Remainder of this Document

1.3.1 Chapters

The remainder of this document is structured as follows. Chapter 2 covers various methods used in my research to examine the conceptual importance of accessible-technology artist tools, including the Agile Manifesto. Chapter 2 also sets out the restrictions and main theoretical texts that underly my premise of what constitutes an accessible tool. It also addresses a list of tools used to produce the thesis proper, including git revisions, LaTeX, and their shortcomings.

Chapter 3 addresses theoretical documents, and how they relate to the process of crafting solid software from an artistic/conceptual work perspective. This is where I have placed works by Cixous as well as an examination of Galloway's essays on gaming as algorithmic culture. I will also delimit which texts I consider useful for this work, and how to ground a video——game in the broader context of recent Canadian art history.

Chapter 4, Design Research, is the chapter where I work through the process of developing the game engine with reference to the benefits of open, closed, and ideologically-driven software. There is some strategic foresighting here through Doctorow's Pirate Cinema, the text I have used as a How Might We for what a collaborative theatre might be. I have also included reference to the Brechtian active audience, which is useful for gaming. Chapter 4 also contains the bulk of my research on collaborative practice within communities.

Chapter 5, The Trouble With Amateur, addresses some of the questions raised by Galloway in his *Gaming: Essays on a Algorithmic Culture*, which includes an examination of why games built in resistance to gaming tropes are largely unsuccessful and unpopular to play. This chapter is also where I will examine some of the problems of making free content easier to provide to the internet, which includes an economic examination of how the value curve of creative practice goes flat as accessibility increases.

Chapter 6 is my conclusion, a restatement of my arguments, and the source of some optimism.

Chapter 7 is a list of works cited.

1.3.2 Appendices

Appendix A is my annotated literature review.

Appendix B is a compilaton of github commit comments over the course of the writing of the game, which detail the direction of how someone reasonably confident with computers and programming learns a new language.

Appendix C is the code record of screenPerfect proper.

Appendix D is a list of the games made to date with screenPerfect and their installation sites, along with links to where they might be found for future installation.

2 Background

2.1 Existing Software

ScreenPerfect does not exist in a void. Although written fresh in Javascript, it is dependent on many frameworks and libraries in order to work. The code was developed using the Node framework for Javascript on the server, which is supported by Google. It mimics functionality produced by the Dataton Watchout system, which provides simultaneous video windows using a custom, private hardware platform. The software that screenPerfect interacts best with is Google Chrome.

2.1.1 Game Engines: What Are They?

A game engine is a collection of software designed to make it possible for a team of artists, developers, musicians, and producers to work together to produce a complete product. Traditionally, game engines are used to produce 2D or 3D experiences with clear "assets" such as 2D sprites or 3D player character/interaction models, backgrounds, interaction assets - crates, for example - music, and scripts in a programming language to tie all of these together into a play experience.

Some popular professional engines at the time of writing are Unity3D, which features native mobile integration and ease of scripting in both Javascript and C, Crytek, which comes with many high-end 3D resources preloaded for high definition graphic support, the Unreal Engine, which is quite stable and useful to experienced teams that prefer more control over their work.

There are popular hobby engines that de-emphasise programming as well, such as GameMaker, which is prized for PC compatibility, Game Salad for OSX, and Construct 2, which is PC-only but has a powerful engine to manage game physics and interactions.

These engines all assume a certain type of player interaction: they are designed to enable designers to produce specific types of games, such as a "shooter" or a "platformer", similar in style to the Call of Duty franchise or Nintendo's Mario series. The interactions available are easily understood as a language of action by their players, provided players have previous experience with video game play.

ScreenPerfect is distinct from pre-existing engines. It is a piece of software custom written to encourage artists to use their own skillset in image and video creation to explore what is possible in an interactive experience. Screenperfect has a set of play mechanics that have been pre-written. While they can be *extended* by anyone who knows the **daimio** language, the mechanics are straightforward to use and not designed to be altered by artists. This means that artists have a consistent environment in which to place their work, which will reliably showcase that work without them needing to learn how to program - an entirely new creative skillset - to do so.

2.1.2 Twine

Twine is the closest game engine to screenPerfect at the time of writing. Twine is a locally-installed hypertext-based branching narrative platform, which produces an interactive narrative that can be accessed through any web browser. It encourages expressive type styling and elements of multimedia, including music, coloured type, and well-designed game screens, but does not require them. Twine does not yet support video narratives, and is not entirely stored online as of yet.

The Twine engine was popularized by indie gaming celebrity Anna Anthropy in her 2012 book Rise of the Videogame Zinesters Anthropy, 2012, p. 2012. Since then, hundreds of Twines have gone live.

Twine is most notably popular with the queer indie gaming community. It has been used in wide popular release by Anna Anthropy, Merrit Kopas - author of *Lim*, Porpentine, Zöe Quinn, and games critic Soha El-Sabaawi, among others.

2.2 Multiscreen Video Technology

Dual screen technology, or more accurately, multi-screen synchronous web technology, is one of the big new ideas being heavily backed by Google in 2013. As a consequence, its Chrome browser has been designed to support software developed with a specific suite of frameworks, many of which are wholly supported by Google. This is an example of how software is not free: we cannot guarantee *what* is being done with the whole of

our software installations, or whether there is a security flaw in a system written to be dependent on development tools from major software houses.

That being said, Google supports it, so multi-screen technology using web browsers is something that is accessible. ScreenPerfect relies on Node.JS, which is based on Google's V8 engine, and MongoDB, which is owned by a separate organization. In addition, although the engine was originally authored independently using exclusively javascript, later versions have been reauthored using the **daimio** dataflow language, which has been released under the MIT licence by Bento Miso in Toronto.

The architecture of screenPerfect is wholly new, but the concept is based on the expensive Dataton Watchout system, which encourages producers to develop large multi-screen *single* video experiences on custom hardward. What screenPerfect does is permit people to use existing hardware to synch either the same video, or multiple videos, to one set of controls. This is also distinct from ChromeCast, which allows people to wirelessly pair a television with a touchscreen for control and consumption of the touchscreen at a larger size.

Neither of these software packages provide any kind of support for a branching video experience natively, nor do they provide the ability to use existing hardware with same.

2.3 Theory

2.3.1 Helene Cixous and the ecriture feminine

This is where a discussion of ecriture feminine as a rebellion against constructed language goes.

2.3.2 History of Women in Technology

This is where references to the Indiana paper and Sadie Plant goes.

2.3.3 Lev Manovich and Software Takes Command

Alongside feminist written history, this thesis falls into the frameworks described by Lev Manovich in his 2013 book Software Takes Command. This book emphasises what Manovich sees as a gap in the academy's examination of *media* as the central component of art and literature, and seeks instead to directly address questions of how *software* can

be itself analysed as possessing a direct impact on the systems of production with which it interacts.

I believe Manovich is overly aggressive in discounting the value and input of actual producers - I do not agree that individual forms of media are dead any more than I believe that print is dead - but I do think that his writing is directly related to my central research questions as to what impact a simple software tool might have on artistic prouction.

2.3.4 Pragmatic Reference Works

I forget what this is for. Is it for technology works?

3 State of the Art

3.1 Twine and Accessible Tools

- Twines are super accessible but still fettered by presentation
- CYOA is a basic interaction
- This is often described as "not a game"

3.1.1 Unity

3.2 Game Theory: Galloway, Anna Anthropy

- Games should be personal
- Chapter on artistic mods
- Mention Merrit Kopas' work on LIM and HUGPUNX and etc.

3.3 Cyborg Technology: Haraway

- Emma's paper at DiGRA

4

Industry Engagement and Design Research: Community Collaboration in Software Development

4.1 Industry Engagement

In order to run a game jam, one requires space, and people interested in working on games that are in line with any themes you might want to test. In order to access that space, I worked with the Bento Miso coworking facility here in Toronto.

Miso is a not-for-profit bricks and mortar site that serves as home for both Bento Box, a local development hub, and the Dames Making Games, a feminist initiative to introduce women to digital game development processes. As a board member of DMG, I have repeatedly witnessed the limitations of extant gamemaking tools. The software has bugs, runs on only a few systems, or relies heavily on metaphors and software constructs that are understood to those who already play a range of commercial digital games, but which are not clear to those of us who are new to gamemaking practice.

As a not-for-profit, Miso also serves as the hub for a great deal of Toronto's indie - independent - game development community. They offer professional support and development advice, and I felt there was a good match between their professional skillset and my research interests. The DMG traditionally run a jam in November, and felt that screenPerfect - a new software designed to be accessible in a short time frame to people with extant skills - would be a good match for the audience associated with the organization.

Miso, and Bento Box, offered to help me with coding a more accessible front end to the screenPerfect engine in time for the jam, so that I could get feedback on the system mechanics rather than just the interface.

4.1.1 Interfaces, engines, and interactions

A software interface is the part of the software that a person interacts with directly **interactiondestext** where a software engine is the part of the code that detects and defines what a computer can do with that interaction. The engine that I coded responds to interactions sourced from users. Interaction is largely what seems to define a user experience, but how the engine responds to an interaction can be just as important. A key part of the design of screenPerfect is that it was laid out to handle things like auto-saving invisibly, so that a user's work would not be lost.

The interface of the software is just as important as the engine, however, because a poorly designed interface will confuse a user, thereby rendering the experience of using the engine figuratively opaque. ScreenPerfect's roots are as a software engine, which takes user interaction and then does things with it. The user interacts with the interface, which speaks to the engine, which then returns values to whichever interface the user has selected.

FIGURE 4.1: screenPerfect software communication model

In the case of screenPerfect, the interface is laid out in three parts. The first part is the setup

4.1.2 Dames Making Games, Design Charrettes, and Game Jams

- Describe Dames Making Games here, history of jams

4.1.3 What is a game jam?

- Explain the Game Jam practice and premise
- Ground with references to design charettes.
- refer to appendices for jam participant experiences

4.2 Design Research

4.2.1 Artist Collaboration

Beginning development from a first position within the arts is unusual, even for an Agile workflow, but in the case of this software, it has been wholly driven by artistic collaboration. I firmly believe that simple tools to relieve the friction points

of the artistic process will lead to better art which is more widely available. This is to say, although the developer is themselves a creative who will decide *how* to solve problems, the problems to solve may be better handled by an outside party. This is down to an issue of demand.

In order for software to exist, and to be seen to exist, it requires an interaction. Unlike a hammer, which takes up space on a shelf, software is essentially a text document until it is used. As Gallowaysays, a game is defined by action 2006. Galloway, 2006

- Explain initial process with Hannah's YouTube idea

4.2.2 Summer installation of psXXYborg

- Describe installation of psXXYborg.
- Include photographs of original van in figures list.

4.2.3 Fiction Input

- Describe Cory Doctorow's Pirate Cinema scene. Out-take.
- Include photographs of original van

4.2.4 GitHub and learning on the internet

- Explain gitHub
- Explain open source software, benefits and questions therein.
- Explain closed-source software.

4.2.5 Problems in Artistic Collaboration

- Artist selling off installation
- No ownership for technician

5 Complications and Problems

5.1 Problems and Complications

In this section, I address issues with the format of ScreenPerfect games. I also address the problems of accessibility in tools, including the issue of avocatonal versus vocational creative practice.

5.1.1 Problems with CYOA as a format

* People want the full achievement * There is a huge pressure on creators for consistency in order to bring out catharsis and etc. * This is super tough in a CYOA because by definition, it's not a one-playthrough thing with accomplishments.

5.1.2 Avocational Work versus Professionalisation

* We currently live in a money-based system and need to earn it via our work. * The easier it is to make a game, does this reduce the percieved value of the game itself * Puts weight on the object that distributes the experience.

5.2 Good Branching is Expensive and Difficult

6 Conclusion

6.1 Conclusion

The work that has gone into the production and release of screenPerfect is not inconsiderable. As a creative project based in grounded theory and reflective practice, code is a tricky thing to pin down. It must be declarative, yet it reveals the internal architecture of the people who write it. To write code is to be a craftsperson, and to be a craftsperson is to reveal some of yourself with every line. Programming solo is as rewarding as any other solitary occupation, yet it leaves many loose ends. An excellent piece of software is likely to require input from a wide array of specialists in graphic design, interface development, and logic. There is an inevitability to induced flaws - bugs - that cause the program to fail. Once complete, it is likely that finished software will fall out of fashion: software is the unfinished symphony of the 21st century. Just as there is no way to call a piece of writing finished, because another word can always be added or cut loose, code is subject to scope creep. Code changes. It is not a silkscreen, once pulled and forever finished. It is not a painting, which, dried and delivered, is safe until the conservators come for it. Code lives, like writing, in context and within an ecosystem. Unlike writing, code answers to its context; without the machines to whom it speaks, it is without consequence. Within those machines, it may have a concrete effect on the world around it, and for that reason it continues to be valued; this is the craftsmanship that, unlike art, continues to make a living. Code cannot be set aside; although it will work as intended, it will break without permission. To code is to attempt to write a coherent world into being, to attempt to factor in all the diverse chances within. In Cixous' Laugh of the MedusaCixous, 1976, she expresses that to be considered real, women must write for themselves. In my thesis, I have extended this to the world of code: one must write after one's own interests, because to take only that work which is assigned is to fall behind. Writing a projection/presentation/game system has been

work designed to address the problem of what, exactly, a game is, or what is a valuable piece of work. screenPerfect is designed to evaporate, leaving only the experience of its content to represent itself. It does not care what your content is, or to whom you're serving it, or where. The emphasis is on allowing your audience to experience things as quickly and easily as possible. Works produced using screenPerfect can be displayed anywhere in the privileged world; anywhere a series of screens and a single server can be set up. This emphasis on experience moves the interaction sphere into the world. The display of video-art and collaborative gameplay made possible through screenPerfect can be anywhere at all, and indeed works best at night, outdoors, in temporary installations. These are the new/old/new exhibits, the unmissable one-time-only parties, the experience that happens in a hard to access place but leaves no marks for future visitors to interpret.

The reflective portion of this research has been to address the question of use and pragmatism, as well as what constitutes research within an artistic context. The answer is difficult to quantify; research is pursuing a read, book-learning, critical examination of a practice, as all art is practice. Art is fundamentally blue collar, as is coding; both are works of craftsmanship, both can be helped along by automation, but ultimately, their declaration within a finished state is dependent upon the person who has produced them. The blue-collar trade of art has been sold out by the academy, linked tightly to the idea that thinking about a thing is more valuable than working on a thing oneself. Coders are legendarily difficult to organize, resistant to unions or to the thought that they are themselves labourers.

The newest video games try to dream worlds past being human, and most fall far short. Rather than permitting a wide exploration of possibilities, many possibilities narrow to the point of a gun, to the same forearms for twenty years (twitter). At the heart of screenPerfect is the idea that we can pull away from artificial distance and have instead on-site participation, unique experiences that project real, contemporary art into real, contemporary spaces. We can have events anywhere, and these events can bring anyone together, with even terrible technology. Underlying the architecture of this code is the idea that all the different flavours of classism should be undone with the opportunity to see amazing things, no matter who you are. Space should belong to the people who occupy it most often, not only the people who pay for it at a distance.

Learning new programming languages is always a challenge, as is enacting a pragmatic device from the perspective of art theory. There is the concern that these works are not for anything, not for a job or an application or a visible piece of content, a series of products released to do something in the world. Many applications, games and devices are released into the world that simply consume resources for the sake of their own

consumption, simply to show that the people involved have the resources to spend. A new digital toy is frequently out of reach for the vast group of people who depend on their existing technology to work for as long as they can make it do so, and therefore, these tools are designed in the same way as the first university mainframes: they live somewhere else, and can be accessed by even the most unfortunate smartphones, five years out of date and slow. The idea is to permit this sort of advanced media to get to places that I do not expect it to turn up, in places that are specifically not shiny. The idea is to use existing technology to permit exploration: use what there is, and then make it greater, rather than interject an external device that will require people to spend resources to use.

Ultimately, this sort of software development is about permission. Permission for people to do what they like in the spaces they need to occupy and use, including the digital space. This is not about developing a single app or a platform that will be easily marketed, but is instead about focusing on the value of the exclusive experience. The Game Jam, which is a located, people-in-the-room bit of enthusiasm, and the value of a small community which helps one another to develop, is reflected in the variety of games produced for the system. This is about exploring the possibilities of a space designed to share an experience through a personal connection.

This part is about the different values permitted to different classes of entertainment. At a high level, art can afford to be alienating, and indeed is frequently valued more highly for its power of alienation than for any other thing. This is a distinction made especially true in video games, where the power of "fun" tends to be valued highly for its commercial properties. If a game is not "fun," all elements of its interactive powers of storytelling cease. This means that games which are not "fun" are widely called by other names - interactive new media art, for example.

screenPerfect is designed to permit the development of games that are not only not necessarily fun, but which may be narratively incomplete, or nonsensical, while still being absorbing. These are games that do not require reading, but permit themselves to be experienced as deeply as a given group of readers wish to experience them. The tool can be perverted: perverse use is built in, with video copyright being at such a premium. It can be used to host experiences in galleries or in warehouses, by people with minimal technical knowledge and little ability to mask their normal online activities. This is a device to let people make maximum use of the devices they already have, to host a dance party on a subway or a massive art tour through a gallery. The demonstration content may be upsetting, but the access permitted is broad. This is about sharing things, for the better.

7 Endnotes, References, Works Cited

7.1 Works Cited

A

Agile Manifesto

A.1 Agile Manifesto

Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more. © 2001, agilemanifesto.com

A.1.1 What Is Agile?

* Put in what is pair programming as well.

Agile is an ideal based on a manifesto released at the turn of the century. It is not specific enough to serve as a development framework, but instead serves to structure a way of thinking about software development in collaboration between developers and their user population. The original Agile Manifesto was released in 2001, and has a vast number of signatories to date.

Agile is intended to address the gap between planned software, which is typically fixed at release, and reality, which is that software breaks routinely and needs regular maintenance and iterative upgrades. The manifesto suggests that developers should focus on results over process, software that works over software that's well-documented, collaboration over contract specifics, and responsiveness to change over following a specific plan.

Agile has proven useful for this project because its emphasis on responsiveness and deliverables has ensured that the working software of screenPerfect can move forward to permit more games to be made, rather than locking the software to a concern for itself as an object. This is a valuable way to move through a development process which spares us from needing to commit to a given method, and instead emphasizes results from practice.

This practice must be documented, which is where the code-commit process appears. Agile deprivileges documentation for itself, preferring to include the necessary as the code itself is logged. Therefore, I have chosen to use the Git toolset to retain my version control on both this document itself and the code developed to run the main game engine.

B

Annotated Literature Review

B.1 Literature Review

http://kotaku.com/the-weird-escapism-of-life-sims-730629952 Leigh Alexander on aspiring to pay a mortgage in real life, which is important because people won't be able to do that, a lot, in North America.

http://agilemanifesto.org/principles.html The Agile Manifesto, which backbones my actual software development style: working software is what counts. Important, as Agile is in direct defiance of corporate control structures.

Maly, T. (2013). We Have Always Coded. Medium.com. https://medium.com/weird-future/2acc5ba75929 This article investigates gender essentialism from the perspective of biological essentialist arguments frequently used to say women can't or shouldn't code because of various, entirely specious, evolutionary problems. This is an argument that happens on top of a perceived resource scarcity; the scarcity is in this case employment of women in technology, so opportunity.

Fashion article about hegemony of fashion writing and the death of exclusivity, with notes on shows being locked down to resist the blog invasion. http://thenewinquiry.com/essays/cool-fronthot-mess/[I need some work on how basic economics works with supply and demand in the absence of a good money supply.

bell hooks. (1992). Is Paris Burning?. Black looks: race and representation (pp. 145-156). Boston, MA: South End Press. bell hooks is always useful in concert with Derrida to explain that you can't actually know what anyone else is actually thinking; having sympathy for other people is not the same as understanding their direct experience. Useful because it underlies a lot of feminist practice. This is an article about exclusivity,

which is an issue of privilege, which is an issue of perceived resource scarcity, in this case access.

Bizzocchi, J. Tanenbaum, J. (2011) Well Read: Applying Close Reading Techniques to Gameplay Experiences. In Well-Played 3.0, Drew Davidson Eds., Etc press www.etc.cmu.edu—well-read-jim-bizzocchi-joshua-tanenbaum Something to explain game design processes in the technical section of the document.

Buxton, W. (2007) Sketching User Experiences: Getting the Design Right and the Right Design. Morgan Kaufmann Publishers. Something to explain user interface design practices in the technical section.

Chun, W. H. (2011). Invisibly Visible; Visibly Invisible and On Sourcery and Source Code. Programmed visions software and memory (pp. 1-54). Cambridge, Mass.: MIT Press. Still need to read this, but the title is pretty relevant to the central research question of my thesis, which is on the value of code and invisibility as a permission.

Cixous, H., Cohen, K., Cohen, P. (1976). The Laugh Of The Medusa. Signs: Journal of Women in Culture and Society, 1(4), 875. Woman must write her own desires into being in order to be seen. This is a paper that reinforces arguments about scarcity of perception, and issues of control, specifically of women and women's opinions.

Deleuze, G., Guattari, F. (1987). Introduction:Rhizome. A thousand plateaus: capitalism and schizophrenia (pp. 3-4). Minneapolis: University of Minnesota Press. Everyone else is doing it [including them]. May as well. I gather they're popular right now.

Deleuze, G. (1992). Postscript on the Societies of Control. October, Winter(59), 3-7. Surveillance culture is bad for people, yet inevitable as it becomes automated. This is an issue of control, which is subverted by taking possession of even a single means of production; to refuse to code is to be illiterate of the systems by which production is governed. To refuse to acknowledge the implicit control of a system is to lie to oneself; if other people have designed the system, the system operates[Alex Leitch, 2013-10-01 2:18 PM There are no complete systems, particularly in computers, because Godel's incompleteness theorem says so. This is a joke that is also true and I am not sure how to cite it, but it holds for most systems of even informal logic, which computers are composed of. This is a purely theoretical way to look at a computer system that is both true and not true; the incompleteness theorem concerns math systems, not people, specifically. You can't test people for their incompleteness. But people are still incomplete. If they weren't incomplete, they wouldn't have religion.] as it is intended. The way out of this is to read the system, then find the gap within it.

Dell, K. (1998). Contract with the skin: masochism, performance art, and the 1970's. Minneapolis:University of Minnesota Press. Pain or revulsion is another way to escape a system, which is to make it so dear - dear as in price - that it is difficult to reproduce the work because it costs too much. Abjection, or the ability to pay for something with revulsion, is one of the less efficient but more effective systems of resistance. The liminal space represented by a willingness to publicly maim oneself is reserved for those who do not fit well within a system that relies on completion: broken skin stands in for a refusal to submit to hierarchy. This collapses in various ways over time - it's unlikely that even facial tattoos will keep people out of the workplace for long - but still represents a real way that people refuse to be included in a more perfect/uniform work. This is an article on how masochism, the revealing of the inside, has a recent history in art and what role that sort of display performance contributes to feminism.

Doctorow, C. (2008). Little Brother. New York: Tom Doherty Associates. A detailed look at how surveillance culture breaks down social contracts, and a how-to guide on resistance via action disguised as a novel. Also has a variety of excellent, accurate examples of ways to disguise data so that it cannot be confirmed by a rogue authority. Connected to Deleuze on Societies of Control, and specifically addresses homeland security spy tactics.

Doctorow, C. (2012). Pirate Cinema. New York: Tom Doherty Associates, LLC. Contains a scene with multiple tiny projectors used to set up a cinema in a park from pockets, which is more or less what the software itself is supposed to do when it runs. The entire book is about resistance to copyright authority. Contains a quite didactic passage on how even hardware control chips do not actually control or prevent smart-enough people from using controlled software, and in doing so, presents a vision of the world where the most privileged are no longer privileged with money alone, but also with knowledge or access, which is a type of prestige - which is a type of magic trick. Concerningly libertarian.

Gleick, J. (2011). The information: a history, a theory, a flood. New York: Pantheon Books. A survey text of the history and development of data-centric information technology. Explains a little of the context for how tools like screenperfect can be expected, themselves, to proliferate to the point of uselessness. This is useful because it prevents needing to look up each paper about completeness theories and map-rename signal-to-noise mathematics independently. Signal-to-noise mathematics are important because they provide a way to think about how to privilege information in the learning process, or the internet search process; people passively look up how to find what they

need. Downside: The noise often contains trace characters that allow further exploration. Upside: there really isn't that much signal out there, no matter how much noise is happening.

Gram, S. (2013, March 1). Textual Relations: The Young-Girl and the Selfie. Textual Relations. Retrieved April 12, 2013, from http://text-relations.blogspot.ca/2013/03/the-young-girl-and-selfie.html Young women's bodies are not only super-powerful, but can be the stereotype vehicles for all of consumer culture. This is important because young women are disproportionately discouraged from tech culture, even as they control the scarcity that is approved sexual relations within North America, a scarcity that cannot be overcome with money alone - you can't buy love, they say. So this is valuable because it is an excellent analysis of how stereotypically correct bodies benefit from fitting into a system of action, which is related to code practice because only stereotypically correct bodies are encouraged to participate. Per masochism, however, there are no correct bodies, only correct images of bodies. Resist the system in a predictable direction, and you become a new format of marketed body, with a new predictability. This predictability can be coded, but the closer one gets to perfect, the harder it becomes to occupy the role while remaining human. This is a deeply misogynist text, but is a perfect text for examining the role of image on the internet, and things which are seen but hard to describe, as code is.

Grosz, E. A. (2008). Chaos, Cosmos, Territory, Architecture. Chaos, territory, art: Deleuze and the framing of the earth (pp. 15-28). New York: Columbia University Press. Technology as sexual performance and definition of space. Not terribly well-realized but more academic than other sources on the same subject. Useful because code is a creative process, and creative processes - per Wilde - are useless ... like peacock feathers, or any other sort of look-at-me performance. Even things which do things are useless.

Haraway, D. (1987). A Manifesto For Cyborgs: Science, Technology, And Socialist Feminism In The 1980s. Australian Feminist Studies, 2(4), 1-42. Primary text on women restructuring their bodies, invisibly, to take over the world. See also Quinn Norton on IUDs (http://www.quinnnorton.com/said/?p=404) - this is useful because women can resist commodification, such as that described by TIQQUN, invisibly. Code is, in Agile practice, shifting from architecture to a sort of cooking; this library and that, all put together in a frame to pursue an idea, rather than to do a specific thing from the outset. This is a text about resisting control systems by allowing oneself to cooperate until there is a space to break free. Frequently, people don't even notice you have.

Haraway, D. (2009). The companion species manifesto: dogs, people and significant otherness. Chicago, Ill.: Prickly Paradigm Press. More Haraway. Now on cancer, not

sex. I need to read this but I don't think it will be too useful, except that it articulates that humans, with their tool-use, are not actually special; we are part of a system of mammals. This may be useful elsewhere.

Hunicke, R., LeBlanc, M., Zubek, R. (2004) MDA: A Formal Approach to Game Design and Game Research. sakai.rutgers.edu—hunicke$_2$004.$pdfMoreongamedesigntechniquesforthetechnica$

Kristeva, J. (1982). Powers of horror: an essay on abjection. New York: Columbia University Press. (http://www.csus.edu/indiv/o/obriene/art206/readings/kristevaHorrifying things have a power that is more potent than any non-horrifying things could hope to possess. This paper details why that is. It goes very nicely with Cixous and discussions of the IUD, because it is about what happens when barriers truly break down. I think Kristeva's horrors are basically the key to the entire news cycle and Grand Theft Auto to boot. This paper is the original on how revolting things are fascinating but resist being part of a system, unless they're cleaned away and perfect. See above comments on masochism paper; the awesome attraction of the awful.

Krug, Steve (2000) Don't Make Me Think: A Common Sense Approach to Web Usability. Riders Publishers. This is another technical paper for arguing that software design should be totally invisible. Useful because it ties together the systems of control argument - control is implicit, presented as undefeatable, a smooth surface - with the idea that things should be useable, so that people can find their own uses for the tool beyond what is initially intended by the author.

Schafer, T. (1998). Grim Fandango (1.0) [Video Game]. USA:LucasArts. Classic adventure game with minimal interface and a fixed runthrough. One of the last great adventure games. An excellent exercise in game design where the game itself is preset, but the ideas the game displays, including an interest in a subculture that is not much popularly examined (Mexico), and a good narrative. Evidence that narrative is important in gameplay, which is key to the development of screenperfect as a narrative branching tool. Also remarkably and incredibly broken on contemporary systems, as the initial code was rendered directly by processor speed, rather than at a stage or two removed; the game was broken by Moore's Law, which is good evidence for why tools need to be considered unto themselves. The new narratives are temporary. This is a narrative about the temporary; death, and the waiting period before leaving - while being wound up in a longstanding celebration.

Luvaas, B. (2006). Re-producing pop: The aesthetics of ambivalence in a contemporary dance music.International Journal of Cultural Studies, 9(6), 167-187. Retrieved April 10, 2013, from the Scholar's Portal database. An interesting look at what ethnographic research can be, and the speed of cultural shift and recycle since the rise of the internet.

To be read in concert with various VICE mag articles about cocaine, new york. Used originally in article about Seapunk movement.

Moggridge, Bill (2006). Designing Interactions. MIT Press, Cambridge MA. More technical reading about how people interact with software, about how people can control interactions.

Moyer, J. (2012, September 14). Our Band Could Be Your Band: How the Brooklynization of culture killed regional music scenes - Washington City Paper. Washington City Paper - D.C. Arts, News, Food and Living. Retrieved April 22, 2013, from http://www.washingtoncitypaper.com/articles/43235/our-band-could-be-your-band-how-the-brooklynization-of/

Cultural uniformity because the internet makes things from different places seem the same, even though they're really not the same. Relates to the Young-Girl article about how if you are one perfect shape, that perfect shape will always sell at least a little, which obfuscates the truly beautiful and interestingly specific evolutions with things which have been data-optimized to be more popular. Popular isn't better, and neither is monoculture, but also no good is the sort of individuality that is itself a sort of monoculture.

Mulvey, L. (1975). Visual Pleasure and Narrative Cinema. Screen, 16(3), 6-18. On the male gaze, which is the central gaze in most videogames, particularly first-person shooters. This is important because the male gaze sets how most blockbuster video games are allowed to be perceived. Important because video games, like most software, are mainly compared to cinema, even though they have very little in common with cinema for elements beyond the technical. Core to arguments about how women are seen, which is essential to understand the TIQQUN readings in their slightly tongue-in-cheek misogyny.

One Laptop per Child. (n.d.). One Laptop per Child. Retrieved July 3, 2013, from http://one.laptop.org/ I was thinking about discussing how the OLPC project led to various other tech advances, including the rasPI - it made netbooks happen, then tablets happened. The OLPC was the project that said "wait, things don't need to be faster, they need to be better." Absolute disaster; in the countries it was intended for, it was already superceded by mobile phones. Classic example of condescending outsiders trying to Make A Difference rather than examining difference. Probably too broad a scope for this project.

Orlan: a hybrid body of artworks. (2010). London [u.a.: Routledge. Orlan led to Lady Gaga so directly that she has since sued her. Discussing the liminality and limits of flesh without Orlan's surgeries is a challenge; almost no other artist (burden? Shoot)

has gone so far, but this distance is collapsed in film like Nip—tuck and the normalization of Hollywood surgery. Related to Kristeva and articles about the mortification of the flesh for the sake of appearances, which is what I am interested in with the arcade box. Although I want that to be subtly upsetting, not overtly upsetting.

Reines, A., TIQQUN. (2012). Preliminary materials for a theory of the young-girl. Los Angeles, CA: Semiotext(e) The new translation, which includes a feminist preface by Reines about the body of young women and how she almost was sick over the assertions of TIQQUN, which happened about the same time as everyone else was going bananas for Second Life, a game where you make an entirely new body that has since been abandoned by all but the most escapist. TIQQUN accurately observe that people are escaping into their own bodies, not those of the computer screen; the new presentation is that the brain and image on the internet reinforce the physical appearance through the phone, a piece of technology governed by the male gaze.

Stephenson, N. (1995). The diamond age, or, Young lady's illustrated primer. New York: Bantam Books. This is pretty well a perfect piece of fiction about cyborgs and universal education and China as an Oriental-escape paradise. Fun look at a post-scarcity economy that has simultaneously happened and can't happen. This is a book about an alternative resistance to the always-on personal presentation future, where books reflect their users. This is a fantasia, but an appealing one, with a lot to say about the subject of veterans, what abuse looks like from a perspective other than the dominant, and what recovery might look like. The main characters are all female, and all develop in different directions, including one who escapes by using the book to hack out a new life under direct supervision. Contains an unpleasant thesis about the value of personal matriarchal influence on future leadership.

Sternberg, M. (2012). They Bleed Pixels (1.0) [Video Game]. Toronto:SpookySquid Games. Excellent representation of a female lead game character in a genuinely challenging platformer. Useful because it exposes the programmer's preference for difficult-but-rewarding game mechanic loops, along with a conscious choice to show a young woman who has strong personal agency as a hero. A manifesto for better, simpler video games.

Swartz, A. (2013). Aaron Swartz's A programmable Web an unfinished work. San Rafael, Calif.: Morgan Claypool Publishers. The internet doesn't belong to us, but it could, and here are some technical guidelines to pursuing that as a worthy goal. This is the other way to approach technical development; something that should be extended rather than presented as complete in and of itself. Swartz rebels against societies of control by describing systems to expose information at a basic level rather than obfuscate them. This eventually led to his death.

Team Little Angels (2009). Bayonetta (1.0) [Video Game]. Japan:Sega. What a hilariously sexist but also perfect meta-narrative of female power while subject to the male gaze. The rudest, most violent fun game released to ever feature a lady protected, literally, by her hair. A game with a strong female lead in the hilariously Kate Beaton "strong female characters" mold, which is problematic in its presentation even as it is simultaneously winking. Has unfortunately fixed gameplay goals, but allows players the reward of working through the game on a basic mechanic of style rather than skill alone. Fun!

Toom, A. (2012). Considering the Artistry and Epistemology of Tacit Knowledge and Knowing. Educational Theory, 62, 621-640. Retrieved April 12, 2013, from the Scholar's Portal database. More technical information on how to design interfaces so that people understand, passively, what they're supposed to do with it. This is about passive learning, which is how most people learn software: through exposure and experience with previous systems, we understand the language that the developers no longer expose even though help systems. The way of using the software becomes implicit.

Volition Inc. (2011). Saint's Row the Third (1.0) [Video Game]. USA:THQ. This and the followup, Saint's Row 4. Games that took the GTA pattern and subverted it to make a game that is cleverly and strongly and messily about playing video games and the fantasies of those games. Saint's Row is a sandbox video game about playing videogames and what a videogame means at its base. It allows people to play as whatever type of character they like, which exposes the fallacy that videogames are solidly about anything but mechanics; the art and design on top expose the code in their very mutability, but there are no ways to solve the game puzzles except violence. This is entertaining, because rather than being a game about traffic patterns and random mayhem specifically (GTA-V), it is a game about playing games, about false achievements and the ability to do anything at all as long as it's violent. Also notable because first lead female character is a hacker from the FBI. This is an important plot point. You can also play gay or with the robot AI you rescue ... but not the vice president, who's a dude. Central to my argument that software development is a second-stage creative practice because with no fixed skins, the game itself is much more exposed.

C

Game Jam Documentation

C.0.1 Questions To Ask Game Jammers

- What were you expecting when you came to the jam?
- What features did you immediately want in your software?
- How has your group process worked throughout the week?
- How is your group process going today?

C.0.2 Games List

- Porn Game by Maxwell Lander
- Grimoire by Katie Foster and Mikayla Carson
- Kill Fuck Marry by
- Mind Safe by Dann Toliver and Robby
- Glitch95 by Arielle, Rebecca and Bronwyn
- Omm by Brittany and Diana
- Cyborg Goddess by Cara and Kate McKnyte
- Empty Puppet by Danielle Hopkins and Dawn

C.0.3 Bug Discovery

- Room Zero must be the first room edited.
- Room Order cannot be altered in a meaningful way - ID is hidden from users

- WebM video is unplayable on Apple devices
- H.264 video is slow to unplayable on non-Apple devices

C.0.4 Features Requested by Game Jammers

- Sound effects on control input - requested by Arielle
- Timed hotspots which appear and disappear on specific video cues
- A game tracer that tracks which choices players make, and records their games
- Gesture controls - pinch, zoom, throw - on touchpoints.
- Tree View to visualize how a game is laid out
- Rooms cannot be deleted - delete and reorder rooms
- Games cannot be deleted - delete and reorder games
- Copy and paste room layouts so that one does not have to recreate grids - done.
- Hotspots that can move around the room.

C.0.5 Notes from committed jammers about screenPerfect

For Arielle, the most engaged of the jammers, the idea of turning any touch device into a custom console controller, with custom buttons, is engaging. The more traditional the game developer, the harder a time they had with the idea that they'd be showcasing content with the narrowest help from the new tool. The filmmakers were very impressed with the ability to not touch a darn thing and have considerable success.

Bibliography

Anthropy, A. (2012). *Rise of the videogame zinesters: how freaks, normals, amateurs, artists, dreamers, dropouts, queers, housewives, and people like you are taking back an art form* (Seven Stories Press 1st ed.). New York, USA: Seven Stories Press.

Bissell, T. (2013, September 25). Poison tree: an open letter to nico bellic about 'grand theft auto v'. *Grantland*. Retrieved from http://www.grantland.com/story/_/id/9719678/tom-bissell-writes-letter-niko-bellic-grand-theft-auto-v

Cixous, H. (1976). Laugh of the medusa. *Signs: Journal of Women in Culture and Society*, *1*(4), 875.

Galloway, A. (2006). *Gaming: essays on algorithmic culture (electronic mediations)*. Minneapolis: University of Minnesota Press.

Hunicke, R., LeBlanc, M. & Zubek, R. (2004). Mda: a formal approach to game design and game research. *Rutgers*. http://www.cs.northwestern.edu/ hunicke/MDA.pdf.