An Introduction to Retro Software Homebrew:

Sustaining the Growth

of an NES Homebrew Community Through Online Communication

A Major Qualifying Project Report
Submitted to the Faculty of
the Worcester Polytechnic Institute
in partial fulfillment of the requirements for the
Degree of Bachelor of Science, Professional Writing

Submitted by William Poirier

Advised by Jennifer DeWinter

September 16, 2014

wjpoirier@wpi.edu

Abstract

In the modern computer age of information, the line between consumers and producers has become blurry in many cases. For some programming hobbyists and videogame fans, there is no line. There exist several online communities that develop and share “homebrew” software with the world. Homebrew in this context refers to software developed for proprietary system by hobbyists and/or hackers. There is no endorsement usually from official intellectual property holders. The relationships between homebrew communities and companies are strange and awkward, containing some of the most passionate fans for a brand like Nintendo or Atari while also arguably infringing on copyright. From a professional communication standpoint, there is much to be interested in studying. Thanks to homebrew communities, a massive amount of documentation on reverse engineered proprietary hardware has entered the public domain. Yet it is an understudied field in academia and is a heavily misunderstood phenomenon by intellectual property holders. This report hopes to shine light on the community and break down several of the misunderstandings. It also strives to evaluate the quality of the user experience and documentation within homebrew communities and recommend improvements in their communications.

Primarily, this MQP deals with the question of how a homebrew community can sustain membership through their technical communications. A community catered to the development of Nintendo Entertainment System homebrew called NESDEV was selected. Specifically, this study dealt with how well a newcomer could find and comprehend the documentation provided by the NESDEV community. Test participants were tasked with learning basic mathematical and programming constructs necessary for low-level assembly programming. This included binary numbers, bitwise logic, hexadecimal conversions to other bases, and programming in 6502 assembly.

The research showed that both technical minded and non-technical minded people had difficulty at various degrees with various parts of the website. The design of the NESDEV wiki needs improvement in order to attract more members. Also, several documents were found to be too technical for beginners. It was recommended the community looks at the data contained in this report and decide how or if the website should be improved. Although not a full representative sample of all homebrew communities, the study demonstrated many design and communication pitfalls a volunteer and hobbyist driven community could fall victim to. Based on preliminary analysis, other homebrew communities exhibit similar technical communication problems. In the future, I intend to adapt this study to more homebrew communities in order to bring to their attention ways their communications could be improved and made more accessible to newcomers.

Table of Contents

Table of Figures

Chapter 1 – Introduction to Homebrew and Modding

At the edges of the game industry, there exist realms where fandom dedication extends beyond playing games and steps into the world of game development. One such realm is the *Homebrew* movement. Homebrew as related to videogames refers to games and software created for proprietary game systems without permission from the systems' intellectual property holders. Everything in homebrew is completely unofficial and not usually endorsed by the original game developers or publishers of the hardware. Homebrewing is closely related to fan-games and modding. Fan games, as the name implies, are games created by fans of existing franchises and characters. Fan games are almost never officially endorsed by the original developers. Related to the fan-game is game modification, commonly referred to as modding. This practice involves taking a previously existing game and changing elements of it (such as levels, characters, music, etc.) into something different or new. Modding thus is closely related to hacking in the traditional form of the word, as in hacking hardware together sloppily to create something new. In some cases, the two words are synonymous.

The subject of game modification and user creation is sparsely studied in academia. Although some exists, such as Hector Postigo's research, scope is often limited to general discussions of community and intellectual property debates. Many of the subtle aspects and shining potential of game modding and fanworks have yet to be discussed. Homebrewing is even less discussed. Here, the world of fan games and modification overlap, but an important distinction exists. Homebrewing is similar to fangames in that it uses prior existing intellectual property, but do not necessarily star established characters. Instead, homebrewing is primarily a hardware based phenomena. Although some homebrew projects have been conventional fan-games, wholly original works have been created for a variety of

proprietary systems, ranging from classic systems such as the Atari 2600 and NES, to modern systems like the Xbox 360 and Nintendo Wii. Homebrewing also usually involves building code from the ground up opposed to modifying prior existing games. It involves a different form of modding: the reverse engineering of hardware locked within the vaults of corporations and the US Patent Office. Homebrewing, for better or for worse, sets hardware free.

Homebrewing is often associated with legally questionable activities such as software piracy and jailbreaking. It's a natural assumption, given its nature, but it ignores the more interesting and subtle aspects of having extremely dedicated fan-bases. Homebrew is an active community that presents many interesting cases to study in both the triumphs and failings of internet communities. It presents a case of users taking back control from perhaps unnecessary corporate secrets, and giving new life to obsolete hardware. John Vanderhoef describes this interest in obsolete hardware effectively, "[H]omebrew games illustrate one set of alternative practices and stories that refuse to abandon yesterday's gaming technology based on market forces; that instead want to insist on the utility and value of older consoles in a culture where computer technology is considered obsolete upon purchase" (The Best Games of Yesterday Tomorrow, Vanderhoef, 5). With such passion for a hobby combined with the unique communications of the internet, few communities possess better potential than the power to demonstrate transformative and participatory media.

A strange mix of different social theories and economic models can be applied to the homebrewing communities. On one hand, a communitarian, shared information mentality is present. As Will Jordon discusses in his ROM Hacking study *From Rule-Breaking to ROM-Hacking: Theorizing the Computer Game-as-Commodity*, the mantra "information should be set free" is fully in application for homebrew communities. Nick Dyer-Witheford in *Cybermarx* discusses the general trend

of information being freed from its state as a corporate commodity. “Because advanced communications networks can circulate information good very fast and very widely, good that are by their very nature dependent on extensive availability of appropriate machines, skills, and knowledge, imposing commodity exchange has proven extraordinarily difficult. A wave of everyday media 'piracy,' including photocopying, home taping, bootlegged videos, unpaid reception of satellite signals, copying of computer software, and hacking is informally decommodifying information flows” (Dyer-Witheford, 202). On the other hand, many homebrew products have been turned into commercial products, and pirating original commercially sold is looked down upon fellow homebrewers due to their status as small independent game developers. Strange intellectual property questions emerge. Under US patent law, protection for electronics expires after 25 years. But copyright lasts far longer; and trademarks are indefinite so long as they are enforced. For instance, there have been several cases where the current owners of Atari have stepped in to case domains using the Atari name[1], but will generally let homebrew software be sold so long as the Atari logo is not officially printed on them. Also, somewhat related to homebrew games, a variety of hardware clones for obsolete hardware like the NES are available for gamers to purchase in specialty game stores and online retailers. But the strangest and most blurry case within the open economy is reproduction cartridges. Some online entrepreneurs and hobbyists have made businesses out of creating reproduction cartridges of rare and unreleased games.[2] An unusual hybrid economy has emerged that juggles freedom of information and open hardware specifications with a 1990s internet entrepreneurial spirit.

Why Study Homebrew?

1 http://games-beta.slashdot.org/story/11/08/22/1959255/atari-targets-retro-community-with-cease-desist
2 http://www.nesreproductions.com/

A review of the academic literature reveals sparse discussion on homebrew. The most thorough study I could find directly on video game homebrew was Brett Camper of MIT's study on GameBoy Advance homebrew. His research primarily focused on detailing how the limitations of the system sparked engineering creativity, a theme I believe is even more true as the capabilities of a system shrink; much of the beauty in designing for small computer systems is what can be done within tiny system limits. Mirko Tobias Schäfer also made a passing reference to Xbox homebrew in *Bastard Culture,* a treatise on the participatory culture technology has brought forth. While useful for understanding the motives and nature of homebrew culturally within the context of the modern internet, this paper was not focused exclusively on the homebrew movement. Slightly more research has been done regarding modding.

While homebrew is not heavily studied, community related research exists in droves.. Many questions arise from the development of communities. Is it sustainable? How do we factor in the needs of the individual versus the whole community? Community management itself becomes a significant factor to consider, and several books have been written specifically on the subject. Relating to my community approach to studying homebrew, work on community sustainability is thorough and abundant; quite the opposite of the scarcity of homebrew related research. For this research project, I looked into books such as *Community Open Spaces* by Francis, Cashdan, and Paxon, *Sustainable Communities* by Calthorpe and Ryn, and *Creating Succesful Communities* by Mantell, Harper, and Propst. Even though the books I've chosen for this project deal specifically with real world community management and environmental problems, much of the same mindset of sustainability in the real world

can be applied to online communities. We even may find research into community sustainability for homebrew also translates back into the so called “real world.”

Understanding Homebrew from a Communication Standpoint

Homebrewing is a poorly understood and often misrepresented phenomena in the mainstream. Ask a gamer the meaning of “homebrew” and his or her likely response will be related to alcohol. Reactions to the products of homebrewers from outside the community are often ambivalent. Two years ago, I wrote for a journalism class for James Dempsey a short article about NES homebrewing. I had the following to say: “Prior to the release of *Battle Kid, the Fortress of Peril,* several commenters on sites such as Destructoid, GoNintendo, Kotaku, and others accused the game of being nothing more than a cheap knock-off of *Mega Man.* One poster in the comments section for *The Escapist* said, 'Another nostalgia wank fest from an idiot who thinks spending a long, hard time making a game exclusively for a console very few gamers own which ceased development twenty years ago will earn him any money or respect' (Poirier, 2012).” While many users were willing to step in and defend Sivak, these sorts of reactions suggest the homebrewing community has an image problem either unacknowledged or ignored by homebrewers.

From a technical communication viewpoint, the homebrew community provides a rich and untapped source to study. A unique perspective regarding the relationship with corporations and consumers exists. With homebrew, proprietary information regarding various game systems only intended for internal company usage has leaked into the public domain. For example, Steve Wright's 1979 *Stella Programming Guide*[3]*,* an internal document used at Atari for programming the Atari 2600

3 https://archive.org/details/StellaProgrammersGuide

VCS is easily accessible through the internet archive's public domain sources. For companies without such leaks, an enormous community effort in reverse engineering and documenting several proprietary systems has taken place. For the Nintendo Entertainment System, little to no official documentation has been leaked or released, yet dozens and dozens of documents have been written and a massive (by homebrew standards) community of amateur coders have gathered in forums and IRC chats to assist one another with programming the antiquated hardware with no official support from Nintendo.

Understandably, the relationships between homebrewers and the official intellectual property rights holders are often rocky and ambiguous. Purely from an intellectual property standpoint, homebrewing raises many questions regarding ownership. It's a practice that takes power away from the owners of propriety hardware, and gives it to a class of unofficial hobbyist programmers. The reactions from rights holders are often strange and confusing. For instance, Nintendo has for years permitted the existence of various homebrewing communities dedicated to reverse engineering their hardware, but has stepped in to send cease and desist letters on several occasions. AtariAge, an Atari 2600 homebrew website, apparently received a gag order from Nintendo ordering the removal *Princess* Rescue from the commercial Atariage store. The reactions from the homebrew community to say the least were disapproving, with one user contemplating the consequences for the entire homebrew community. On NintendoAge, user *Slobu* commented, "This has a serious chilling effect on homebrew developers. Gameplay alone cannot be protected. Developer Sprybug used ZERO resources from the game it pays homage to"[4]. Some bizarre copyright situations have emerged from the homebrew community regarding intellectual property. The creator of *Full Screen Mario*[4], a remake of the original Super Mario Bros. in modern HTML5 and javascript, was recently sent a cease and desist notice[5], but a

4 http://www.fullscreenmario.com/
5 http://www.webpronews.com/full-screen-mario-attracts-the-ire-of-nintendo-2013-10

complete and commented disassembly of Super Mario Bros. in 6502 assembly exists on romhacking.net[6]. The only explanation I can think of is because it has removed the pictorial assets from the code, requiring the user to extract the necessary data from a ROM file to assemble the code.

Some consider the attempts of Intellectual Property rights holders to suppress homebrew and other fan games misguided. Homebrewers by their dedication alone to programming for these systems demonstrates a level of fan affection beyond ordinary consumer. They are arguably the inner-most core of dedicated fans for videogames, and blur the line between consumer and producer. I often wonder about the effects of cease and desist letters from companies on these communities. Fans, no matter how crazy, annoying, and dedicated, are a game developer's early adopters. As we know from Everet Roger's research *Diffusion of Innovations*, without early adopters, a product is doomed from the outset.

Meanwhile, homebrewing provides an outlet for a group of people who are very odd, creative, and technically inclined. From my personal observations alone, I've seen many people with a broad range of interests come together over the common love for retro games. To misunderstand and attempt to suppress homebrewers might have unintentional consequences within a company. I wonder if killing homebrew projects might cut off key early adopters and enthusiasm for games products; creating a spiral of negative sentiment through the broader fanbase of a game company, thus killing the morale of fans and so-called early buzz for a product. But I think allowing these communities to prosper can meet the needs of both fans and game developers. Homebrewing is not just some funny thing a few odd people do, it is a creative outlet with a strong community that can keep even a fledgling brand-name alive and relevant. For the current owners of the Atari name for example, homebrew games are one of the only reasons Atari as a brand-name is still relevant for many people.

Speaking of Atari, the creative spirit inside the homebrew community reminds me of why and

6 http://www.romhacking.net/documents/344/

how companies like Atari became successful in the first place. The homebrewing community reminds me of the best days of Atari and other 1970s and 80s Silicon Valley companies. In the early days of the home computing industry, the work was almost completely free. Programmers lacked management and had to be self-disciplined, but had total creative control over their software. In *Once upon Atari*, former Atari employee Howard Scott Warsaw interviews several of his ex-colleagues about the early days of Atari, and presents a lost lesson within the annals of early game programming for educators and managers: people will be willing to work hard and learn difficult skillsets with enough freedom, playful mindsets, and constraints that permit creativity. This mindset is still alive in the homebrew community. If there is one place answering the questions communicators and managers have on how to harness labors of love into work environments, it is within the homebrewing community.

Problem Statement

Years ago when I was a teenager, I tried to enter into the NES homebrew community, but was overwhelmed with the both the high volume and highly technical nature of the documentation. I used NESDEV.com for my programming resources. Programming the NES also meant learning 6502 assembly language, a task I found daunting initially. I tried my best to understand what everything meant, but I barely comprehended what I was doing. I was completely lost when it came to assembly programming. Years later, I took an introductory computer hardware course using Yale N. Patt's and Sanjay J. Patel's *Introduction to Computer Systems, Second Edition* textbook. The authors used a simplified and fictional computer called the LC3 (with its own set of instructions) to explain core hardware and assembly language concepts. The explanations in this textbook I found much more cohesive, and afterward I saw that assembly concepts from one system can translate conceptually to

another. The technical documentation for programming the NES then made much more sense. Speaking from experience, I feel my exposure to NES homebrew helped me with the hardware course. Concepts such as the binary and hexadecimal system I learned going into the hardware course thanks to homebrew. Although I barely understood assembly language prior, I can't help but feel it helped me adapt to LC3 assembly more easily than I otherwise would have. I wonder had the community's documentation followed more professional practices similar to Patt's and Sanjay's book, if I would have learned assembly much sooner? From an educational standpoint, homebrewing for retro computers could be an excellent introduction to computer hardware and assembly. Research done by professors such as Mark Claypool suggests studying software development from a game standpoint greatly improves student engagement. But many issues within the homebrew communities currently hold back this potential.

There exists a subtle alienation of beginners for people who aren't already experienced in hardware assembly programming in some parts of the homebrew community. Much of this comes from the design itself of several homebrew communities. Although I don't believe this is a conscious decision, it effectively contributes to an insider tone. Now, this doesn't reflect all homebrewers in any way; it would be misleading to say nobody cared about beginners. In fact, many do, and many have tried to take steps into helping newcomers. But they are too small, and too far in-between. Breaking into homebrew is difficult for a variety of reasons. Some I've identified in both my and others experiences are:

- Difficult technical documentation to comprehend regarding hardware
- Unorganized documentation, and an overwhelming information overload for a beginner
- A lack of gender diversity. Over 99 percent are male, as seen by refining member searches

based on gender and Alexa data.

- An attitude by people outside the homebrew community that programming for retro systems is useless. Many gamers treat obsolete or outdated hardware with disdain. For a personal example, several years ago I asked for help on how to get started programming an NES game on a non-homebrew board called Gametalk. The user who replied told me I was wasting my time and should just learn C++.

Aside from the lack of diversity, I was completely set back when I attempted to learn NES programming. I don't remember why I didn't join NESDEV's forum in 2004, but perhaps it was because of a sense of alienation. There's no introduction board on NESDEV for new users. I find it interesting that NintendoAge.com, another NES homebrewing community with overlap in membership with NESDEV, does have an introductory board. Maybe there is a sort of fracturing of where newcomers and experts should be? I have tremendous respect for the efforts within the homebrew community to keep alive my favorite gaming system, and I did not want to say there was much elitism in this community, but unfortunately the evidence suggests otherwise. This segregation of elites and newbies is not healthy for any community, especially one that was founded on the principles of common ideas to share. I will return to this in my chapter 4 discussion of my results.

Research Questions

The number one research question from which all our other questions for this study will be derived from is simply, *"what makes a successful community?"* This question is fairly high level and broad; there are many factors to consider what makes communities successful, some simple, and some complex. The questions and points of research that arise from studying what makes a successful

community is a tangled web, with many points inter-relating to each other, and some that on the surface are independent factors of each other. For our study, we must therefore focus only on a tiny segment of possible topics to explore in the homebrew community. Due to the limitations of this study, I shall focus only on one mode of communication: the communications in the homebrew community towards newcomers to the community.

Studying newcomers specifically makes sense because new people arriving into a community are a key component of community sustainability. There also exists an influx of documentation and attempts to aid newcomers in the homebrew community at varying degrees of success, making this an excellent starting point to study communication. A high level of hospitality to community newcomers is necessary to maintain their intake. Long-time members ought to be open minded, patient, and welcoming to first time visitors. In such an ideal community, someone completely new and unfamiliar to the community can pick up on the rules, where to go and navigate, and learn and adapt to community expectations quickly and without being discriminated or ostracized, either directly or subtly, by the long-time and elite members of a community. In communities focused on software development such as homebrew communities, easy access and usage of educational resources are especially important. It makes sense to narrow our focus then onto the different modes of communication available specifically to newcomers. If an educational community or movement wishes to sustain themselves, the treatment of beginners becomes an important focus to evaluate.

The following research questions regarding homebrew as related to beginners emerge:

- What modes of communication are available? What modes are not?
- How user-friendly are these communications for beginners?
- How well written are communication forms?

- Are these communication methods well organized and easy to find?
- What new communications and documentation can be created?

From these questions we will devise a test protocol to see how friendly a homebrewing community is towards beginners. But first, we must discuss issues of community sustainability and some general forms of communication used on these sites.

Chapter 2 – Background

Who are the homebrewers? And what sorts of communications do they use? This section will answer these two preliminary questions, and also provide an introduction for many of my motives for researching homebrewers.

2.1 Homebrew Demographics

In John Vanderhoef's article *Retro Revolt*, he stated in his conclusion, "From my experience, this production culture is overwhelmingly composed of white males in their late 20s and early 30s, men who were introduced to video games as children through the Nintendo Entertainment System and have an emotional, nostalgic relationship to the hardware and games" (Vanderhoef, 8). I can confirm the audience for homebrewing is overwhelmingly male. Of 1527 pages of members on AtariAge, only 9 appear when the "female" filter is applied appear. More difficult to empirically judge is race given the nature of discussion forums and the usage of aliases and avatars. However my personal experiences have been the same as John's so far. That being said, users of many ages exist, as retro gaming is not exclusive to the 1980s generation. Although a more detailed study is required to determine exact numbers, I have encountered homebrewers ranging from teenagers to men in their forties.

When I first began researching Homebrewing in 2011, I created an account on NESDEV and posed a simple question: why do you homebrew? The answers told me a great deal about who homebrewers are and what motivates them. Almost all are geeks for Retro Gaming and hold personal nostalgic value for their system of choice. Almost everyone answered my question discussing their nostalgia for the NES. If homebrewers aren't nostalgic for the system, then they tend to be nerds who enjoy programming hardware at bare metal. 'I like to get to the metal and get my hands as dirty as

possible, know how things work and take something out of it.' The interests in how limitations effect creativity also factor in. Tokumaru told me, 'I have nearly zero nostalgia for the NES, I'm just fascinated by how much can be done on such limited hardware.' One NESDEV user's response I found quite telling and reminiscent of indie gaming itself, 'I can make a game that competes with Super Mario Bros. all on my own, but not Super Mario Galaxy.' From just this short selection of posts, the interest in homebrew on these particular communities is primarily programming based. It is not unusual for one person to code and create their assets by himself.

It is important to know homebrew does not have a focal center or main "hub" community designed for general systems. If one exists, I am not familiar with it, and I searched high and low for one. The closest website to a hub I could find is “pdroms.de,” a blog collecting freeware homebrew products for several systems. But this was a personal blog, with barely any comments; hardly a central source.

2.2 Types of Positions and Distribution

When I discuss "Positions," I refer to jobs homebrewers take upon themselves such as programming, the creation of art assets, composing music, etc. Digital games have the unique property compared to other media of possessing equal importance in both technical and artistic aspects. No digital game is complete without a perfect combination of code programmed at its base and the art assets that lie above the abstract world of computer instructions moving data around. For retro videogames, the challenge of creating game pieces recognizable to humans can be particularly difficult due to high memory limitations and low graphics and sound resolutions. In 2D graphics, a small

animated character is known as a sprite. Sprites on the NES for example are limited to a color palette of 3 visible colors and one transparent color. The NES's total available color palette is limited to 54 unique colors. Additionally, of the 54 colors, only 25 colors can be viewed simultaneously on screen. 64 sprites, each measuring 8 by 8 pixels, can be displayed on screen at once; and across a single scan-line, only 8 can be visible at once. To create larger characters, multiple sprites would need to be separately arranged next to each other and moved together as if they were one sprite. The system also has only 2 kilobytes of memory available on-board. For the Atari 2600, the limitations are even more severe. The Atari 2600 only possessed 128 bytes (yes, bytes without a prefix) of on-board memory, effectively meaning programmers would have to draw the screen manually scan-line by scan-line without the help of a frame-buffer. Graphics for the Atari are thus highly abstract, and to create effective artwork programmers must take advantage of negative space in the background to give the illusion of more color. All this must be timed to the MOS 6507 microprocessor's cycles, a variant of a1975 8-bit processor called MOS 6502 Processor (which is also used in the NES). But the 6507 is even more limited than its brother the 6502; as it runs slower and addresses less memory than the original processor. So far as computer systems go, the Atari 2600 is extremely poor and limited from a pure hardware perspective. More information on the limitations of programming the Atari 2600 can be found in *Racing the Beam* by Ian Bogost. With such limitations imposed, creating games is simultaneously a highly technical and artistic challenge for old systems.

These artistic limitations stretch people's creativity. Many people without technical background in assembly often enjoy the style of these system limitations and create artwork artificially constrained to the limits as an aesthetic. Given the limitations of the Atari 2600 for instance, short artistic demos

such as *Liquid Candy* by Noice[7] are highly impressive. Since creating art is not exclusive to homebrew but heavily related to it, I wondered how many people in homebrewing boards would be interested in the artistic challenges of retro games. Looking at NESDEV's forum, we can see where priorities for most users are by observing post count in five different boards:

Figure 2.1 is a screen-capture of the top of NESDEV's forum (descriptions are cropped for the page to fit). As we can see, the amount of posts in the forum's hardware and programming boards far outnumber music and graphics related topics.

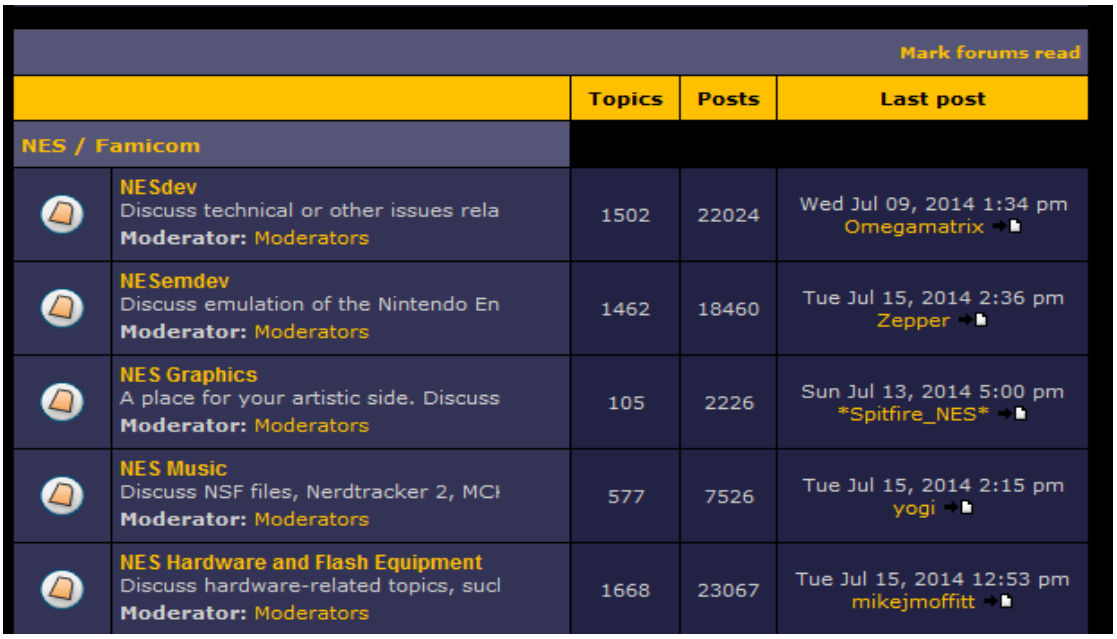


Figure 2.1: A screenshot of the NESDEV forum

2.3 Connection to the mainstream game industry and "Indie" game development

The relationships between the dedicated fanbases who create these games and the official intellectual property (IP) rights holders is complex. Let us first review some examples of how

7 https://www.youtube.com/watch?v=XM1CbBePcdk

companies treat game modification. Relationships regarding modding range from total endorsement, to merely tolerant, to downright hostile. Some developers such as ID Software, EPIC Games, and VALVE encourage modding, to the point where they make their games easy to modify and give many of their developer tools away for free. ID in particular released their source code for *Doom* under the GNU General Public License (GPL), effectivly ensuring the game's modding community would continue indefinitly. Other companies are merely ambivilent, remaining silent on fangames and projects, neither endorsing them nor condoning them. For example, when Nintendo did not release *Mother 3* – a sequel to the 1995 cult classic *Earthbound* - for the GBA outside Japan in 2006, fans of the series took upon themselves to translate the game's text for the international English speaking audience. Nintendo remained silent. It's an uneasy silence, but community efforts such as the *Mother 3* translation project move on without intervention from IP holders. Other fan projects are not so lucky. One Project titled *Chrono Ressurection*, a fan-made remake of Square-Enix's 1995 RPG *Chrono Trigger,* was sent a cease and desist letter by the company in 2004[8]. This is but one of many cases where IP holders have stepped in to protect their copyrights from the very fans who made their games popular.

Within homebrewing communities themselves, the relationship with intellectual property holders can be even more uneasy. As a general rule of thumb, companies like Atari and Nintendo will allow homebrewers to operate, but also remain silent regarding their existence. Cease and Desist notices have occurred though on several occasions, mostly in the name of trademark protection. Nintendo allegedly sent a cease and desist notice to AtariAge for selling a recreation of Super Mario Bros. for the Atari 2600 titled *Princess Rescue*, despite best efforts to present the game as a parody by the game's developer Sprybug. On the other end of the spectrum, a Halo fan-game for the Atari 2600 met praise from Bungie employees. And Atari's founder Nolan Bushnell (who has not controlled the

8 http://www.opcoder.com/projects/chrono/

company since the late seventies) participated in a Q&A on AtariAge. Clearly, the relationships between companies and homebrewers presents a tangled web of bizarre intellectual property questions and variable viewpoints across different developers and individuals.

Figure 2.2 Screen capture of AtariAge.com

The overlap between indie gaming and homebrewing is much stronger. Many independent games pay tribute already to the style of 8-bit games from an artistic standpoint through their visuals and sound. Some famous examples of games made on modern PCs done in this style include *Cave Story, La Mulana, Retro City Rampage*, and *Shovel Knight*. *Retro City Rampage's* developer even designed a scaled down NES homebrew version of his game after completing the PC version. 8-bit throwback games could even be described as a trend due to their frequency lately. Homebrewing for retro systems in essence is an extension of these types of games by working on actual hardware. By definition, retro homebrewers are also independent game developers who happen to work on obsolete hardware.

2.4 -- Available Modes of Communication

Across the homebrew community as a whole, many different forms of technical communication and media of information are present. Common public forms of communication include 1) Traditional Manuals, 2) Web Tutorials, 3) Forums, and 4) Wikis. Although other forms of communication, including private modes such as instant messaging email, these four forms of communication are the most easily accessible to the common user and heavily utilized by all of the homebrew community.

TRADITIONAL MANUALS

Traditional documentation forms exist on several locations throughout the homebrew community. NESDEV's primary purpose for several years was to collect as many different documentation sources possible related to NES development. The main page for NESDEV still exists for people who wish to view standard documentation for the hardware. AtariAge under its programing section[9] lists regular documentation here. This is found rather easily under the google search term "Atari 2600 programming." On the Drunken Coder's homepage, the side bar links to documentation for libnds, a software development library for C++ programming for Nintendo DS[10]. Generally, documentation sources in this form exist as downloadable text or pdf files. Many of these manuals have fallen out of favor in the homebrew community for introducing beginners due to their highly technical nature.

9 http://atariage.com/2600/programming/

10 http://libnds.devkitpro.org/

- Google searching for the following terms; results yielded described
    - NES programming manual: wikibook's guide, and many links to nesdev txt files
    - NES programming: similar results to above; also includes forum posts
    - Atari 2600 programming: yields And-rew Graives of Randomterrain's tutorial for newbies, AtariAge's programming sections *(do these count as web tutorials or manuals? Or is there little distinction?)*
        - Adding "manual" to this gives a pdf link to the 1979 Stella Manual
- Generally these are accessed via main websites

For specific hardware, some sources exist related to chipsets. These are not directly intended for homebrew given microprocessers have broad applications. 6502.org is the "go to" source for 6502 assembly. It requires some hunting to find if you are a beginner to homebrew related to 6502 systems, but once found, is an incredibly useful source of information. If there's a common link for retro hardware, it is the 6502 microprocessor. The documentation on 6502.org is thorough and incredibly useful for learning the basics of 6502 assembly. We will discuss the details of 6502.org later. Perhaps equally important for retro systems is the z80 processor, the second most commonly used chip in 1980s hardware and the processor used for the Nintendo Gameboy. The easiest and most useful source for the z80 is http://www.z80.info/. The documentation on this page is straightforward and similarly organized to the NESDEV main page. We will discuss the implications of this on gameboy homebrew in Chapter 4.

WEB TUTORIALS

Related to conventional manuals, several tutorials on homebrew programming are scattered across the web. *[Include an academic source defining what a tutorial in this context are]*. Web based tutorials have the unique property of being capable of mixed media and hypertext. For homebrewing, tutorials range from (), the forum based "Nerdy Nights" NES Tutorial by Bunnyboy, to conventional web pages as can be found on loirak.com and atariage.com.

- Pater NES Assembly tutorial http://www.patater.com/nes-asm-tutorials (easily googled)
- AtariAge lists several here: atariage.com/2600/programming/ (easily googled)
- Loriak Gameboy Tutorial http://www.loirak.com/gameboy/ (one of the only ones in this case even after so many years)
- http://gbdev.gg8.se/wiki/articles/Tutorials gba dev tutorial. (Homebrewers for handheld nintendo systems tend to move on to the latest system due to similar arcitectures between them. When the DS was cracked, gba dev mostly switched over to DS dev. Nintendo's policy of using legacy hardware for new products has produced a mixed homebrew community that effectivly has the same limitations as other retro systems, but traits and modern political attitudes common to modern homebrew)

FORUMS

What is a Forum?

An online forum, also known as an online message board, is a popular form of communication for specialized topics of discussion. Taken from vBulletin's (a popular message board service) FAQ, a

forum is defined as, "A community bulletin board is an online discussion site. It's sometimes also called a 'board' or 'forums'. It may contain several categories, consisting of forums, topics and individual posts. The content on a community bulletin board is often created by the users or members of the site. They are used for support, to share ideas and for a variety of different topics." The forum is a descendent of the older Bulletin Board System technology of the 1980s, and an ancestor of current Social Networking communication.

There are several core components common to all forums. The most important component is the user base, known members usually. Without users, a forum is pointless and dead. Members must sign up for a forum using a registration page. Upon joining, a user profile is generated, which members have the option of adding an avatar, personal information, and contact information to. Structurally, first exists a main page listing categories of discussion, known as boards. Usually these are organized by tables *(define?)*. Within each board, users have the ability to create posts of text akin to a real-world bulletin board. Posts appear in order of date created with details above or to the side containing information such as the member name who posted and date of creation.

Some message board technologies such as vBulletin have now integrated social networking services by allowing users to log into the forum using their social network ID. Effectively, this eliminates the need juggle multiple accounts together across several different forums.

In the 1990s and up until the mid 2000s, forums were amongst the most popular mode of communication outside of email for general social discussion. Since the advent of social networking services such as Facebook, forums have fallen out of use as the default location for general chit chat. Social networks like Facebook and Google+ contain the DNA of message boards. At first glance, many of the superficial features such as private messages, discussion posts, and user groups are present in

both social networks and conventional message boards. Never the less, forums remain useful for specific, specialized topics due to their inherent nature of clearly defined boards. Forums are thus more controllable and scalable compared to social networks. They also lack news feeds built into the front page, making them better suited for organized, specific topics.

Two particular boards on NESDEV are relevant to this study include *General Tech Talk*, the standard location for NES programming; and the Newbie Help Center, a board intended for helping beginners. NESDEV lacks a specific board containing a set of rules, or a place for new users to post introductory topics. New users are expected to read a short post describing how to fit in, and then read a long getting started guide on the forum describing everything from a list of software necessary to install, to disparate links on 6502 and NES programming. One particular tutorial of interest linked here is *Nerdy Nights*, a forum based tutorial specifically intended for beginners on the related homebrew community NintendoAge. On NESDEV, as far as I could tell, the forum is the only means to access this tutorial.

WIKIS

What are Wikis?

The realms of the online world involving highly technical work require intuitive collaboration to be effective long term. In the early days of the web, it was difficult if not impossible for average users to edit and modify a web page. In the beginning, there was no easy way to collaborate and change web-pages without having direct access to the web server and html pages. To solve this problem, Ward Cunningham in 1994 introduced the wiki, a term coined from Hawaiian for “quick” according to many different academic sources. Wikis are information sources specifically designed for communal based

editing and collaboration. They allow for any user the ability to edit and contribute to the creation of web-pages on the wiki, the editorial process, and usually a discussion page of some sort to facilitate the collaboration. The most well known example of a wiki is Wikipedia. “This convergence of roles occurs because the wiki is a technology (tool) that regulates interaction (rules, division of labor) among participants (subjects) working on communally authored documents hosted on the wiki (objects, outcomes); thus, the wiki supports almost the entire writing activity” (Walsh, 11).

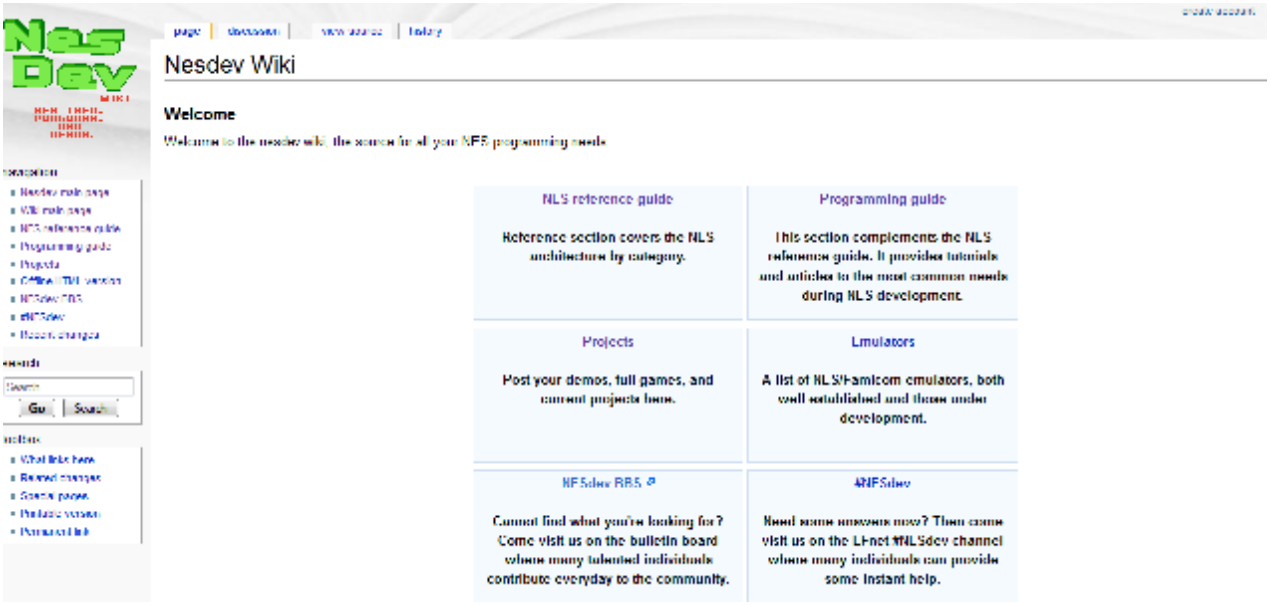


Figure 2.3: The NESDEV Wiki

In a wiki, community is everything. Its audience and content makers are one in the same. Without a thriving community, wikis die. In a sense, wikis are like biological systems in how they live and die. They are prone to attack from or malicious users (commonly known as trolls) and need antibodies (moderators and administrators) to regulate the community. Like genes, the best information survives. Most wikis allow users to register and create a profile like forums in order to easily track their page edits. Though, many wikis allow users to edit pages without registering. Knowledge of HTML is

unnecessary, as wikis contain tools to aid in building pages. Several different wiki software packages exist, each with their own pros and cons.

From a technical communication standpoint, the wiki is a powerful tool for keeping information living and up to date. Wikis are living documents. Information can be corrected, redacted, and revised to control errors and keep information accurate in a way not present in conventional paper mediums without the expensive process of central editors and new editions. In the homebrew community, several wikis exist with varying interest in contributions. Some such as the NESDEV wiki are (or at least attempt to be) a deep part of the community and used frequently in forum posts for documentation and reference. Other communities however, do not have visible wikis, or their wikis have died out unnoticed due to lack of interest.

So far as wiki usage is concerned in the general homebrewing community, their usage and accessibility is highly variable. I have found the NESDEV wiki is relatively easy to find given it's featured on NESDEV's main page and frequents itself in search engines when looking for NES hardware information. On the hand, homebrew sites such as AtariAge have no official wiki. One user talked of making one (found here: http://atariage.com/forums/topic/113051-atariage-wiki/) but I found this wiki difficult to find, and it seems to have since been deactivated due to lack of interest. Although technically not officially part of the homebrew community, homebrew information can also be found on resources owned by the Wikimedia foundation including Wikibooks and Wikipedia. On Wikibooks, several homebrew programming related articles exist for many different gaming systems. Technical specifications of various systems and assembly are accessible for reference on Wikipedia. As we shall see, usage of Wikipedia comes into play very frequently on the NESDEV wiki.

NESDEV Wiki History

In the early days of NES homebrew, NESDEV was the “goto” source for all NES related information. Back then, information was loosely organized, with a large collection of different user guides created by different members of the homebrewing community listed as sets of bullet points. Often, the information within these documents overlapped and made it difficult to effectively find specific hardware information. The approach to learning NES hardware was resembled an old-school engineering learning style: here's a bunch of technical information on the system, here's some tools, now go learn the system. Even for someone with prior experience to systems programming, the documentation was messy and difficult to navigate.

Starting in 2004, the NESDEV community took the initiative to start creating a significantly more user-friendly wiki. Based on archived NESDEV forum posts (http://forums.nesdev.com/viewtopic.php?f=5&t=404&hilit=wiki), we can conclude in the beginning, the wiki was baron and poorly organized. Searching *The Internet Archive* for snapshots of http://nesdev.com/wiki yields no results, thus the early days of the wiki are not possible to study. From archive.org's snapshots of the main page however, we find the wiki was not even linked until

The main web-page still exists for archival purposes, but a disclaimer now appears at the top reading “The docs linked below are old and may contain inaccurate information. For the most up-to-date information, please see the NESDev Wiki.”

2.5 Community, Sustainability, and the Untapped Potential of the Homebrew Community

A community is a group of people with mutual interest, communicating and working together for the benefit of some common goal or shared interest. The health of a community can be measured by

a) its long term sustainability, b) the effectiveness of achieving goals for the broad members of the community, c) the effectiveness of communication between members, d) the connection between leadership (decision makers) and general members (perhaps if all members are involved in the decision making, it is also a sign of health).

How can a community's sustainability be measured? For one, a community's management in relation to resources can be measured as effective. And if a community is retaining, growing, stagnating, losing people, etc. we can also see how sustainable one is. Constant growth might seem like a good measurement of community status and health, but there exists a danger of unmanageable, runaway growth. There are many historical examples of the natural cycles of community, of exponential growth followed by exponential die-off when looking at population in relation to resources. Looking at our global community, much sickness is present in the ways of our social structures. It is not sustainable. For many complex reasons beyond the scope of this discussion, our systems of economics, governance, etc. have come to be structured around wasteful industrial production that contaminates the overall environment and serves more to benefit a small group of individuals (financial elites, wealthy business interests) at the top of a food chain of unhealthy competition. If our global society is to survive long term, there's no doubt many changes must occur at a dramatic level soon. [**Include a reference perhaps?**]

Within the homebrew community there exist the seeds for such dramatic changes to our real world systems necessary for our survival. There is a mindset necessary for programming these small computer systems beneficial to sustainability movements. Creating a game within the crazy memory limitations of only 128 bytes of memory involves both deep critical thinking and a stream of creative thought on how to realize the vision of a game within the severe limitations of the system. For art, the

value of limitations is well known. A well-known quote often attributed to Orson Welles, “the enemy of art is the absence of limitations” summarizes the feelings many have about art and limits (although I couldn’t find a reliable origin for the quote). I argue this mindset of squeezing the most out of a limited system can be applied to more than art. For instance, how could our current energy needs be squeezed into the current framework of alternative energies? Compared to fossil fuels, the energy output for alternative energies like solar and wind are miniscule. The transition to alternate energies will not be easy, but it's a necessary step before we can turn away from fossil fuels realistically. These sorts of problems regarding limitations of regarding sustainability and limited resources require the same methodical and creative thinking necessary to programming microcomputer systems. If the training in this kind of thought could be channeled from homebrew into the mainstream consciousness, perhaps the world could transition to a sustainable future much sooner.

The homebrew movement itself is a community too. Small it may be the same questions regarding sustainability apply as much to it, as does the global community. The internet is full of communities that have failed for a variety of reasons. Homebrew is a community that could be much more than it is, but must evaluate its own sustainability first before its ready to channel these more powerful modes of thought. Are homebrew communities currently sustainable themselves? If not, why? We must study what is wrong with the modes of communication available, and realize what can be learned about the communication aspects in this community on maintaining them. We can then think about channeling what we learn from improving the homebrew community into the real world by drawing lessons on sustainability from inside this community. Then perhaps, it can be applied to our own social structures and institutions in the real world.

Primer on Communication Ethics and Sustainability

In the homebrewing community, it's necessary to discuss the ethics of communications if we are to discuss achieving a user friendly and newbie friendly environment. In this sort of environment. Amitai Etzioni in her study *Positive Aspects of Community and the Dangers of Fragmentation* discusses the communication needs of the community versus individual. “While people survive without communities, the thinner their community bonds, the more alienated and unreasoning they tend to be. Moreover, because for communities to flourish they require that their members not be completely self-oriented, the common good has a normative standing in the same sense that life and health do: they all are essential for our physical and spiritual well-being” (Etzioni, 303).

It's important to factor in the humanistic side of communication that can often be lost within the technical side. A good case study in the ethics of technical communication can be found with Steven Katz in *The Ethics of Expediency.* In his extreme example, he discusses a well-written memo from Nazi Germany discussing the transportation of Jews to concentration camps. Despite its good technical writing, something is clearly wrong here. Although homebrew is not going to kill anyone (so far as we know), it presents itself the question of humanistic communication in relationship to the ease of use for both finding and using documentation.

Communication is highly important to creating successful and sustainable communities In *Community Open Spaces,* the authors describe (page 79) how the common goal of creating a public garden space brought people together and improved communication, including amongst adults to children.

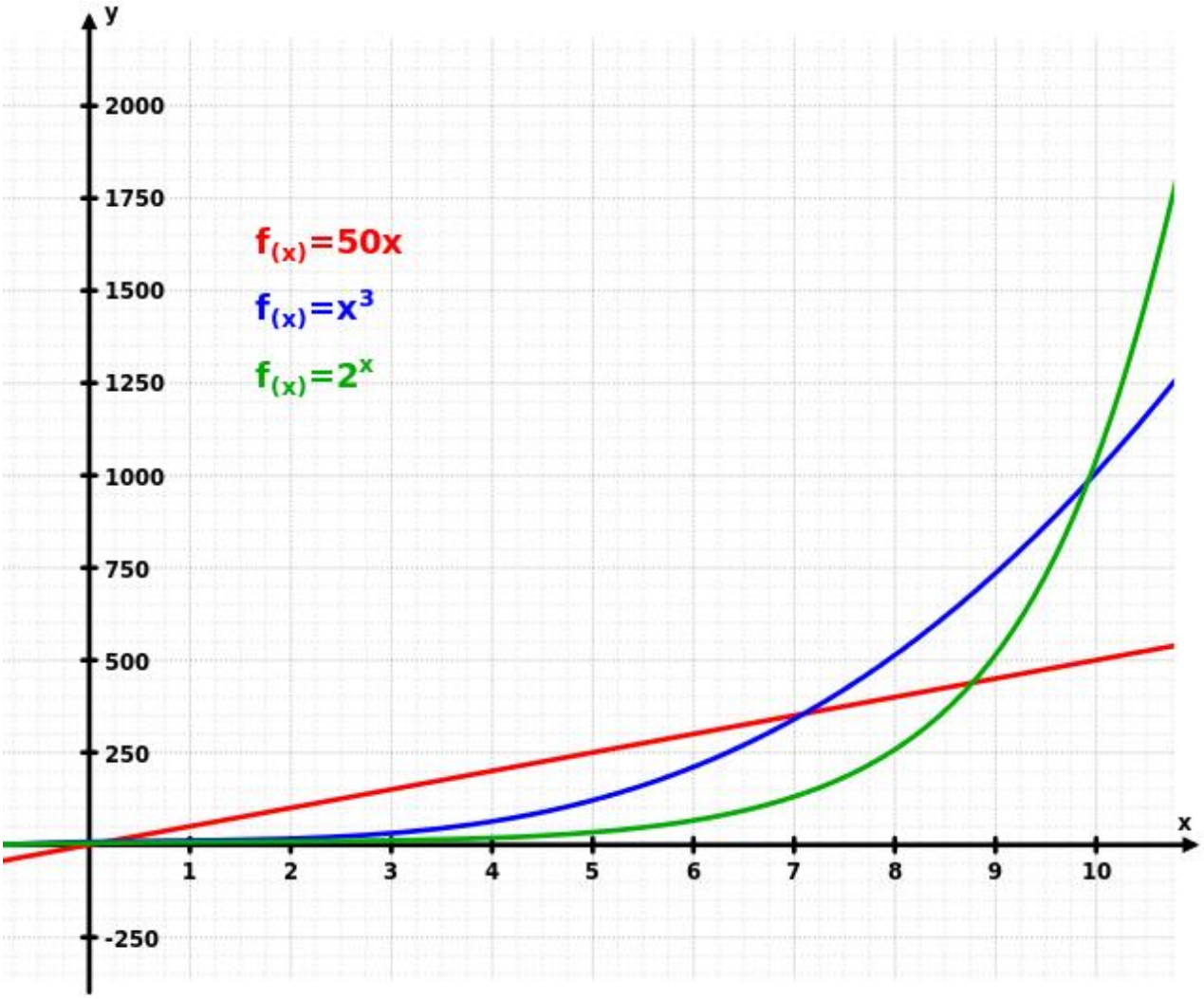


Figure 2.4 The exponential curve, via Wikimedia Commons

Nearly all conventional sources I've read on sustainability discuss Growth or Growth Management. Commonly discussed and accepted is the exponential growth curve and quick bell-curve-like decline that follows rapid growth. Viewing Google Analytics data on various search terms for homebrew (website names, common homebrew terms, etc) can reveal where a homebrew community stands on this curve.

Although conventional, physical resources like clean water do not exist in online communities, many of the same principles can be applied to people themselves and information volume. Brian S.

Butler in his research on online communities demonstrates how in an online community, the main resources consist of the number of members, and their actual pieces of communication and information being contributed to the community. As an online community grows too steeply, discussions become unmaintainable and drowned out by the mass of topics. But equally dangerous, a community can also gain no membership over the long-term, and eventually stagnate. From this perspective, the parallels to communities in real space become clear. But can homebrew communities in the long term really aid in real world problems?

Overview of Real World Problems Relating to Sustainability

The question must arise, even with everything appearing functional on the surface, how come communities can fail? To answer why, one must peer beneath the hood and look at the engine of a community. Should the engine sputter and malfunctions be observed on closer inspection, the time to act in recognizing it must be fixed before breaking down becomes now. For two introductory cases, both in the real world and online, we will look at the global community at large and a smaller online community I was part of called Nintendo City.

The modern industrial system and economy are unsustainable. The reasons are complex, and I cannot narrow every problem in detail within the scope of this project. But we can explore at a basic level the nature of many of these sustainability problems. Many scientists and sustainability experts would agree the most significant threat to modern civilization is climate change. With 97 percent of world scientists in tantamount agreement on anthropomorphic climate change, climate is a crisis that must be addressed by any means necessary and be open to any suggestion for improvement. Related to the burning of fossil fuels is resource depletion. Industrial nations are also highly wasteful. For

example in *Designing for Zero Waste* in chapter 7 (page 138), Ab Stevels et al Describe the dismal state of electronic waste in Europe and Japan. In 2008, 13 percent of electronic waste ended up in landfills, and 54 percent was sent to substandard treatments; in Japan, only 50 percent of electronic waste wound up in the official waste treatment system. Meanwhile, the arrival of Peak Oil by 2030 according to Grahm Turner from MIT and the Club of Rome will bring economic devastation. Even if we ignore all this dismissing it as wrong, we still have to worry about environmental and Eco-system devastation. In a world of finite resources and energy, the only sensible solution is to conserve the energy we have by massively scaling back growth, and look for creative ways to use the resources we have already mined and created. Within the home-brewing community, answers on how to achieve this ambitious goal lie in plain sight.

Big changes should be happening now to society. But within our current systematic paradigm, the leaders with the capacity to allow significant change bend to the will of a small group of wealthy financial elites. This phenomenon has been documented by several people in academia, including Thomas R. Dye (author of *Who's Running America?*) and by Linguistic Professor Noam Chomsky (I would recommend highly to listen to his lecture *Surviving the 21st Century*). The power structures of government and democracy have been inherently corrupted, leaving the vast majority of citizens in democratic republics with little to no power compared to the elite classes. The solution to change thus must come from outside the system, from people themselves in a worldwide (peaceful) revolution to take sustainable practices into their own hands. What's needed is the opposite to current paradigms of information control and hierarchical government. To change or create a new system, new modes of self-governance amongst tightly knit communities and standards of open communication free of government secrecy and the red tape of incomprehensible copyright (that at this point only benefits

large corporations) is needed. Again, the homebrew community contains lessons and answers on how to achieve these goals. When I began this project, I set out with a related research question to my problem statement on how the mindset needed to program homebrew systems might be utilized for real world community problems. I even devised an idea for a serious game called *The Sustainability Game*, which I wondered how people in the homebrew and maker communities would fare against people who are not. Although these ambitions quickly went beyond the scope of this project, I've chosen to leave remnants of the original report within this project's appendices so they may one day be continued. For now though, looking at problems with the global community remains a good exercise in how and why systems can become unsustainable.

For a more similar case in scale and subject matter to homebrew communities, let us look at a website called Nintendocity. Nintendocity.com is an online Nintendo fansite founded in 2003 by Joey Anderson. It is not a home-brewing community, but falls within the category of fan and user generated content. Primarily it specialized in sprite comics and fan fiction all related to Nintendo. Contributing to the site in the beginning meant manually emailing Joey and waiting for him to update the main page. The site was completely non-commercial under Anderson's ownership.

Joey Anderson had started Nintendo City as a hobby, and thus everything about the website exhibited a hobbyist, personal, amateur style. The website design was a sloppy blue, web 2.0 design Joey Anderson made himself. But it was highly functional and easy to navigate. The forums used the open source SimpleMachinesForum software, adding to the amateur feel. It may not have been the most high tech site, or polished looking site, but it had a unique charm to it. The site felt homely and personal. It in essence belonged to Joey, who had originally designed the site to showcase his own comics, and eventually realized the awesomeness of contributor content.

Figure 2.6: Screen-capture of NintendoCity

The reason I have chosen to discuss NintendoCity as an example of a failed community is because I was heavily involved within it. I first joined Nintendo City in 2006 and used it as a vehicle for getting feedback on my early sprite comics and Nintendo fan fiction. I later delved into game reviews and journalistic commentary on Nintendo. Working with Nintendo City as a contributor fostered the creative side of me during a time when I knew nobody in High School as obsessed over Nintendo as I. In 2007, I was appointed the position of a global moderator. Suddenly I had found myself thrust from ordinary regular poster to a person with power and responsibility.

Around the same time, I began to notice as NintendoCity grew, Joey would not be able to keep up with updating the website. I began routinely suggesting giving administrative permissions for the website to the forum's appointed administrators, and for Joey to look into installing an auto-content submission portal similar to Newgrounds.com, which I had known from experience also started off with the owner hand picking content and later switching to an automatic system as it grew. In the meantime, I suggested he schedule a single day to dedicate updating the site opposed to his sporadic updates. Joey's regularly scheduled updates helped for a little while, but were not enough to sustain the website.

In 2008, Nintendo City's ownership changed unexpectedly, creating a downward spiral that would eventually leave the community in shambles. Joey Anderson, overwhelmed by maintaining the site consistently and rising server costs sold the site to an administrator known as Chris. He changed everything about the site without consulting the community, being more interested in financial opportunities than NintendoCity's intended goal as a shared hobby project. Gone was the old homely light blue and SNES sprites, now replaced by boring, clean, ripped off images of official rendered Nintendo CG promotional art from recent Mario games. Also gone was SMF, replaced by the expensive and proprietary vBulletin message boards. Chris in the midst of the transition gave me administrative power. Suddenly I was tasked with the disgusting dirty job of cleaning up spambots and pornbots; a problem we never had under simple machines forum. We all hated Chris for the changes he made to the site, but in hindsight I see his motivations. He was interested in a much bigger vision for Nintendo City; perhaps a commercial fansite on the level of GoNintendo. But we as members were not welcome to such a rapid and sudden transition. Needless to say, none of us staff or the members were happy with these changes and abandoned the community to start our own forums; which all subsequently died for many reasons.

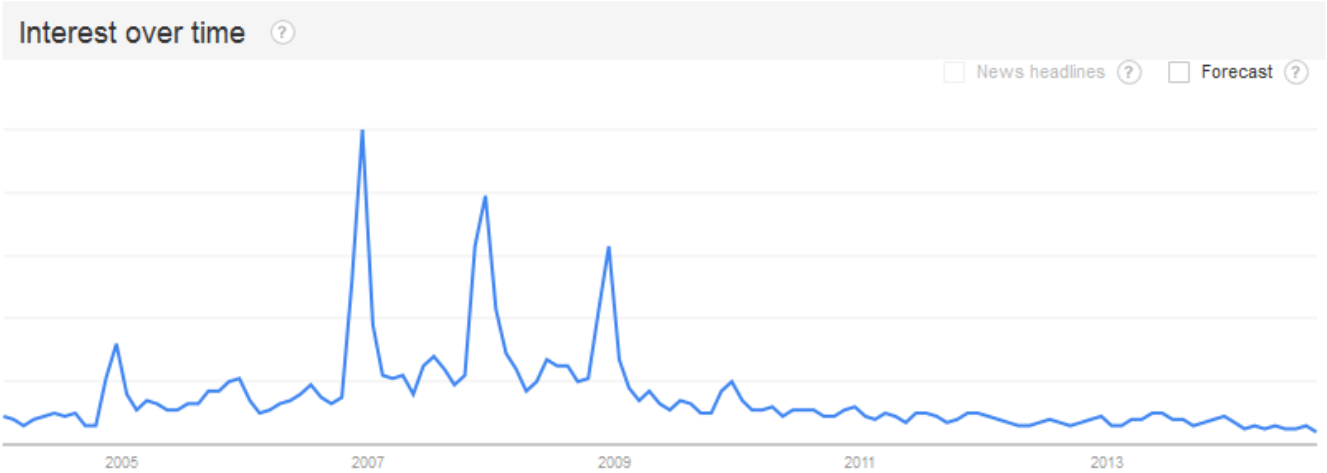
The site would be sold again to another person who I only remember by his first name Scott. He didn't last long, selling it to a new admin with the alias TradeDemon, commonly abbreviated to TD. At first, this third owner restored the site to the former functionality and for a while, Nintendo City regained its user base and the forum became active again by its innermost core members. But TD’s motivations were also financial. About a year later, TD made the same exact mistake as Chris. First the SMF forum was overrun by pornbots (a task that would've easily been averted had TD installed a captcha or manually activated accounts), and he replaced the forum again with vBullitan and generic

CG Nintendo art. We weren't happy, but the site kept updating and we remained active, until eventually TD neglected the site as more members left due to life and at the eventual loss of interest by the owner.

Today Nintendo City still exists, but is effectively dead. In what I can only assume was one last desperate attempt to get members back, TradeDemon switched the website back again to its old layout. All updates from 2008 to 2014 are now erased. Not that it mattered; the last update I recall under TD's ownership was in 2011. The forum is completely dead; its members moved on, many having gone their separate ways, grown up and gone to college. A new generation for Nintendo City never came. And how could it have when the reason why people visited the site in the first place was killed? Perhaps we could have done things differently. Maybe in hindsight we were too interwoven; too self-absorbed within the core group of members to think about attracting new members. But I think the ownership changes and the forced direction they had against the community's views of Nintendo City are fundamentally what killed the site. After Joey sold the site, Nintendo City stopped growing, and died a slow, painful, and unnecessary death.

As a former contributor and staff member of Nintendo City, I cannot help but consider the actions of the owners after Joey Anderson to be greedy and petty, selfish in their motives, and the main cause of the site's demise. As an objective, critical researcher, I must also consider how other gaming communities went through the same or similar transitions as Nintendo City did, but wound up successful. After all, communities such as Penny Arcade and Nintendojo have gone through significant changes over the years and still retain their user base. I suppose the lesson from all this is that after all these years, Charles Darwin is still right: it's not the strongest or most intelligent organism that survives sudden changes; it's the most adaptable organism that does. If there's anything I hope the homebrew community (and other communities for that matter) takes from my personal account of Nintendo City,

it’s that sudden and unexpected changes do occur. And without open minds and quick thinking, communities can fail to adapt.



2.6: Interest in Nintendo City shot up and then declined significantly after the site was sold

Although not a homebrew site, Nintendo City lies within the same overall area. I want people to know how seemingly stable forums can go horribly wrong suddenly, but looking underneath the surface, the problems become visible. Nintendo City was doomed as soon as Joey Anderson's interest in maintaining the site dwindled. The lesson here is to peer under the hood, and not fear what's inside. If the engine is breaking down early, its best to fix it early and not wait before it’s broken beyond repair; or at least recognize that it will be beyond repair soon and action must be taken.

If I had known back then about shared ownership, or had we regulars been just a little older, we could have taken over the site instead of Joey selling it to a random person without the best interest in the community. The poor communication between staff, forum regulars, and the owners lead to discord. The power of passion is far more powerful than the power of money. Greed I feel lead to Nintendo

City's downfall when it was completely fine the way it was. It’s akin today in how Nintendo themselves are increasingly angering their early adopters user base (I could talk all day why). Homebrewing as a trend seems to be in decline based on google trend data evidence. However, if this trend is reversed and Homebrew ever gets big, I fear the cease and desist letters will be thrown around.

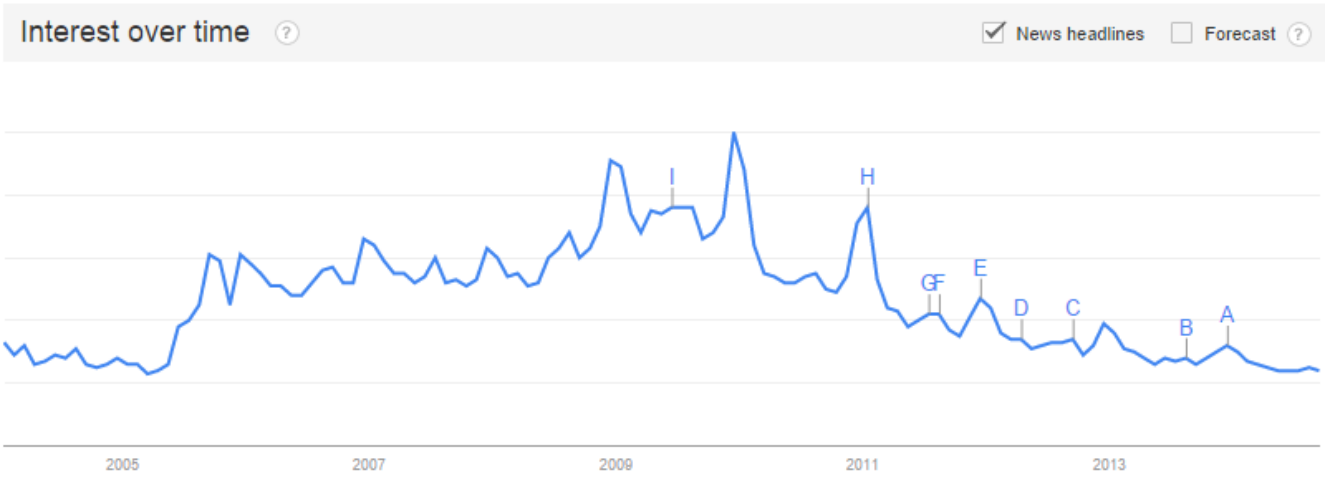


Figure 2.6: Google Trends data for Homebrew seems to indicate it has passed its peak. Typical of interest in trends, it follows a Gaussian Curve

Chapter 3 – Methods

One way to study how easily newcomers can access a community and its information is to study the community from a usability standpoint. Inherently, the web-based nature of these communities causes them to fall in the vein of human-computer interaction. There are many principles and guidelines that have been developed over the years for good web design. In *User Interface Design and Evaluation* by Stone, Jarret, Woodroffe, and Minocha, seven principles are listed for good web design. These include High Quality Content, Often Updated, Relevant to User Needs, Ease of Use, Unique to Online Medium, Net Centric Corporate Culture. Aside from corporate culture (at least from a literal interpretation), these are good starting points for devising a usability test. In the case of a hobby programming website, it's especially important for it to be often up to date on relevant and clear information, as well as relevant to user needs. In our study, we will look especially close at the communication needs for a beginner. Testing the user interfaces of homebrewing websites and the ease of finding information is useful from a communication standpoint. If a new user to a homebrew website has difficulty finding the necessary information for getting started, it will be clear evidence of a gap between the community managers and beginners.

But we also need to test for comprehension. Even more useful to communication is how well users understand the documentation necessary to engage in homebrew. Thus we also must design a set of questions in mind that will test how well the user understands. I developed a series of questions . For my test, due to time and resource constraints, only a small number of users was selected. It may be considered more useful as a preliminary study for future academics interested in researching homebrewing. As said in my problem statement, the main concern of this project is attracting

newcomers to homebrew communities. The test may then serve useful if adapted by other amateur programming and homebrew communities as a method of testing user comprehension.

My test is divided into three sections: a pre-test, the main usability test, and a post-test. In the pre-test, users fill out a series of questions to create a profile of them. The questions were drafted based on relevancy and usefulness to how, such as relation to the gaming and modding communities, programming experience, etc. The usability test itself is divided into two main sections. First, users will be tasked with a user experience test on nesdev.com. Their set of tasks will be navigational; users will be tasked with finding various pieces of important web pages and documentation. From here it can be inferred how friendly NESDEV is towards beginners primarily on their UI design. Next and perhaps more importantly, the bulk of communications work will take place. Users will be asked to complete math and assembly language problems based on the documentation they have found. Failure to complete these tasks, or problems with completely them will indicate communication problems with the documentation they find. Users will be tasked with. Finally, users will fill out a brief post exam to collect any more relevant data.

3.1 -- List of and Justifications for Questions

3.1.1 Pre-Test Questions:

What is your age and What is your gender?

I ask about age and gender per usual with social science studies. In this particular case, age and gender will be a useful comparison regarding generational affect towards the time period of retro homebrewing, and differences between genders in relation to hardware and software concepts.

Do you play video games? How Frequently? What are your favorite games? Favorite Genres? Favorite Generation? What games systems do you own?

I ask this series of questions to gauge the user's interest in video games. If the user is a gamer, their likelihood of being potentially interested in form of homebrew or game modding increases. Although being a gamer does not necessarily reflect programming aptitude, a potential interest in homebrew communities emerges when combined with later questions in this pre-test regarding programming. Asking about genres, favorite games, favorite generation, and game systems owned can also paint a picture of the subject's affinity for retro systems. If the participant chooses earlier generations of games, they are a likely candidate to fit in well with retro themed homebrew communities.

Do you consider yourself a gamer?

Due to the close ties homebrew communities have to the gaming community, I wanted to see if the participant self identifies as a gamer. There may be some cases where users might not frequently play video games but still considers themselves as a gamer, or plays certain games frequently but does not prescribe to the label. I imagine if a user identifies him or herself as a gamer, she or she will feel more comfortable with the homebrewing community.

What is your occupation/major/etc?

I ask about occupation and/or field of study to gague aptitude for programming and attraction to video games. I surmise a more technologically relevant field of work or study will assist the participant

in understanding the documentation on homebrew fields compared to non-technical fields. If both technical and non-technical users perform poorly, it will indicate the faults of the homebrew community's documentation.

Have you ever used a programming language before?

This question will indicate how well the participant is prepared to dive into an learn coding for a new system. Although clearly relevant to ask, on its own this question is not enough. Thus we also ask the following question:

What level would you rate your programming experience?

Users are instructed to circle one of the following options: professional, student, hobbyist, beginner, or none. These criteria cover the basic demographics of various programming skill levels as related to the potential audience for homebrewers. Although not a direct measurement of skill level, it gives a general idea of where the user is currently in terms of coding experience. Combined with other questions, a picture of the user's skill level can be painted.

Do you have any experience in Assembly Language?

I ask the user to gauge the amount of time they've put into learning assembly. The nature writing software for ancient hardware necessitates the need to learn assembly in the long term. In the NESDEV community, learning assembly is pretty much required. Prior assembly experience will no doubt factor in how well the user perform on the questions related to assembly.

What high level programming languages do you know?

Although prior assembly experience is the most useful asset necessary to dive into homebrew development, having high level coding experience can aid in learning assembly (citation). Although high level languages hide most (or all) of the hardware abstraction present in low level languages, many of the same core programming concepts such as control flow, logic, iteration, loading and storing, etc. are the same or similar. If a user has experience in a higher level language, it's necessary to gradually take the him/her down the abstraction ladder (citation) rather than thrusting them into assembly with no guidance.

Do you intend to learn more programming languages?

Ideally, a programming centric community should encourage others to persue their interest in programming. In the post-test, we ask if the subject if they feel encouraged to continue learning to program. If he/she answers yes to this question and no to the later post-test question, we will have an indicator of how encouraged or discouraged he/she feels. This question also distinguishes between users who answer yes to having programming experience but have settled in on their programming languages, and those who wish to keep learning more about programming.

Do you associate "homebrewing" with alcohol or programming?

This measures the user's familiarity with the homebrew community and the jargon used to describe themselves. If they circle alcohol, they've probably never heard of the term "homebrew" used in a programming context, regardless of their occupation or hobbies.

Do you have any experience in homebrewing? and Do you have any experience in game modification?

I suspect users who answer yes to either of these questions will have a much easier time with this test, as they'll have experience with the types of esoteric coding practices and hacking necessary to shrink the learning curve. I also ask what systems and games they've modded/homebrewed for if they answer yes in order to determine how close to the NES they have programmed.

Emotionally, how would you react to the following terms? (ie. anxious, happy, indifferent, etc.)

For a test of this nature, measuring affect is useful. (Pull in a citation?). I picked a small list of words based on how close their ties to the homebrewing community are. They are: Programming, Coding, Assembly, Machine Code, Nintendo, and Gaming. Each of these interests are necessary to fit in with the homebrew community, particularly the NES homebrew community. Measuring affect during testing is highly useful because it tells information about the user's confidence levels. Affect testing is often used in game playtesting, as seen in Fletcher et al's MQP Shattered Sky. For a test of this nature though, some affect testing is desirable due to the notorious reputation of assembly programming versus the general happiness and nostalgia of the computers these terrifying programs run on.

3.1.2 -- Usability Test

The usability test serves two-fold purposes, both related to my problem statement on making homebrew communities sustainable. The goal of this project is to attract newcomers. Thus the test is designed with the following in mind: 1) Measuring qualitative data on the user experience for a

homebrew community including navigation and documentation comprehension. 2) Developing a framework for the communities to build off of.

Due to time and scope, I only designed the study specifically for NESDEV. Although I wanted to test other homebrew communities including NintendoAge, AtariAge and GbaDEV, designing such a study would have demanded far too much on the subject. It also would have overwhelmed test participants quickly had I asked them to learn multiple systems in one straight test. In the future, I am particularly interested in modifying part of the test to focus on Atari 2600 homebrew and BatariBasic. As seen by the trend data discussed earlier, the AtariAge community is much healthier and larger compared to NintendoAge and NESDEV combined. I hypothesize BatariBasic plays a key role for why more homebrew content is produced for the Atari 2600. In my experience, I found finding tutorials, documentation, example code, and information on modding example code for BatariBasic much simpler and streamlined compared to the same tasks in 6502 assembly. Thus designing a modified test for AtariAge and BatariBasic in the future would serve as an interesting control and comparison compared to programming the NES in 6502 assembly.

Design for the exact method of this study was built from iterative pilot tests. Three users were selected and the test and my protocol modified for each iteration based on problems with the design of my tests. These same users were later re-tested under the more refined method. In the initial tests, I wasn't sure how often I should intervene for users if they became frustrated and required help. I frequently and sporadically jumped in to help users in these early tests, potentially contaminating data. By the fourth iteration, I codified into the test to only intervene if asked specifically for help, or under two conditions. 1) In the pilot, it was not uncommon for users to chase an endless path of hyperlinks and become thrown off onto external sites. Because of this, I decided for the final iteration that if users

were greater than two pages deep to intervene. 2) I intervened if the subject could not locate the *Nerdy Nights* tutorial from NintendoAge. Pathways to Nerdy Nights are limited to the forum inside obscure links, but the tutorial contains valuable documentation and example code for beginners. Some similar example code though does exist elsewhere on NESDEV; thus if a user used a different tutorial and did not find *Nerdy Nights*, I linked them to it as an option.

The usability test will measure first how effective it is to find necessary documentation, and then ask the users to perform a series of NES programming related tasks. Part one of the study deals with website usability for NESDEV. First the user is tasked with navigating the wiki. Then they navigate the forum. For each section of NESDEV, a checklist of different web pages and documents will be presented to the user. Their task will be to find these pages. As the user completes each task, they will check it off in a tick-box. If the user asks for help in completing a task, it will not be check marked.

Part two of the study measures comprehension. Using the documentation they found from the forum and wiki, they will be tested on a variety of elementary tasks necessary to assembly programming. These include various binary numeral system concepts, hexadecimal conversions to base 10 and binary, programming simple routines in 6502 assembly, and modifying a file of assembly code intended for making a character on the NES move. I found a tendency in the pilot studies for users to being to suffer burn-out during the assembly programming portions the exam. If after completing the beginner tasks the user feels overwhelmed by assembly programming and does not wish to peruse, they are given the option of ending the study early.

I used a think-aloud protocol similar to ones described in several usability textbooks such as chapter 17 of *User Interface Design and Evaluation*. Users are instructed to speak out loud as much as

possible, though are given the option of staying silent if it hindered their thoughts. At minimum, they will be asked to verbalize any difficulties they might run into. The test practitioner will remain mostly silent unless asked for help and sit next to the test participant taking notes on their progress. Note-taking will be based on personal comments subjects make on the test, observed areas of difficulty, and indications when a subject asks for help. Using a speak-aloud protocol is useful because it indicates to the test practitioner not just what the user is doing, but if they understand why and how they are doing a task. Linda McIver speaks well about the advantages of speak-aloud, "As Eisenstadt and Lewis (1996) point out, 'Symptom and cause are not the same thing.' Knowing what the students did does not mean knowing why they did it " (McIver, iv). For programming and mathematical topics, it's common for users to fall into the pitfall of following a procedure to complete a problem, but not understanding why the procedure works or the underlying concept of the mathematical topic. Since my usability test will be math and programming heavy in the second half, speak-aloud protocol is incredibly useful.

Find the Wiki's getting started guide

On the wiki, the first task a user should complete are tutorials for beginners. NESDEV's wiki contains a getting started guide linking to various wikipedia pages related to assembly programming. Although it's recommended users read all the documents, for the test I selected only a sample based on core concepts necessary for assembly programming. This may limit the intent of the documentation, but I didn't want users to waste their time on topics I would not be testing them on. I selected the samples based on how essential they would be required for programming in assembly. The getting started guide contains lots of information on concepts a beginner probably does not need to know right away, such as ring counters, digital audio programming, bank switching, and MS-DOS command line

information. I narrowed down the topics to the barest preliminary essentials necessary for assembly programming. The topics to find from the getting started guide include:

Find out what a "bit" is

Find out how to add binary numbers

Find out how to perform Boolean Algebra

Find out how to convert numbers between base 10, 2, and 16

Find reference material for programing the MOS 6502 Processer

I decided to leave this question as a stand-alone option even though a wikipedia article on this topic is listed within the getting started guide. There are several pathways to go about finding information on 6502 information. I wanted to see what pathway on the wiki users would be most attracted to.

Find information on the NES's CPU (central processer)

Find information on the NES's PPU (picture processer)

These topics were in the general reference guide next to each other and contain essential information on the NES's memory locations for game logic and graphics programming. The CPU section also contains information on the 6502 processor's instruction set, thus it serves as a redundancy check to make sure users don't only use the wikipedia article under the getting started guide for reference material.

After the participant finishes finding information on the wiki, we task them with finding information necessary to complete the test questions that's accessible primarily through the forum.

Find the new users board

Find the getting started guide

Find a tutorial intended for beginners

Once these three tasks are completed, the subject will have access to a list of topics neccessary for beginners to learn. The test practitioner will have already set up the development enviornment information listed from the getting started guide, so it will be unnecessary to task users with downloading and installing software. From either the tutorial(s) they discover, or other segements of the getting started guide, users will be tasked with the following:

Find information on programming in 6502 Assembly

Find example code for making a character sprite move

Find information on how to change the NES background color

Find information on how to change the NES sprite pallete

In the second part of the usability test, subjects will be tasked with answering various questions related to elementary math and programming topics necessary to assembly. They will be instructed to "use the documentation you found in part 1 to answer the following questions." The beginner level tests basic mathematical concepts including binary and hexadecimal. Afterward, users will be tasked with programming simple routines in assembly such as loading and storing data into memory, transferring information between registers, and some simple control flow. Finally, users will be asked to modify and assemble an asm file that results in an NES ROM that makes a character move.

Beginner Level

1) Add 0010 to 0011 in base 2

2) Do a left bit shift on 0101

3) Evalute:

    1011 AND 0111

    1100 OR 1010

    NOT 1000

4) Convert these hex values into base 10 and base 2

$09

$0B

$1F

Write the following instructions in 6502 ASSEMBLY.

Check your code with the provided debugger.

1) Add the hex values $2 and $10

    -Now store the result in memory locatiom $00

    -Load that memory location into the Y register

2) Load the X register with the hex value $2F and transfer it to the accumulator

3) Push $01 onto the 6502's stack

4) Unconditionally jump to a different label in [Program shown below]

5) Initialize the X and Y registers to $00 and $10 respectivly, then

6) Use labels and 6502 compare instructions to create a loop that terminates after 7 iterations

Use Example Code you've found (or are provided) to modify the asm file

1) Compile the example code to make a character move

2) Change the background color

3) Change the color of the sprite

4) Change the input parameters so that the character's movement is inverted

5) In English, what the programming instructions in this example are doing.

3.1.3 --- Post-Test

As with any post-test, mine must complement the pre-test and usability test to debrief the subject and collect final data needed to tie everything together. After a highly cerebral test, the user will be unlikely to want to sit through a series of difficult post test questions. Thus my post test will not ask too many questions, and they should be relatively easy to answer.

Did you feel overwhelmed by the amount of information present on the website or the documentation itself?

One aspect of the homebrewing community I've found has remained consistent in my personal experience is the lack of cohesion and overwhelming abundance of the information presented. The usability test itself can gauge if a subject feels overwhelmed, but it's nice to have direct confirmation.

Did you find the documentation well organized, easy to understand, and easy to find?

Were there any parts of the homebrew website you found confusing? If so, why?

These two questions will allow the user to put into their own words how they felt throughout the

user experience. Although the test practitioner should take notes and observe difficulties,

Do you feel encouraged after taking this test to continue learning about homebrewing, game modification, and/or game programming?

My main concern throughout the project has been the sustainability of the homebrew community. If the user answers no to these questions but fits the profile of a user interested in homebrew, it'll be an indicator of problems.

Did you feel a need to rely on the test practitioner to help you? Why or why not?

Having a personal tutor navigate through difficult programming and mathematical concepts is obviously much easier than attempting to be self-disciplined and learn the concepts through self-teaching from technical documents. Not many people even through the internet will have this luxury. They must therefore rely on good technical writing practices to produce the most clear and concise explanations possible. If the subject needs to rely on me for the difficult parts of the test, the writing must be confusing.

When you were tasked to compile and modify the asm file, did you understand why your changes were effecting the program?

One common problem with programming tutorials is something I like to call the "code monkey

effect." In these sorts of programming tutorials, users don't really understand why the code works and more often than note are blindly following instructions and are just copying/pasting code. The quote I used earlier from McIver applies well here too. In the test segment where the user is tasked with modifying an ASM file, it may be tempting based on the tutorial format to change the file without realizing why the changes occur. The user must understand why the changes made to the program happen in order to truly master programming the NES.

Chapter 4 – Results and Discussion

A total of six participants were selected based on differing ages, genders, and technical background. Test spreadsheets and more general results can be found in Appendix E and F. Testing yielded several interesting points of discussion on how accessible the NESDEV community in particular was towards beginners. These initial findings seem to indicate a direct correlation between the technical background of users and test performance. Users from conventionally technical fields consistently outperformed non-technical users. All users however seemed to indicate some trouble with the design to some degree with NESDEV. Nearly all gave up at the assembly portions of the test.

Please be aware these test results are highly preliminary as a fact. Since my research was also limited to NESDEV, not all data ma y necessarily be applicable to the entire homebrewing community. However, I suspect many of the same general trends I observed from NESDEV will apply to other homebrew communities.

4.1 – Programmers v. Non-Programmers

Some of the most interesting results stem from how the occupations of the test participants affected test results and interest in the examination material. Although our test participants were limited to only six people, their backgrounds were diverse enough to provide reasonable comparative data relevant to each other. We had two people specializing in programming (one game programming specifically), a dentist, a music teacher, a psychology major, and an artist. Thus we were able to measure a range of technical, semi-technical, and completely non-technical fields.

Our two programmers liked the material the most out of all the test subjects. This suggests

homebrew communications are especially intended for a technical audience. Participant 1 was the only one to complete the assembly portion of the test. This is not surprising, because in his pre-test he listed off having experience with MIPS, ARM, and x86 assembly. Though it's worth noting initially, he mistook based on the 6502 instruction documentation that the processor was stack based. He also thought the LDA instruction stood for “Load Address” based on his prior experience. Because of these two assumptions, his initial code was wrong. But he caught his mistake fairly quickly and corrected it. He gave up after the initial assembly portion and did not modify the Nerdy Nights ASM file. He certainly could have done so, but the test burned him out by that point. Based on 1's experience, I can conclude my test might have been too long. However, the nature of NESDEV's documentation does not encourage users to pace themselves. The requirements necessary to get oneself even into modifying an assembly file first require one to learn binary and hexadecimal, then 6502 assembly, and then set up a development environment. In my personal experience, once one learns the basics and acquaints him/herself with a programming system for enough time, the difficulty in programming drops tremendously. But the initial threshold and learning curve is quite high. 1 also commented he thought a beginner would be completely lost several times. He also thought the wiki was poorly designed, saying, “The wiki is the knowledge base. It shouldn't be random disparate [expletive].” Although he enjoyed the math and programming questions on the test, he commented he would not be interested in continuing game modification or homebrewing based on his website experience.

Participant 5, the other programmer, specifically is studying game related programming. He will be taking a course involving x86 assembly this fall semester. He is a self-identified huge Nintendo fan and has somewhat of an interest in learning more about ROM hacking and homebrew.

He found the hex and bin stuff confusing at first, but persevered through it and got every question right

without much help. But he took one look at the assembly portion and he decided his brain had enough at this point and didn't want to work on assembly more.

He found the documentation confusing and overwhelming for assembly. After he finished the test, I linked him to “easy 6502,” a fairly recently written 6502 assembly tutorial that I have not seen added to the NESDEV wiki or getting started guide yet. I did not collect data on this portion since the test was over, but I wanted to get his comment on this new piece of documentation since he seemed to have a genuine interest in learning more. He said that he was too fatigued to go through, but found its step by step and interactive approach much more understandable. He commented that he might check it out later when he was less tired. Also, he said in his post exam he was interested in learning more about homebrew.

Less technically inclined participants had more trouble with the exam, indicating bias in the homebrew community’s communications towards non-technical users like artists and musicians. Undoubtedly, this is a problem because as important as programming and technical ability to designing a game is, equally important is the artistic presentation to the overall experience. Subject 4 is a senior studying game development with an artistic focus. She Found the navigation fairly simple overall. However, she also seemed to run into some problems with the design. These were the same problems other users encountered from using wiki. She had initial problems locating the getting started guide since it was split between two sections, and complained about unfinished material. Otherwise, she had little trouble navigating the wiki and forum for the remaining tasks, suggesting familiarity with the interfaces of wikis and forums. In hindsight, I probably should have asked users to rate familiarity with wikis and forums, and may if I continue these tests. When she reached the math problems, her performance plummeted. She did not like the mathematics portion at all. In fact, she gave up at the

hexadecimal conversions and commented, “math is not my strong point.” This artist performed the weakest out of all the participants. Although not surprising for a right-brain dominant thinker or anything to be ashamed of, this does pose problems with the community, because artists are important to the game development process and will be scared away at the sight of difficult math. Although it may pain many homebrewers to hear this, effort should be taken to balance out the programming centric focus thus far on homebrewing communities and look for non-technical solutions for discussing art and sound constraints with artists and musicians.

Participant 2's test was an interesting case; she is a musical person with somewhat of an interest in learning programming. Even though she didn't like the assembly and gave up at this part, she enjoyed the mathematical parts of the test. She commented. However she also asked me personally for help frequently during the hexadecimal portions. She did not understand the underlying concept of what it meant for numbers to exist in different bases. At first, she completely misunderstood the concepts and answered the questions wrong. She eventually got every question correct with my help, but I left her initial (wrong) answers in the spreadsheet to indicate her initial difficulty. Never the less, her interest in programming remained. Participant 6 on the other hand had studied BASIC programming in the 1970s on an ancient teletype terminal. He found the assembly portions unlike BASIC. I recall years ago reading an approach to assembly that used BASIC to translate assembly concepts, though I don’t remember the title of the book. I wonder how well this method of teaching assembly would have suited him had he known about it. Regardless, he wasn't interested in programming before, and his mind was not changed. People without an interest in programming are unlikely to be interested much in the homebrew community in its current form.

4.2 -- Documentation Hard to Find

Five out of the six participants indicated that the documentation was difficult to find. When asked if the documentation was easy to find, four of the six participants answered “no” and one answered “somewhat.” Only one participant directly said the documentation was easy to find, though during the test, I observed him having problems locating some documents. People went through several pages to find the documents and apparently had many problems although all of the participants eventually found those documents. Throughout, I observed frequent usage of the “back” button. Counting the pilot run, everyone missed at least one piece of important document on their first run through the website. These problems are significant, as they alienate newcomers right away. If new users can’t find the information they need to get started, and they become overwhelmed by the quantity of information, how can the community expect to be sustainable? Serious efforts are required by the community to look at the basic design of the wiki in order to gain membership and foster community.

Participants had particular problems finding critical documents in two of the presented test challenges. The first was from the wiki; users had trouble locating the g*etting started* guide intended for beginners. On the wiki's main page, the two main sections are a general reference guide for NES hardware, and a programmer-centric guide. Subjects were unsure at first glance about which section to visit. Two users first looked to the left sidebar to see if the *getting started* guide was located there. This is a typical place where high level information is menued for users, which we know because users tend to skim and scan web-pages for quick pieces of information (Stone, Jarret, Woodrofe, Minocha, 366). If users are unable to find the information they are looking for right away, they are far more likely to give up searching and leave the page, and thus newcomers are alienated. Because the information was not

included in a top-level menu, participants only had effectively 50/50 shot at visiting the right page.

Everyone eventually found the getting started guide, but it took several minutes usually. Research shows that people tend to give up really quickly if they can't find the information they are looking for. Some sources suggest it may be as little as ten to twenty seconds[11]. Much of the documentation resembles this setup; not too cryptic to find, but not as easy as it should be either. Once users located the getting started guide, they were able to locate the remainder of the documentation within this category relatively easily. Having related topics all together is good design and conforms to user expectations. What is not good is the way this information is presented. Much of the documentation exists not on the wiki itself, but as external links to many different sites. I will return to this point in a moment.

A couple users however did have some trouble navigating back to the main page and finding CPU and PPU information from the reference guide, instead opting to dive back into the programming guide. Given the best practices for web design, information as crucial as a *getting started* guide should be somewhere on the front page, and a clearer distinction between "reference" and "programmers" guide might be wise. When it came time to test, Subjects *3* and *4* became lost when navigating back to the getting started guide and ran into the same problem of which page to go to first. Once again, this sort of design only benefits regulars to the NESDEV community and proves the wiki was not designed with new users in mind.

The second major problem was locating a good tutorial for beginners. On the wiki, most of the tutorials are utterly unfinished. Much simple preliminary work has not been acted upon; a tutorial on adding binary numbers says "to be written" in nearly every section. Almost as bad, several important articles on the wiki remain as stubs or a series of links to external sources instead of being a cohesive

11 http://www.nngroup.com/articles/how-long-do-users-stay-on-web-pages/

integrated community experience. Test participant *3* wanted to use the tutorials on the wiki instead of the provided general documentation from the getting started guide, and grew increasingly frustrated from constant "to be written" messages. The wiki is not new, and many of these "to be written" pages were made several years ago, but were never completed. There is no doubt from a communitarian and sustainability standpoint, this is really bad. The state of the wiki suggests a community on NESDEV either negligent of the wiki, or content with it serving their purposes as a general reference guide for experienced programmers. Building a community requires effort to recruit and maintain membership. Perhaps part of the reason . As discussed by Lynda Walsh, "[T]hese clients may have felt as though they were not a part of the wiki community because their participation was not solicited and reciprocated repeatedly—just as many of us feel more comfortable lurking on new forums instead of posting or joining" (Walsh, 26).

Some documentation was only accessible effectively through the forum. *Nerdy Nights,* the tutorial I specifically wanted users to find (since it contained example code I wanted them to use) was locatable from NESDEV via the getting started forum topic in the newbie help center board.

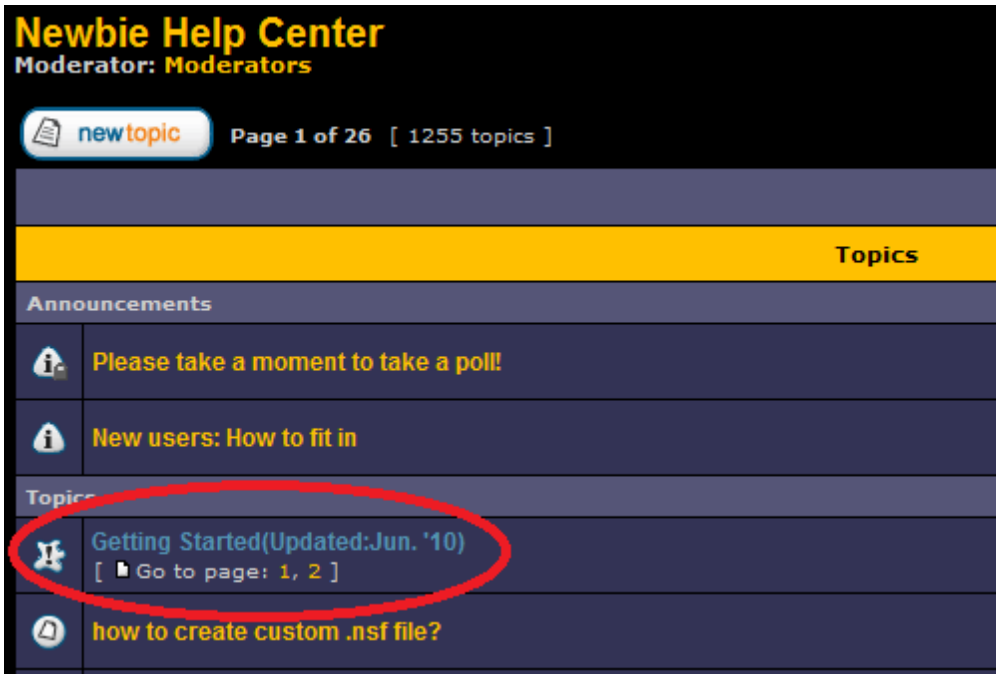


Figure 4.1: The “Getting Started” topic is stickied, meaning it is floated at the top of the page. This is good because it means this important topic will always be stuck to the top of the page and not disappear.

It was buried in the thread underneath several paragraphs of discussion on setting up a development environment, given a vague description that didn't specify it was really for beginners, and was beneath a broken link to another tutorial written by the same user.(again, screen capture to show us how crazy this is) The same section also provided tutorials on 6502 assembly that were highlighted as specifically for beginners. Some of the users navigated to these, but found the documentation highly technical (what do you mean by that? Is this good or bad? Will you get to that later as you indicate? If so, tell your reader the name of the section that you will discuss this finding under.). We will discuss the nature of these documents in a later section.

Part of the problem for NESDEV in particular (though I've seen this issue in some other homebrew sites like AtariAge to a lesser degree) comes from a reliance on off-site links for much of the documentation. This structure discourages community development, instead treating the wiki more like

a glorified instruction manual and less an active community where people can gather to study and learn. The wiki does not need a massive amount of users to improve the content; just a few more people to take active interest and realize improvements need to be made. I myself am considering making edits when this research project is wrapped up, but have so far avoided making edits in order to avoid contaminating data.

In the future, I would like to adapt the test for different homebrew communities in order to compare. I wanted to study NintendoAge and AtariAge especially. NA has significant but unofficial overlap with the NESDEV community, and AtariAge has a slightly older audience and development into BatariBasic, a high level language for Atari 2600. But despite being more polished, they have many similar dis-organizational issues like documentation inside forums, cryptic hyperlinks, and offsite documentation. They have no functioning wiki either, which has its advantages and disadvantages, as discussed by Wei et. al. Wikis may foster community and be good for collaborative writing, but they can also intimidate novices. Learning how to edit a wiki and all the syntax involved on top of learning an entirely new programming language may be too much for some users. When someone is trying their hardest to get up to speed with programming in assembly, what time would they have for a wiki?

4.3 – Documentation Too Technical

Every test participant commented in some form about how technical the documentation was. The 6502 assembly documentation in particular was a point of complaint. For the basic information on the Wiki, people did not feel the documents were good for beginner explanations.

I also feel most users didn't really understand at a fundamental level what they were doing. The

documentation in the getting started guide linking to information on the binary numeral system did a poor job of explaining exactly what it meant for a number to have a different base. I remember personally having the same problem several years earlier and used a search engine to figure out hexadecimal and binary. One user commented she felt like she was merely carrying out operations, not really understanding why what she was doing worked. Another said, "this math is beyond my level of understanding" regarding hexadecimal conversions. He was reminded of "the new math" movement from the 1960s, a notoriously botched attempt to teach computer related concepts to school-children of the time. Although mathematically hexadecimal and binary are not complicated subjects, without understanding the core concept of why a numeral counting system can be in a different base, the concept becomes confusing, abstract, and limited to only a procedural understanding. Even though in the test results might show decent performance purely from correct answers, five out of the six participants asked for help for some degree. Less technical people in particular tended to rely on me to help them through the mathematical portions. The one who did not had prior experience with hexadecimal and binary/bitwise functions. I observed emotions ranging from nervousness to frustration frequently while users completed this part of the test.

With the exception of one subject, everyone I tested gave up at the assembly portion of the test. The most common conventional document users chose was *Assembly in One Step* by RT[12]. This particular document is similar to the 6502's official manual. Users also generally accessed assembly related documents from the getting started guide on the wiki[13]. Most frequently users would look at a list of instructions for the 6502 found from the Wiki's page "6502 instructions" and try to figure out

12 http://nesdev.com/6502guid.txt

13 http://www.obelisk.demon.co.uk/6502/reference.html

what the instructions meant from this list. Subjects would read through the assembly related documentation and comment they did not understand it. Later when asked to program in assembly, they would pull up the 6502 instruction set and got intimidated and/or confused. Nobody tried going back to look for more documentation such as the official manual.

Several methods for teaching hardware and assembly more easily have been proposed within the academic community. A common practice for teaching assembly is to relate ideas and programming concepts to higher level languages. . Another approach is to gradually ease students into learning hardware concepts. In *A New Approach for Teaching Microcontroller Courses,* the authors suggest a step-by-step process for teaching micro-controllers. Students would theoretically be given small and digestible, but real world tasks for programming micro-controllers. The approach described here reminds me of how *Introduction to Computing Systems* was structured. The NES homebrew community should take cues from these models. Currently, the documentation instead overwhelms users by throwing them into a mass of different documents to select from. General reference material seems designed for people already familiar with a system. We can see evidence of this in the study itself from the two programmers tested. 1 was fairly easily able to figure out programming in this particular variant of assembly, though even he made some initial mistakes. Participant 5 was interested in learning assembly in order to program the NES, and gave up because the listed reference material style overwhelmed him. Given the trend of people giving up on assembly on the test, it seems fair to state the highly technical nature of the reference material fails to take a humanistic approach.

Outside Source Reliance

Wei, Carolyn, et al. "Wikis for supporting distributed collaborative writing."*Proceedings of the Society for Technical Communication 52nd Annual Conference*. 2005.

A heavy reliance on outside sources is present within the wiki. Although using outside web pages makes sense for , for the NESDEV wiki it seems more an excuse to leave pages as unfinished stubs and call it a day. For instance, the page “6502 Instructions” (http://wiki.nesdev.com/w/index.php/6502_instructions) links to several outside resources instead of discussing the information on the 6502 instruction set directly. These documents are certainly useful to experienced assembly programmers like participant 1, but don't take advantage of the full capabilities of a wiki, and are bad for beginners. Users were completely lost when browsing the documents on the 6502 instruction page. Some of them also only found 6502 information on the wiki’s CPU page. Linking to outside sources does little to foster community on the wiki. Wei et al and many other academic articles discuss the collaborative learning experience of a wiki being its greatest strength. But this wiki takes little advantage of the unique functionality of wikis and instead works like a glorified web page. It seems clear to me writing a polished wiki is not a high priority for the NESDEV community.

Undoubtedly, it seems NESDEV's primary mode of communication is the forum and not the wiki. Compared to the beginning when the wiki was first started, the relationship is stronger, but the overall use of the wiki is limited to a simple more organized replacement of NESDEV’s main webpage. Most discussion relating to the wiki seems to be as a reference manual of sorts. Several examples of this can be seen by observing the communications within the newbie help center and general programming board. Users will ask a question, someone will respond with the necessary technical information, and then post a link to the wiki.

Users sporadically update and add to the wiki. It seems the NESDEV community is overall

content with the wiki in its current form. It took several years for it to get off tie ground, but now that all the essential and basic technical information is present on the wiki, most users seem content to use it as a reference guide in order to answer general questions and refresh their memory regarding NES hardware specifics (ie. register locations, menu addresses, etc.). Information continues to be added to the wiki, but slowly. One of the more current posts regarding the wiki from user Tepples detailed June 25 2014 simply says, "This topic inspired me to *finally* create the wiki page about tile compression that I promised four years ago."

4.5 – Why the Documentation is so technical

The short answer for why the documentation on NESDEV often does not exhibit best practices, is full of technical jargon, and demonstrates an overall unprofessional feel is because NESDEV is not a professional community. This should be expected; after all homebrew communities are hobbyist communities, and are especially programmer centric. To put the sloppy and highly technical presentation of these communities in perspective, first think of how many programmers in the professional world don't adhere to good writing practices and documentation. Now imagine an entire community of them with absolutely no structure or discipline. While this might give a unique charm to these communities in the minds of many, it also makes the communication less usable for curious learners and perhaps encourages elitism in some circles.

One particular trend I have noticed with documentation in the NES homebrew community and other game modding communities is a lack of relatable analogies a beginner could relate to. Communication alienates beginners because it often assumes prior knowledge of hardware and programming concepts . Research such as Candace Soderston's and Carol German's *Toward bridging*

the gap between theory and practice: Analogy and person in technical communication has since the 1980s revealed analogy plays a crucial role in communicating complex technical concepts to students and novices. The documentation test participants found from NESDEV such as *Assembly in One Step* and *Nerdy Nights* although intended for beginners exhibit a distinct lack of relatable analogies for beginners. It is no wonder then on *Nerdy Nights*, one user commented on week 1, "Ahhh my brain hurts."

For a typical example of how the wiki articles are written, this page detailing the NES's graphics card, known commonly as the PPU (Picture Processing Unit) makes several assumptions: (http://wiki.nesdev.com/w/index.php/PPU). Users assumed to know the definitions of terms such as composite video, address space, Object Attribute Memory, and the difference between static and dynamic memory. My test participants took interest in reading these actual pages. Participant 6 commented on both the CPU and PPU pages that he didn't understand the document. It then proceeds to list more wiki articles detailing information on the processors registers, pattern tables, pallete, hardware behaviors, etc. Thus the wiki's main audience is thus an audience familiar with NES programming already, or at least hardware level programming.

For users experienced with low level programming, the wiki is a useful guide. It uses an exhaustive style that catalogs the major hardware related topics necessary for programming. This is notable in how participants 1 and 5 reacted to the test. For beginners to this sort of programming like the remainder of my test participants, the wiki is less useful and possesses many of the same problems as the older documentation being too difficult to read. The NESDEV wiki does a good job of organizing general information on NES hardware, but is not written in way accessible for newcomers. Information remains too technical for beginners, being full of charts full of endless memory addresses.

Although the NESDEV wiki contains a “getting started” guide within the “Programming Reference,” (http://wiki.nesdev.com/w/index.php/Before_the_basics) the quality is dubious for newcomers. It's first (and only complete) paragraph reads as:

> *A digital computer, such as a PC or NES, is filled with millions of tiny switches that can be turned on or off. Some switches store information for short time; these are called memory. Other switches calculate things based on the output of other switches; these are called logic. Each switch can be on or off, representing true or false, or the numbers 1 or 0. Multiple switches grouped together can represent larger values. A CPU reads machine code from a memory and treats it as instructions to perform arithmetic and logic operations on other values in memory or to perform input and output. Some of the input and output involve user interaction: the user gives commands to the computer, and the computer displays the result.*

Barring the information overload in this paragraph itself, the page proceeds to list about a dozen different topics of wikipedia pages the writer feels is important. These topics range from bits, binary number systems, memory, CPUs, raster graphics, command line interfaces, and more. Each of these topics has several sub-categories of additional information to read. It seems inevitable a new user could become overwhelmed. Integrating information needed for newcomers into the actual wiki itself and not outside links should be on the top of the priority list for the NESDEV community if they intend to make this information accessible for beginners. My test subjects did not take liking to having to scroll through pages and pages of technical lists of assembly instructions and disparate Wikipedia articles. If enough users decide to edit and complete the wiki by finishing tutorials and tying sections together into more concrete, small chunks, it will both attract newcomers and strengthen the already existing community.

Conclusion

My concern from the beginning has been how homebrewing communities can sustain themselves and grow. The hope of sustained growth as far as NES homebrew is concerned comes from NintendoAge. While NESDEV based on the trends data has been a niche site for years, NintendoAge demonstrates potential for growth and seems to be more interested in attracting new members. The comparison between these two sites and their overlap is very interesting. Carrying out an adapted usability study for NintendoAge in the future would be very useful.

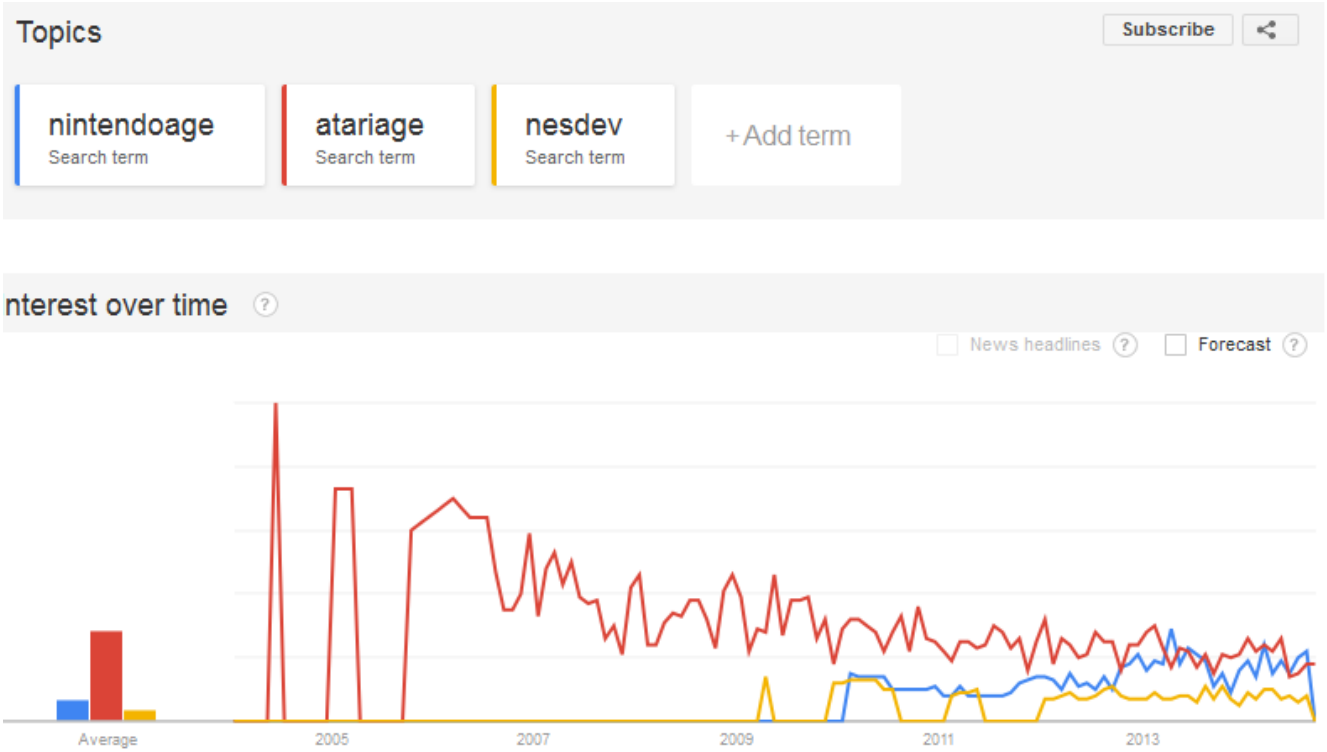


Figure 5.1: Google Trend data on NintendoAge vs NESDEV vs AtariAge

The forum itself shows a massive disparity in max membership and now (Most users ever online was **321** on Fri Jan 06, 2012 10:56 pm; As of 9-9-2014, 20 to 30 seems to be the average, and

that's including bots. Although NintendoAge is growing, Google trends shows an overall decline in general homebrew interest. As we can see from the chart, it has passed Past the “peak” at the top of a bell curve already. NES homebrew is a blip compared to other modern homebrew, but even that's in decline. Refer back to the charts from Google Trends demoed in the background sections. Even Nintendo itself as a search time has declined.

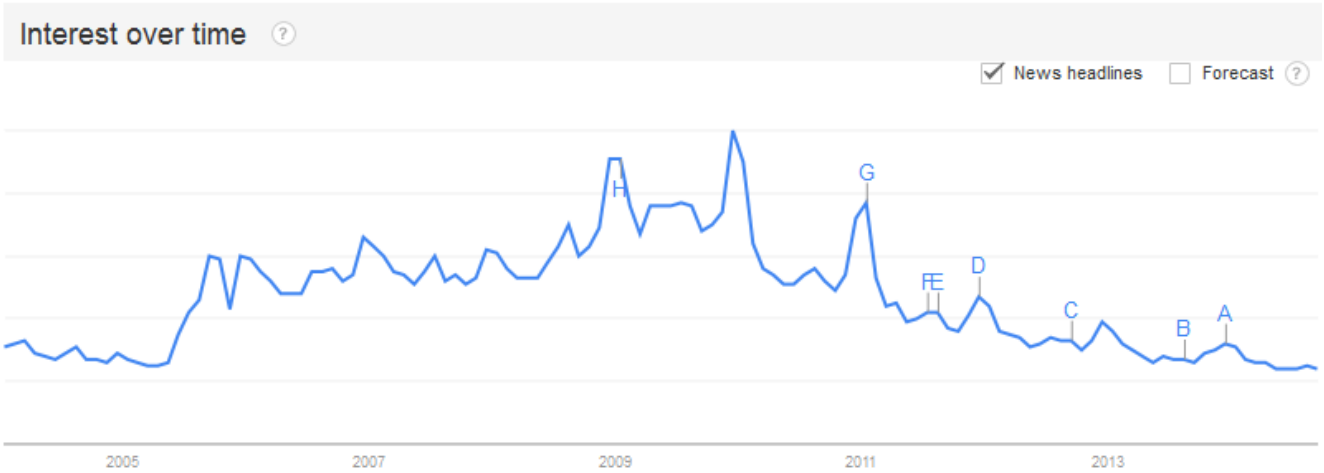


Figure 5.2: Google Trends Data for “Homebrew”

I believe the piracy crackdown as of late has a strong influence on these declines. NESDEV has enough mirrors that all the core documentation won't disappear. Still, if the community itself died, it'd be a blow to other parts of the homebrewing community's morale and perhaps create a negative atmosphere for years to come.

In NESDEV's case, I see two main choices present. These are certainly not the only options, but they are the most apparent to me. The community can improve documentation and attract newcomers (write a list of recommendations). Or they can focus themselves as a small group of insiders and close friends like Nintendo City was. There is certainly nothing inherently wrong with the later option, but I know first hand how being too “insider” can backfire on a community when presented with unexpected

and sudden changes.

I hope the homebrewing community finds my study useful for improving the quality of their documentation. Even for moderately skilled to expert users, several problems were encountered. In the future, I intend to develop a textbook and coursework from this initial research. It will be the user guide I would have wanted as my 12 year old self; hopelessly lost and in desperate need of guidance for navigating the gauntlet of programming for a reverse engineered system.

Attracting newcomers will be key in the future to keeping homebrew communities going. A new generation of children who are parents of the geeks of the 1980s is in the process of growing up. The time is ripe to think about keeping the spirit of retro gaming alive, and what better celebration is there of retro gaming than to develop games for the real hardware? The groundwork for a retro Renaissance has been laid down through the hard work of incredibly dedicated people. It is now up to us to continue the work, and popularize the magic of 8-bit creativity.

References

Augar, Naomi, Ruth Raitman, and Wanlei Zhou. "Teaching and learning online with wikis." *Beyond the comfort zone: proceedings of the 21st ASCILITE Conference, Perth, 5-8 December*. ASCILITE, 2004.

Butler, Brian, et al. "Community effort in online groups: Who does the work and why?" *Leadership at a distance: Research in technologically supported work* (2002): 171-194.

Butler, Brian S. "Membership size, communication activity, and sustainability: A resource-based model of online social structures." *Information systems research* 12.4 (2001): 346-362.

Camper, Brett Bennett. *Homebrew and the social construction of gaming: community, creativity, and legal context of amateur Game Boy Advance development*. Diss. Massachusetts Institute of Technology, 2005.

Camper, Brett. "Reveling in Restrictions: Technical Mastery and Game Boy Advance Homebrew Software Development." (2005).

Claypool, Kajal, and Mark Claypool. "Teaching software engineering through game design." *ACM SIGCSE Bulletin*. Vol. 37. No. 3. ACM, 2005.

Crookes, D. "Teaching assembly-language programming: a high-level approach." *Software & Microsystems* 2.2 (1983): 40-43.

Duffy, Peter D., and Axel Bruns. "The use of blogs, wikis and RSS in education: A conversation of possibilities." (2006): 31-38.

Etzioni, Amitai. "Positive aspects of community and the dangers of fragmentation." *Development and Change* 27.2 (1996): 301-314.

Fletcher, Holly. *Shattered Sky: An Exploration in Rising Drama*. Diss. WORCESTER POLYTECHNIC INSTITUTE, 2012.

Hamrita, T. K., and R. W. McClendon. "A new approach for teaching microcontroller courses." *International Journal of Engineering Education* 13.4 (1997): 269-274.

Hunter, Scott B. "Teaching assembly language without using (as much) assembly language." *Journal of Computing Sciences in Colleges* 20.5 (2005): 68-78.

Jordan, Will. "From rule-breaking to ROM-hacking: theorizing the computer game-as-commodity." *Situated Play DiGRA 2007 Conference*. 2007.

Katz, Steven B. "The ethic of expediency: Classical rhetoric, technology, and the Holocaust." *College English* (1992): 255-275.

Kelley, Nicole. "Sentence Structure of Technical Writing." Program in Writing and Humanistic Studies. MIT, Cambridge. Web. 23 June 2014. <http://web.mit.edu/me-ugoffice/communication/technical-writing.pdf>.

McIver, Linda. "Evaluating languages and environments for novice programmers." *Fourteenth Annual Workshop of the Psychology of Programming Interest Group (PPIG 2002), Brunel University, Middlesex, UK*. 2002.

N.p., n.d. Web. 11 Sept. 2014. <http://cs.kutztown.edu/pdfs/Documentation_Levels.pdf>.

Parker, Brian "BunnyBoy" "NA » Nerdy Nights Week 1." *NintendoAge RSS*. NintendoAge, 04 Oct. 2007. Web. 26 June 2014. <http://www.nintendoage.com/forum/messageview.cfm?catid=22&threadid=4147>.

P. Kraut, M. Patterson, V. Lundmark, S. Kiesler, T. Mukopadhyay, W. Scherlis. *Internet paradox: a social technology that reduces social involvement and psychological well-being?*

Slobu. "NA » AtariAge Yanks Princess Rescue after C&D?" *NintendoAge RSS*. NintendoAge, 11 Sept. 2013. Web. 26 June 2014.

Smith, Summer. "What is" Good" Technical Communication? A Comparison of the Standards of Writing and Engineering Instructors." *Technical Communication Quarterly* 12.1 (2003): 7-24.

Soderston, Candace, and Carol German. "Toward bridging the gap between theory and practice: Analogy and person in technical communication." *Journal of Business and Technical Communication* 2.1 (1988): 78-102.

Walsh, Lynda. "Constructive interference: Wikis and service learning in the technical communication classroom." *Technical Communication Quarterly* 19.2 (2010): 184-211.

Wei, Carolyn, et al. "Wikis for supporting distributed collaborative writing."*Proceedings of the Society for Technical Communication 52nd Annual Conference*. 2005.

Wellman, Barry, and Milena Gulia. "Virtual communities as communities."*Communities in cyberspace* (1999): 167-194.

Bibliography

Barnum, Carol M. *Usability Testing Essentials: Ready, Set-- Test*. Burlington, MA: Morgan Kaufmann, 2011. Print.

Dye, Thomas R.. *Who's running America?: Institutional leadership in the United States*. Englewood Cliffs, N.J.: Prentice-Hall, 1976. Print.

Francis, Mark, and Lisa Cashdan. *Community open spaces: greening neighborhoods through community action and land conservation*. Washington, D.C.: Island Press, 1984. Print.

Gilbert, Neil, and Harry Specht. *Dynamics of community planning*. Cambridge, Mass: Ballinger Pub. Co., 1977. Print.

Lasica, J. D.. *Darknet: remixing the future of movies, music, and television*. Hoboken, NJ: J. Wiley & Sons, 2005. Print.

Lehmann, Steffen, and Robert Crocker. *Designing for Zero Waste*. London and New York: Routledge, 2012. Print.

Mantell, Michael A., and Stephen F. Harper. *Creating successful communities: a guidebook to growth management strategies*. Washington, D.C.: Island Press, 1990. Print.

McGonigal, Jane. *Reality is Broken: Why Games Make Us Better and How They Can Change the World*. New York: Penguin Press, 2011. Print.

Montfort, Nick, and Ian Bogost. *Racing the beam: The Atari video computer system*. Cambridge, MA: Mit Press, 2009.

Rogers, Everett M. *Diffusion of Innovations*. New York: Free Press, 2003. Print.

Ryn, Sim, and Peter Calthorpe. *Sustainable communities: a new design synthesis for cities, suburbs, and towns*. San Francisco: Sierra Club Books, 1986. Print.

Schäfer, Mirko Tobias. *Bastard culture!: how user participation transforms cultural production*. Vol. 6. Amsterdam University Press, 2011.

Stone, Debbie, et al. *User interface design and evaluation*. Morgan Kaufmann, 2005.

Tapscott, Don. *The digital economy: promise and peril in the age of networked intelligence*. New York: McGraw-Hill, 1995. Print.

Witheford, Nick. *Cyber-Marx: cycles and circuits of struggle in high-technology capitalism*. Urbana: University of Illinois Press, 1999. Print.

Appendix A: The Sustainability Game

The Sustainability Game Treatment

By William Poirier

Elevator Pitch: The sustainability game is a large-scale real world Alternate Reality game that can span an entire community, ranging from a local neighborhood to a whole municipality should people opt-in. The premise is simple: the community has two weeks to prepare themselves for an impending long-term blackout due to a catastrophe. After that, the power in the community goes out, and the players must survive as long as possible.

The Rules

The magic circle takes place in real space. Depending how the players and game managers want to set the game up, it could take on a variety of forms. It could take place in a regular, every day community.

The game is divided into two parts. The first being the preparation phase, and the second being the blackout phase.

1. Preparation

2. Blackout

The Community Can:

- Use any resources within the magic circle to aid them by any means necessary.
- Create alliances with other players, either inclusive or exclusive; or can go lone-wolf
- Share or steal resources from other players
- Commit *simulated* violence (ie. with paintball/nerf guns, foam swords, etc.) on other players and non-playable characters

The Community Cannot:

- Receive outside help or support. In our scenario, we assume all major markets have been exhausted of supplies. Our players must work with what they have.

- Step outside the magic circle without forfeiting their role in the game. Walking away will be treated as "death" within the game.

- Exert any *real* violence on one another or any other participants.

Brainstorming Ideas

Should it be okay for player's to shun or exile other players collectively?

I feel it's important for this game's purpose to have a greater impact; real world communities must play this game. That said, we also risk property damage with the current rules being as loose as they are. But I also don't want to tighten the rules too much because it would defeat the purpose of the experimental side of this project: how communities will react in collapse scenarios. If the players are all properly insured by the game managers, this might be a non-issue. But if not, should the rules be tightened to avoid serious property damage? Perhaps players could agree ahead of time designated storage areas for items they don't wish to be damaged that are off-limits in the duration of the game.

We could have a variety of outside "attackers" or insurgents infiltrate the community during either the of the phases. Characters could range from looters, zombies, authorities, , etc. These types of characters could be considered a kind of NPC.

Speaking of NPCs, we could have more developed characters pop in too. Some of them could be benevolent, or fiendishly deceptive.

Appendix B: Pre-Test

PRE-TEST Participant_____

What is your age? _____

What is your gender? _____

Do you play videogames? [YES] [NO]

If yes to the previous Questions

How Frequently?

a. Less than once a month
b. Less than once a week
c. Two to five times a week
d. Once per day
e. Several hours per day

What are some examples of your favorite games?

What are your favorite game genres?

What generation of video games is your favorite? (circle one)

1. 1st Generation (Early 1970s; *PONG, Magnavox Odyssey*)
2. 2nd Generation (Late 1970s to early 1980s; *Atari 2600, Intellivision*)
3. 3rd Generation (Mid 1980s; *Nintendo NES, Sega Master System*)
4. 4th Generation (Early 1990s; *Nintendo SNES, Sega Genesis*)
5. 5th Generation (Mid to late 1990s; *N64, Playstation, Sega Saturn*)
6. 6th Generation (Early 2000s; *Gamecube, PS2, Xbox*)
7. Previous Generation (Mid-2000s; *Wii, PS3, Xbox 360*)
8. Current Generation (Now; *Wii U, PS4, Xbox One*)

What Game Systems do you own?

Do you consider yourself a "gamer"? [YES] [NO]

What is your occupation/major/etc? _____

Have you ever used a programming language before? [YES] [NO]

At what level would you rate your programming experience? (circle one)

a. Professional
b. Student (studying to be professional)
c. Hobbyist
d. Beginner
e. None

What High Level Programming languages are you proficient in? (ie. C++, Java, BASIC, etc.)

Have you ever programmed in Assembly Language before? [YES] [NO]

Do you intend to learn a new programming language in the future? [YES] [NO]

If so, what language or languages?

Do you associate the term "homebrewing" with alchohol, or with programming?

a. Alchohol
b. Programming

Do you have any experience homebrewing (programming-wise)? [YES] [NO]

If yes, what systems?

If not, do you plan on homebrewing at some point?
[YES] [NO]

Do you have any experience in game modification (modding)? [YES] [NO]

If yes, what games?

If not, do you plan on engaging in game modification?

Emotionally, how would you react to the following terms? (ie. anxious, happy, indifferent, etc.)

Programming

Coding

Assembly

Machine Code

Nintendo

Gaming

Appendix C: Final Iteration of Usability Test

USABILITY TEST

PART 1: Complete the following tasks from the root nesdev.com without using a search engine

Wiki

For each non-segmented task, please return to the main wiki page

[ ] Find the WIKI
[ ] Find the Wiki's getting started guide
    [ ] Find out what a "bit" is
    [ ] Find out how to add binary numbers
    [ ] Find out how to perform Boolean Algebra.
    [ ] Find out how to convert numbers between base 10, 2, and 16
[ ] Find reference material for programing the MOS 6502 Processer
[ ] Find information on the NES's CPU (central processer)
[ ] Find information on the NES's PPU (picture processer)

Forum

[ ] Find the FORUM
[ ] Find the new users board
[ ] Find the getting started guide
[ ] Find a tutorial intended for beginners

From the tutorial and/or getting started section...
    [ ] Find information on programming in 6502 Assembly
    [ ] Find example code for making a character sprite move
    [ ] Find information on how to change the NES background color
    [ ] Find information on how to change the NES sprite pallete

PART 2: Use the documentation and information you've collected to answer the following:

Beginner Level
1) Add 0010 to 0011 in base 2

2) Do a left bit shift on 0101

3) Evalute:
    1011 AND 0111

    1100 OR 1010

NOT 1000

4) Convert these hex values into base 10 and base 2
$09

$0B

$1F

Write the following instructions in 6502 ASSEMBLY.
Check your code with the provided debugger.

1) Add the hex values $2 and $10
    -Now store the result in memory locatiom $00
    -Load that memory location into the Y register
2) Load the X register with the hex value $2F and transfer it to the accumulator
3) Push $01 onto the 6502's stack
4) Unconditionally jump to a different label in [Program shown below]
5) Initialize the X and Y registers to $00 and $10 respectivly, then
6) Use labels and 6502 compare instructions to create a loop that terminates after 7 iterations

Use Example Code you've found (or are provided) to modify the asm file
1) Compile the example code to make a character move
2) Change the background color
3) Change the color of the sprite
4) Change the input parameters so that the character's movement is inverted
5) In English, what the programming instructions in this example are doing.

Appendix D: Post-Test

POST TEST

1) Did you feel overwealmed by the amount of information present on the website or the documentation itself?

2) Did you find the documentation...

a) well organized?

b) easy to understand?

c) easy to find?

3) Were there any parts of the homebrew website you found confusing? If so, why?

4) Do you feel encouraged after taking this test to continue learning about...?

a) Homebrewing

b) Game Modification

c) Game Programming

5) Did you feel a need to rely on the test practitioner to help you? Why or why not?

6) When you were tasked to compile and modify the asm file, did you understand why your changes were effecting the program?

Appendix E: Spread Sheet Data

Participant	Age	Gender	Occupation
#1	21	Male	release engineer

PRETEST			
Question Topic	*Response1*	*Response2*	*Response 3…*
Plays Games			
YES/NO	yes		
Frequency	less than once per week		
Favorite Games			
Favorite Genres	Action, fighting	RTS	MMO
Favorite Generation	5th generation		
Systems Owned	Wii	PS2	PC
Identifies as Gamer	yes		
Programming			
Coded before	yes		
Experience Level	student		
High Level Languages	C++, C, LUA, Javascript	HTML, CSS, PHP	Python, VB, Unix Shell
Assembly Experience	yes		
Languages to Learn	Perl, D, C#	Objective C	
Modding			
Homebrew Association	Alcohol		
Homebrew Experience	none		
Systems Coded For			
Intends to Homebrew	no		
Modding Experience	no		
Games Modded			
Intends to Mod	"technically no experience. Have looked into and briefly Modified config files for minecraft, wolfenstein, and mapplestory"		
Affect			
Programming	happy		
Coding	indifferent		
Assembly	indifferent		
Machine Code	indifferent		
Nintendo	happy		
Gaming	happy		

TEST

Page Type	Task	Found	Where
WIKI	Find the Wiki	x	
	Getting Started Guide	x	
	Bits	x	
	Adding in binary	x	
	Boolean Algebra	x	
	Base Conversions	x	
	MOS 6502	x	
	NES CPU	x	
	NES PPU	x	
Forum	Find the Forum	x	
	Newbie Board	x	
	Getting Started Post	x	
	Beginner's Tutorial	x	
Tutorial	6502 programming	x	
	Sprite movement code	x	
	Change BG color	x	
	Change sprite palette	x	

Question	Answer	Correct
Bitwise		
Add 0010b+0011b	0101b	yes
Left Bit Shift 0101b	1010b	yes
1011b AND 0111b	0011b	yes
1100b OR 1010b	1110b	yes
NOT 1000b	0111b	yes
Hex Conversion		
0x09	9, 1001b	yes
0x0B	11, 1011b	yes
0x1F	31, 11111b	yes

Asked for help	no
Gave Up At Assembly	no

Question 1	Question 2	Question 3
ADD: LDX #2 ; Get the val 2	LDX #$2F ; GET avlue 43d i	LDX #0 ; LOAD 0 into x
STX $5000 ; store it	TXA ; transfer to accum	LDY #10 ; LOAD 10 into y
LDA $10 ; get the val 16d	PHA ; push accum to stack	LDA #6
ADC $5000 ; add prev vals	JMP ADD ; GO TO ADD	LOOP: INX ; increment x
RES: STA $0 ; store result		STX $5000
LDY $0 ; load result into y		CMP $5000 ;
		BNE LOOP ;

NOTE: initially, tried to push numbers onto the stack to add;
He thought the processor was stack based based on the instruction docs

POST TEST

Question	Response	Note
Overwhelmed	no	"However, I think a newcomer would"
Documentation is..		
Well organized	no	
easy to understand	yes	
easy to find	no	
		Found the design confusing
Found site confusing	somewhat	But otherwise everything is standard
Felt encouraged to...		
homebrew	no	
game mod	no	
program	no	
Relied on practitioner	no	

Participant	Age	Gender	Occupation
#2	60	Female	Teacher

PRETEST			
Question Topic	*Response1*	*Response2*	*Response 3…*
Plays Games			
YES/NO	Yes		
Frequency	< once per week		
Favorite Games	Skeeball	2048	
Favorite Genres	Casual Games		
Favorite Generation	8th generation		
Systems Owned	Iphone 5S		
Identifies as Gamer	No		
Programming			
Coded before	No		
Experience Level	none		
High Level Languages	none		
Assembly Experience	none		
Intends to learn	Yes		
Languages to Learn	unsure		
Modding			
Homebrew Association	Alchohol		
Homebrew Experience	none		
Systems Coded For	NA		
Intends to Homebrew	No		
Modding Experience	none		
Games Modded	NA		
Intends to Mod	No		
Affect			
Programming	Intimidating		
Coding	Scary		
Assembly	What's this?		
Machine Code	Puzzling		
Nintendo	Happy and fun		
Gaming	Enjoyable		

TEST

Page Type	Task	Found	Where
WIKI	Find the Wiki	X	
	Getting Started Guide	X	
	Bits	X	
	Adding in binary	X	
	Boolean Algebra	X	
	Base Conversions	X	
	MOS 6502	X	
	NES CPU		
	NES PPU		
Forum	Find the Forum	X	
	Newbie Board	X	
	Getting Started Post	X	
	Beginner's Tutorial	X	
Tutorial	6502 programming	X	
	Sprite movement code		
	Change BG color		
	Change sprite palette		

Question	Answer	Correct	
Bitwise			
Add 0010b+0011b	0101b	yes	
Left Bit Shift 0101b	01010b	yes	
1011b AND 0111b	FALSE	no	note: I had to explai
1100b OR 1010b	TRUE	no	Once I personally e:
NOT 1000b	FALSE	no	she wrote the corre
Hex Conversion			
0x09	1001b	yes	
0x0B	~~0000b~~, 11, 1011	no initially	
0x1F	115		

Asked for help	YES
Gave Up At Assembly	YES

POST TEST

Question	Response	
Overwhelmed	YES	
Documentation is..		
Well organized	Somewhat	
easy to understand	Somewhat	
easy to find	No	
Found site confusing	For hex conversions	
Felt encouraged to...		
homebrew	No	
game mod	No	
program	Somewhat	"If I tried again, I feel like I might be able to
Relied on practitioner	YES	"He encouraged me when I was frustrated a

Participant	Age	Gender	Occupation
#3	21	Female	psychologist

PRETEST			
Question Topic	*Response1*	*Response2*	*Response 3…*
Plays Games			
YES/NO	yes		
Frequency	less than once per month		
Favorite Games	The Sims	Super Mario Bros	Wii Sports
Favorite Genres			
Favorite Generation	4th generation		
Systems Owned	None		
Identifies as Gamer	No		
Programming			
Coded before	No		
Experience Level	None		
High Level Languages			
Intends to learn More	No		
Assembly Experience			
Languages to Learn			
Modding			
Homebrew Association	Alcohol		
Homebrew Experience	None		
Systems Coded For			
Intends to Homebrew	No		
Modding Experience	None		
Games Modded			
Intends to Mod	No		
Affect			
Programming	indifferent		
Coding	indifferent		
Assembly	indifferent		
Machine Code	indifferent		
Nintendo	gaming		
Gaming	gaming		

TEST

Page Type	Task	Found	Where
WIKI	Find the Wiki	x	
	Getting Started Guide	x	
	Bits	x	
	Adding in binary	x	
	Boolean Algebra	x	
	Base Conversions	x	
	MOS 6502		
	NES CPU	x	
	NES PPU	x	
Forum	Find the Forum	x	
	Newbie Board	x	
	Getting Started Post	x	
	Beginner's Tutorial	x	
Tutorial	6502 programming	x	
	Sprite movement code	x	
	Change BG color	x	
	Change sprite palette	x	

Question	Answer	Correct
Bitwise		
Add 0010b+0011b	0001b	no
Left Bit Shift 0101b		no
1011b AND 0111b		no
1100b OR 1010b		no
NOT 1000b	GRRRRR	no
Hex Conversion		
0x09	1001b	did not convert to base 10
0x0B	1011b	did not convert to base 10
0x1F	0001_1111b	did not convert to base 10

Asked for help	yes
Gave Up At Assembly	yes

POST TEST

Question	Response	Note
Overwhelmed	YES!!!	
Documentation is..		
Well organized	yes	“eh, for the most part”
easy to understand	No	“not really”
easy to find	somewhat	
Found site confusing	yes	Found second part of the test more difficult
Felt encouraged to...		
homebrew	No	
game mod	No	
program	No	
Relied on practitioner	yes	For the second part of the test

Participant	Age	Gender	Occupation
#4	20	M	Game Dev (art)

PRETEST

Question Topic	*Response1*	*Response2*	*Response 3...*
Plays Games			
YES/NO	yes		
Frequency	several hours per day		
Favorite Games	Boarderlands 2	Batman Arkham Asylum	Dungeon Defenders
	Minecraft	Pikmin	Pokemon
Favorite Genres	Platformers	Mild RPG	Stealth
	Racing	Some FPS	Turn Based Strategy
Favorite Generation	6th generation		
Systems Owned	All Nintendo Systems	Xbox 360	
Identifies as Gamer	yes		
Programming			
Coded before	yes		
Experience Level	beginner		
High Level Languages	Javascript	C++	
Intends to learn more	no		
Languages to Learn	none		
Assembly Experience	none		
Modding			
Homebrew Association	Alcohol		
Homebrew Experience	none		
Systems Coded For			
Intends to Homebrew	no		
Modding Experience	yes		
Games Modded	Never-winter Nights 2	Left4Dead 2	
Intends to Mod			
Affect			
Programming	Fear		
Coding	Fear		
Assembly	indifferent	The participant did not know what these were	
Machine Code	indifferent	If he did, he'd probably write "extreme fear"	
Nintendo	nostalgic		
Gaming	happy		

TEST

Page Type	Task	Found	Where
WIKI	Find the Wiki	x	
	Getting Started Guide	x	
	Bits	x	
	Adding in binary	x	
	Boolean Algebra	x	
	Base Conversions	x	
	MOS 6502	x	
	NES CPU	x	getting started guide
	NES PPU		Wrote "=(" on page
Forum	Find the Forum	x	
	Newbie Board	x	
	Getting Started Post	x	
	Beginner's Tutorial	x	
Tutorial	6502 programming	x	
	Sprite movement code	x	
	Change BG color	x	
	Change sprite palette	x	

Question	Answer	Correct
Bitwise		
Add 0010b+0011b	0101b	yes
Left Bit Shift 0101b	1010b	yes
1011b AND 0111b	0011b	yes
1100b OR 1010b	1110b	yes
NOT 1000b	FALSE	no
Hex Conversion		
0x09		
0x0B		
0x1F		

Asked for help	yes	
Gave up at Assembly	yes	(actually, bolted at the hexadecimal section)

POST TEST

Question	Response	Note
Overwhelmed		
Documentation is..		
Well organized	no	
easy to understand	no	
easy to find	yes	wrote "scary"
Found site confusing	yes	"I have no math know-how"
Felt encouraged to...		
homebrew	no	
game mod	no	
program	no	
Relied on practitioner	for math questions	

Participant	Age	Gender	Occupation
#5	20	Male	IMGD Student

PRETEST

Question Topic	*Response1*	*Response2*	*Response 3...*
Plays Games			
YES/NO	YES		
Frequency	Several hours per day		
Favorite Games	Bayonetta	Dark Souls	Metal Gear Solid
Favorite Genres	Action RPG	Spectacle Fighter	
Favorite Generation	6th Generation		
Systems Owned	PS1,PS2,PS3,PSP	GCN,Wii,WiiU,GBA,DS	Xbox,Xbox360,PC
Identifies as Gamer	YES		
Programming			
Coded before	YES		
Experience Level	Student		
High Level Languages	C++, C#	Ruby	Java
Assembly Experience	none		
Intends to learn more	YES		
Languages to Learn	many		
Modding			
Homebrew Association	programming		
Homebrew Experience	none		
Systems Coded For			
Intends to Homebrew	maybe		
Modding Experience	NO		
Games Modded			
Intends to Mod	YES		
Affect			
Programming	indifferent		
Coding	indifferent		
Assembly	anxious		
Machine Code	anxious		
Nintendo	happy		
Gaming	happy		

TEST

Page Type	Task	Found	Where
WIKI	Find the Wiki	x	
	Getting Started Guide	x	
	Bits	x	
	Adding in binary		
	Boolean Algebra	x	
	Base Conversions		
	MOS 6502	x	
	NES CPU	x	
	NES PPU	x	
Forum	Find the Forum	x	
	Newbie Board	x	
	Getting Started Post	x	
	Beginner's Tutorial	x	
Tutorial	6502 programming	x	Assembly in One Step
	Sprite movement code		
	Change BG color	x	
	Change sprite palette	x	

Question	Answer	Correct
Bitwise		
Add 0010b+0011b	0101b	yes
Left Bit Shift 0101b	1010b	yes
1011b AND 0111b	0011b	yes
1100b OR 1010b	1110b	yes
NOT 1000b	0111b	yes
Hex Conversion		
0x09	0001b,9	yes
0x0B	1011b,11	yes
0x1F	0001111b,31	yes

Asked for help	NO
Gave Up At Assembly	YES

POST TEST

Question	Response
Overwhelmed	NO
Documentation is..	
Well organized	YES
easy to understand	YES
easy to find	NO
Found site confusing	Sometimes
Felt encouraged to...	
homebrew	YES
game mod	YES
program	YES
Relied on practitioner	Sometimes

Participant	Age	Gender	Occupation
#6	61	M	Dentist

PRETEST			
Question Topic	*Response1*	*Response2*	*Response 3...*
Plays Games			
YES/NO	yes		
Frequency	Less than once a month		
Favorite Games	Galaga	Space Invaders	Centipede
Favorite Genres	Arcade	Action	
Favorite Generation	2nd Generation		
Systems Owned	none		
Identifies as Gamer	no		
Programming			
Coded before	yes		
Experience Level	beginner		
High Level Languages	BASIC		
Assembly Experience	none		
Intends to learn more	no		
Languages to Learn	none		
Modding			

TEST

Page Type	Task	Found	Where
WIKI	Find the Wiki	x	
	Getting Started Guide	x	
	Bits	x	
	Adding in binary	x	
	Boolean Algebra	x	
	Base Conversions	x	
	MOS 6502	x	
	NES CPU	x	
	NES PPU	x	
Forum	Find the Forum	x	
	Newbie Board	x	
	Getting Started Post	x	
	Beginner's Tutorial		
Tutorial	6502 programming	x	
	Sprite movement code	x	
	Change BG color	x	
	Change sprite palette	x	

Question	Answer	Correct	
Bitwise			
Add 0010b+0011b	0101b	yes	
Left Bit Shift 0101b	1010b	yes	
1011b AND 0111b	1110b	no	
1100b OR 1010b	1110b	yes	explained personally
NOT 1000b	0111b	yes	explained personally
Hex Conversion			recalled this math from "new math" in the 60s still does not understand The underlying principle
0x09	9, 1001b	yes	
0x0B	11, 1011b	yes	
0x1F	31, 1111b	yes, no	

Asked for help	yes
Gave Up At Assembly	yes

POST TEST

Question	Response	Note
Overwhelmed	yes	
Documentation is..		
Well organized	sometimes	"I'm not really familiar with it, and I'm old"
easy to understand	with difficulty	
easy to find	no	"I was frustrated with going back and forth. You need to know what you're looking for." "the math is beyond my level of understanding"
Found site confusing	yes	"I found the explanations unclear"
Felt encouraged to...		
homebrew	no	
game mod	no	
program	no	
Relied on practitioner	yes, for explanations	

Appendix F: Outline of Results

Quick Background about Test Participants

Six participants were chosen for this first study based on their varying levels of experience and relationship to both gaming and programming.

Part 1: User Interface test

Overall, the participants were able to find most of the information, but generally needed help finding at least one or two webpages.

Two users technically "cheated" by modifying the web address in the navigation bar on a few occasions.

Finding information on the Wiki

Everyone found the wiki, but how they found it was sometimes different. On the main page, the links to the forum and wiki are located directly beneath the website's logo. Half of the users found the wiki from the top link. But the other users ignored the links to the forum and wiki at the top of the page and accessed the wiki through the link in the disclaimer bar warning against outdated documentation. One user did not read the warning about the outdated documents on the main page and proceeded to look for the wiki below the old documents on the main page. However she scrolled back up to the top and noticed her mistake shortly after. Another user (*1)* saw the links at the top, but almost missed them and commented he felt it was poor user interface design to put crucial links underneath the website logo.

Some people had trouble locating the getting started guide.

For 6502 information, some (2) located from the getting started guide, while the rest went back to the main wiki page

Finding information on the forum

Some users jumped ahead of my prompt to bring them back to the NESDEV homepage. They accessed the forum from the Wiki main page's link.

It might be worth noting some users tried to go back to the main page from both the forum and wiki, but were unable to access it due to a lack of navigation links bringing the user back to the main homepage.

Once users found the forum, they had little to no trouble locating the newbie help board. The wording of my question though perhaps could have been clearer, as I wrote "find the new users board" rather than "newbie help center." Users who were unclear about my meaning would scroll past the newbie help center board, and then click it assuming it was what I meant. Part of my rationale for asking this question was because NESDEV technically doesn't have a board specifically for new users to directly introduce themselves. I'm not sure if this is intentional or an oversight, but having a board where users can introduce themselves is necessary to effectively grow a community.

From the getting started guide, users were asked to find various pieces of information that would theoretically aide a newcomer in programming for the NES.

Part 2: Math and Programming Questions

A usability survey alone is not enough to determine how friendly the NESDEV community is towards newcomers. Although finding information is important for communication, the documentation itself needs to be measured for comprehension. Participants were tasked with answering a variety of questions common to introductory low level programming courses (see textbook citation). From the preliminary results, participant comprehension varied heavily based on mathematical and technical background.

Bitwise Topics

Five of the six participants answered the binary adding problem correctly, but were significantly helped.

I asked the question about left bit shifting to catch the participants off guard. From the getting started guide, bit shifts aren't inherently obvious, but can be found within pages. Most participants guessed the answer, while 2 users reprimanded me and demanded I tell them how to solve the problem.

Hexadecimal Conversions

Five of the six participants asked for help during these mathematical parts of the exam. The one who did not had prior experience with these sorts of comparisons.

Assembly Programming

During the initial pilot test, I realized people without assembly experience would be likely to give up. As a result, I decided to make the programming section of part

Everyone except for *1* gave up at this portion of the test. They would read through the information on 6502 assembly, become intimidated or scared. Most people selected for their 6502 reference material the standard

instruction.

In my experience, the documentation IS technical and off putting to newcomers. 6502 isn't a particularly complicated language compared to more modern languages either. For my recomendation, I'll discuss the practices and structure in the LC-3 textbook, an approach to assembly I found much much easier, despite about the same level of complexity as 6502 Assmbly.

I discovered the limits to new information people could take regarding mathematical topics from this exam. Burnout on assembly is frequent. Perhaps this could be viewed as a limitation of my test. Or maybe this is more telling of learning assembly itself.

Appendix G: Future Homebrew Topics of Discussion

Ways Homebrew can be Channeled for Sustainability

Homebrew as Waste Management

Significant amounts of electronic waste fill landfills across America. How much of the garbage consists of video games remains uncertain. As an example, we know for certain Atari buried 700,000 different games in an El Paso landfill, and perhaps legendarily buried over two million unsold ET cartridges in Almagordo, New Mexico. All this silicon and plastic is sitting in the dump. Not just videogames, but other cheap electronics containing embedded systems microchips and integrated circuits. Toaster ovens, microwaves, old computers, cheap calculators, cars, and much more contains the dusty relics on burned out circuit boards and perfectly recyclable integrated circuits; especially Z80s and 6502s. All this electronic waste is a major contributor to the environmental problems discussed earlier. Is there anything that could be done to manage the situation? The Atari VCS contains only contains only 128 bytes of memory. Could it be possible maybe, that the same reverse engineering applied to the 2600 could be also applied to the many embedded systems out there? Could millions of garbage electronics be gutted, recycled, and re-purposed into brand new uses the hardware manufacturers never could have conceived?

Energy Saving

The potential for conserving energy by refitting older systems makes sense on multiple levels. Both the Atari VCS and Nintendo Entertainment system were powered by 9 Volt Alternating Current. Powering these systems with a outlet in essence is a colossal waste of power best used elsewhere. Meanwhile, the low level of energy required *(the Atari has a very low amperage; I intend to experiment with how long a standard 9-volt battery can power it)* for these old systems remains a viable option. One project I saw on NESDEV involved powering the NES off a 9-volt solar panel. It was a fairly simple project relative to homebrew hacking standards.

Networking and a Smart World

Going off the energy and waste saving uses, there's the potential for networking capabilities being added to retro systems. Already for Home Computers of the era, many projects have existed. It's indeed possible to access the internet on hardware as primitive as the Commodore VIC-20 https://www.youtube.com/watch?v=HtONphrd19M. Factor in digital Ham Radio modes and a wireless network on reused electronics powered by solar energy could exist everywhere as part of the internet infrastructure.

The solar powered retro systems and recycled imbedded systems could be used for everything from ecosystems monitoring, infrastructure maintenance, creative arts and entertainment, resource and other systems monitoring, communications backups, and much more.

Open Information and New Democracy

At last we are brought to the Homebrew communities themselves. Within them, we've seen an economy built on tinkering and figuring out how to reverse the secrecy of proprietary hardware. Information and technical communication are not always professional quality, but are free. With some smart initiative on people who understand the importance of sustaining communities, the modes of communication can be refined and improved. The seeds of open information and self managed community exist within the homebrew community waiting to germinate. When they do, it will be an ultimate pilot test for self governance and open information in a community with the potential to change the world.