

Marking New Ground: Flash, HTML5 and the Future of the Web Arcade

Anastasia Salter
University of Baltimore
1420 N. Charles St
Baltimore, MD 21201
anastasia@selflound.net

John Murray
University of California Santa Cruz
1156 High St
Santa Cruz, CA 95064
john@expressive.ai

ABSTRACT

The recent outcries announcing the death of Flash, a foundational platform of the interactive web and online arcades, seemed to mark the end of an era. While such rumors of Flash's death are as-yet greatly exaggerated—the Flash plug-in is still installed on most computers, and Flash content is widely available—the gradual dismantling of the Flash platform's role in the framework of the interactive web has begun. This paper focuses on the consequences of Flash's apparent demise (and, by extension, the consequences inherent in the obsolescence of software platforms) to the web as arcade. We focus on the Newgrounds community and the emergence of HTML5 as part of a transformational moment in gaming platforms with larger implications for the creation and preservation of web gaming experiences. The stratification of genres between the platforms, the uptake both in the forum and the number of views, as well as game mechanics resulting from the affordances and ease of authoring of HTML5 libraries all provide lenses with which to understand how the rise of HTML5 as a platform will impact game design and particularly amateur development. Flash's legacy is intermingled with the emergence of HTML5 and Unity, and inexorably part of the future of the web as a platform for amateur and independent game development.

Categories and Subject Descriptors

K.8.0 [Personal Computing]: General – Games

General Terms

Design, Theory

Keywords

Flash, HTML5, web games, genres, mobile development

1. INTRODUCTION

For the past two decades, Flash established itself as a fundamental platform for online gaming, thanks in part to its near ubiquity on desktops and ready adaptability to web-based play. In software

studies a platform is defined by Ian Bogost and Nick Montfort as a hardware or software system that provides the “foundation of computational expression” (2009). The web is not a single platform: if anything, it is an interwoven mess of platforms, with each web browser, server and client-side technology, and client-side hardware impacting its use and capabilities. But the role of individual software platforms in shaping computational expression on the web cannot be underestimated. Flash's readily understood development environment, metaphors and strong user development resources made it essential to enabling amateur web game development and the arcade-like sites, including the flagship Newgrounds community that housed them.

1.1 Flash Games in Context

While hardcore gaming is primarily associated with dedicated console hardware or installable PC games that make full use of the computer's processing power, online gaming evolved to fill a more casual niche of instant gratification. The Flash platform, which for the past two decades has been a dominant browser plug-in for interactive content provides the basic affordances for interactive gameplay: display of color graphics synced to sound and capacity to handle user input. The style of Flash, with its vector graphics roots, was so distinctive that Manovich referred to “Generation Flash” and the procedural aesthetics they'd brought to the web and beyond (2005). Flash games were in many cases self-published and independently developed without a budget, which went hand in hand with the platform's original purpose. Such games are generally little more than a footnote in the history of video games (Wolff 2008), but essential to our understanding of casual games (Juul 2012). Independent (or “Indie”) game development was not a new concept when amateurs began programming Flash games, and its definition continues to be debated as indie gaming continues to be defined by outsider status even as it is more broadly integrated into the gaming mainstream (Ruffino 2013), but Flash enabled it on a highly accessible level.

Traditional consoles like the Wii, PlayStation and Xbox typically required developers to purchase hardware development kits and license expensive software development kits, in addition to providing fees to distribute the game itself. Thus game developers who identify themselves professionally are often members of larger companies, as truly independent development on consoles has historically been limited (Deuze, Martin, Allen 2007). Each console had its own graphical limitations and controller layouts, providing additional obstacles to independent developers creating games for multiple platforms. Flash, on the other hand, was developed using a single software package at the time and had no licensing fees. Flash games typically presuppose a desktop interface consisting of a mouse, a keyboard, and a monitor—a

schema that would lend itself to certain interfaces that would gradually dominate PC gaming thanks to the combination of browser-based and installed games. For instance, the use of arrow keys, the control key, and spacebar became a primary platformer combination, while many casual games relied only on the user choosing objects to manipulate with their mouse.

In part thanks to these limitations, Flash lent itself to different types of interactions than other popular gaming platforms of the time, while reaching an audience of “non-gamers” whose expectations were shaped by Flash’s controls rather than console hardware. While hardcore PC gaming took advantage of the latest joysticks and game controllers, focusing more on 3D navigation and environments, Flash creators were left free to explore as-yet-unrealized potentials of 2D gameplay. Console games continually rose their graphics specifications and complexity to meet the expectations set by high pricing, while Flash spread out to colonize each possible mechanic which could be realized using simple mouse and keyboard input. The constraint of not having access to specialized controllers or hardware forced developers seeking to use Flash to develop accessible games—games which could be immediately played by most computer users who often had Flash already installed. However, this design schema would make most Flash content hard to translate to the touchscreen phones and tablets on the horizon.

1.2 Newgrounds and Flash Gaming

Flash offers an incredible utility to the game development community even today as Daniel Plemmons, a designer for Zynga, noted:

As a toolset for rapid, low barrier to entry, creation, iteration and publishing it is hard to beat flash as a development platform. It’s a veritable swiss-army knife. Unity is probably the next best thing and it struggles in terms of web publishing (getting new users to install the unity web player is still a hurdle). There’s a huge community knowledge-base and a large variety of libraries and tools to aid development. There’s also a huge number of designers, artists, and developers who are already familiar with Flash development pipelines and optimization (that last part is key). You can probably throw a rock and build a team of people where everyone is familiar with developing in Flash (2012).

The affordances Plemmons observes in Flash are essential to its two-decade-reign as a dominate web platform, and also key to understanding Flash game content. Thanks to its suitability for small, easily distributed works, Flash lends itself to viral and remixed content. Flash works fall into what Henry Jenkins, Sam Ford, and Joshua Green have termed “spreadable media,” or media designed to be passed, remixed, transformed and moved through social networks (Jenkins et al 2012). But Flash content isn’t the only part of Flash that moves this way: knowledge about Flash development, and particularly tutorials, resources, and libraries for game development, has also spread as part of Flash communities.

While Flash games exist on a range of networks and sites, one of the first and still foremost Flash communities is Newgrounds, founded by Tom Fulp in 1995. The Newgrounds community has been a case study for analyzing online culture creation and game design thanks to its range and visibility. One study of Newgrounds as a site for collaboration and remixing noted that the Flash animation standard limits the ability of amateurs to enter into the technical discourse: the source components are packaged and inaccessible (Yardi, Luther, Diakopoulos, Bruckman 2012). This does not mean that participation within the community does not influence content, as an examination of Newgrounds animations (Paolillo, Warren, Kunz 2007) found a correlation between creators’ positions in social network cliques within the site and the genres of work they produced.

While Newgrounds bills itself as a site of “Everything, by everyone,” the actual demographics of Newgrounds participants have generally been found to be homogeneous, with creators of works primarily being young men (Kendall 2007; Luther and Bruckman 2008). However, an examination of demographics and participation in the community over time revealed that more experienced users were valued, and women were increasingly gaining social acceptance (Warren, Stoerger, Kelley 2012). Within the community, successful collaboration is modeled primarily through distributed labor and a hierarchy of influence, making the choices of experienced and visible users trickle down to the creative works of newcomers (Luther, Caine, Ziegler, Bruckman 2010). Participants of the community are part of what Bollier has termed the “viral spiral,” or “the way in which the innovation of one Internet cohort rapidly becomes a platform used by later generations to build their own follow-on innovations” (2008). The communal nature of Newgrounds makes it a key site for this type of movement. Thus ethnographic studies of Newgrounds have previously been used as a means to categorize the emergence of genres in web-based multimedia through amateur Flash production (Paolillo, Warren, Kunz 2011). If a next generation of the web arcade is to emerge, it will likely be built on top of the innovation of the Newgrounds cohort and its traditions and genres.

2. MOBILE AND HTML5

If Flash’s death rumors have a single point of origin, it is in the moment when Apple eliminated Flash from all iOS devices, single-handedly ending any hope for Flash game-makers and animators of reaching the mobile market without first adapting their content to a new platform. The decision came with a much-cited open letter from Apple founder Steve Jobs highlighting what he saw as core weaknesses of the platform, particularly “100% proprietary” nature of Flash and the touch interface: “Even if iPhones, iPods and iPads ran Flash, it would not solve the problem that most Flash websites need to be rewritten to support touch-based devices” (2010). His claim was particularly true for Flash games, whose interface assumptions had always been predicated on the availability of a full keyboard. The death of Flash seemed to be confirmed by headlines across the Internet when Adobe announced they would no longer be developing Flash plug-ins for mobile devices, with Adobe’s Danny Winokur instead endorsing HTML5 in its place: “HTML5 is now universally supported on major mobile devices...this makes HTML5 the best solution for creating and deploying content in the browser across mobile platforms. We are excited about this, and will continue our work with key players in the HTML community, including Google, Apple, Microsoft and RIM, to

drive HTML5 innovation they can use to advance their mobile browsers” (Isaac 2011). Adobe backed its announcement with an investment in HTML5 development tools in their Adobe Edge series, as well as with an extension for Flash users to help transition to HTML5 using a JavaScript library, CreateJS (Adobe 2013).

While this announcement only concerned mobile Flash, the endorsement of HTML5 seemed to point to Adobe’s intention to abandon the desktop version of Flash in the future. An Adobe employee, Mike Chambers, tried to further clarify that the future of Flash was still alive (2011):

We feel that Flash continues to play a vital role of enabling features and functionality on the web that are not otherwise possible. As such, we have a long term commitment to the Flash Player on desktops, and are actively working on the next Flash Player version. Of course, with the growth and continued improved browser support of HTML5, the role of Flash will change. We feel that for the foreseeable future, Flash is particularly strong in delivering advanced video, as well as providing a robust, and graphically rich gaming platform. We are focusing our Flash Player efforts around these areas.

Despite this reassurance, Flash developers felt justifiably abandoned by the decision, and the announcement was hailed by some as a “triumph of open standards” and a move away from proprietary technology in the future of the web (Lee 2012).

2.1 The Death of Flash

The proclaimed “death of Flash” offers us an opportunity to watch as an entire generation of games has already been rendered unplayable on some devices. A final death of Flash would mean the loss of a generation of games—a closing as final as the deaths of physical arcades. While efforts are ongoing at preserving games as cultural artifacts, such as the Preserving Virtual Worlds project (NDIIPP, UIUC, Maryland, RIT, Stanford) and the Preserving Cultural Software Objects project (ODH, UCSC), online games tend to be the most ephemeral of game objects.

The Newgrounds community acts as archivist and preserver of games, a trend familiar to the commons-oriented movement to keep videogame culture and history playable (Coleman, Dyer-Witheford 2007; Guttenbrunner, Becker, Rauber 2010). However, such efforts become impossible to sustain if the Flash plug-in itself dwindles into obscurity or is rendered incompatible with modern browsers and devices. The extensive archive of Flash games operates on assumptions of interface (the presence of keyboard and mouse) that once seemed unassailable: however, the rising demand for touchscreen devices suggests that might no longer be the case as “natural user interfaces” become more common (Wigdor and Wixon 2011). Just as consoles with dedicated and proprietary interfaces are difficult to preserve for the future through emulation (such as Microsoft’s Kinect or classic arcade games), Flash games could be rendered inoperable

by changes in the dominant paradigm—just as their online cousins, massively multiplayer role-playing games, have proven particularly resistant to archiving or preservation (Winget 2011). Questions of preservation and archiving games are furthermore not merely academic: in a community such as Newgrounds, where innovation builds, parodies, and constantly remixes the past, the community’s collective memory is a resource.

As of December 2013, the Newgrounds site is host to games alongside movies, audio, and art, with a focus on game genres familiar from arcades: Action, Adventure, Gadgets, Puzzles, Rhythm, Simulation, Skill, Sports, and Strategy. There’s also a small collection of games produced this year designed as tutorials, which is still entirely dedicated to Flash and includes introductions to the Flash library Flixel, creating inventory systems, and simple utilities such as sound design or buttons in games (Newgrounds 2013). The most populated game category for games produced in 2013 is Action, with subgenres including Fighting, Platformer, Shooter and Other. This high level of activity supports the continued relevance of Newgrounds as one of the major web arcade hubs, with new content highlighted on “Best Of” lists each month often receiving between 20,000 and 150,000 views. This is not a portrait of a community on a death spiral, but it is a community faced with a moment of change alongside an incredible legacy of investment in Flash’s future.

2.2 HTML5 as Platform

HTML is not a new platform, as it remains a foundation of the web: however, the specification for HTML5 and the extensions of CSS3 and Javascript as game development languages are much more recent. As a relatively new platform for gaming, HTML5 has received relatively little attention in scholarly study. The greatest emphasis when discussing HTML5, CSS3 and Javascript as a software platform is on its promise of independence from hardware requirements, particularly when targeting multiple mobile devices (Santanchè et al 2013). HTML5 is drawing attention as a dominant platform for pervasive media, thanks to its advantage of cross-platform compatibility over more demanding native solutions for iOS or Android (Melamed and Clayton 2010). However, scholars and designers note the challenges for the end-user (particularly the amateur developer) in building meaningful interactive works on these platforms given the current level of implementation (Jazayeri and Ahmadi 2011).

Technical frameworks such as WebGL and WebSockets offer a potential for 3D in the browser that Flash itself only begun to explore (Curran and George 2012). Ahmadi, Jazayeri and Repenning (2012) noted that many of these early HTML5 games are professionally developed, as opposed to the more amateur world of Flash games, but the introduction of higher level development environments could change that trend. Early tools are being tested and distributed for simple HTML5 game design, such as Ahmadi’s AgentWeb (2011), but when compared to the tools that surround Flash game production they significantly fall short. Professional early adopters such as Chrome Experiments and Mozilla’s MMO Browserquest are outliers in their use of dedicated and individualized engines, which do not lend themselves to remixing due to their complexity and specialization.

It is impossible to fully consider the significance of HTML5 without also considering how mobile interfaces transformed expectations of game development. Games built for the web arcade audience could not possibly translate easily to a

touchscreen interface, as even onscreen keyboards do not contain the arrow keys that are essential to nearly every arcade-style game. Imitations of traditional controls with simulated buttons are one option for touch screen designers, although a small study by Browne and Anand found those less successful for users than accelerometer-based interfaces (2012). Jobs’s reminder that most Flash works cannot run on touchscreens for reasons as simple as the need for a mouse in order to use functions like mouseovers is only the beginning of the interface rabbit hole.

3. CASE: HTML5 ON NEWGROUNDS

HTML was first introduced as an officially supported format for Newgrounds in December of 2012. Tom Fulp announced support along with a few initial games, including Dual Custody (Badgergames 2012), AirScape (sqiddster 2012), and Adventures of Froghead (Lost-Head-Studio 2012). In the history of Newgrounds, Tom Fulp recorded these decisions as part of securing the future of Newgrounds (2012 December):

We did...roll out some big features that shake up the dynamic of how people have viewed NG for the past decade. In mid-April we introduced our video player, allowing for the first time ever movies that weren't in SWF format. Improvements have been ongoing, with our video conversion server slated to launch by the end of the year, allowing for much more mobile-friendly versions of existing videos. We are also beginning an initiative to create video versions of all the classic NG content, so that it may exist forever on future platforms that may not support Flash. In October we unveiled support for HTML5 games on NG, making it official that we won't live or die by Flash, as much as we still love it.

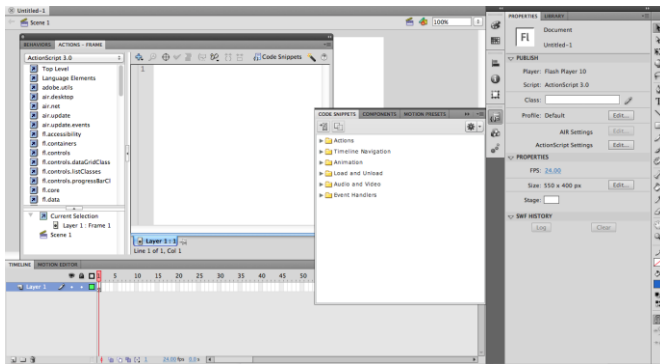


Figure 1. Adobe Flash Professional CS5 User Interface

The early games Fulp pointed to in his initial announcement were not groundbreaking examples of HTML5, but they did promise greater accessibility in the future for developers to offer games from the platform of their choice. However, Fulp further noted that of all of the early HTML5 games, only one “has the unique honor of being the ONLY game on NG that you can play on an iPhone. The other HTML5 games require a keyboard” (2012 October). The reactions in the forums were ongoing and often

resulted in intense arguments that debated the readiness of HTML5 with the openness and technical capability of Flash. Many felt that HTML5 was being “thrust” upon them by Steve Jobs in an act of business competition. Adobe was likewise blamed for the platform's perceived demise. Others pointed out that HTML5 also has significant flaws and security vulnerabilities (Elitistinen 2013). Despite being supported unofficially for over a year, HTML5 games are still relatively scarce compared to the more familiar SWF (Flash) format among contributors to Newgrounds, and they also receive, as a consequence, relatively fewer views. The most popular games, many of them online games produced by major studios, are almost universally in Flash. In this section, we will describe the development tools used in Newgrounds as well as representative games created with them. We will compare these entries with the subgenres established by Flash and consider the larger implications for amateur developers as well as established Flash developers in light of the changing platforms.

3.1 Methods

To gain a clearer picture of HTML5’s acceptance and growth as a platform within the Newgrounds community, we examined every game tagged with “HTML5” or included in the HTML5 game collections and competition listings. This method is not exhaustive, as not every game is appropriately tagged with its platform, but it does provide a working sample of current games.

We examined the dominant mechanics, genres, mobile-compatibility, and interfaces of games that attracted either a significant amount of community discourse or popularity relative to other games in the category. This study of HTML5 as a platform builds on our recently completed work on Flash (Salter and Murray 2014) as a platform integral to the existence of web content communities of this kind. As Newgrounds is an exemplar network for Flash, its acceptance and integration of HTML5 as a platform is particularly revealing of the future of web arcades.

3.2 Evolving Community

The Newgrounds community has an identity that focuses on the production of content and the continual improvement and feedback on that content. As a result, there is a lively forum which

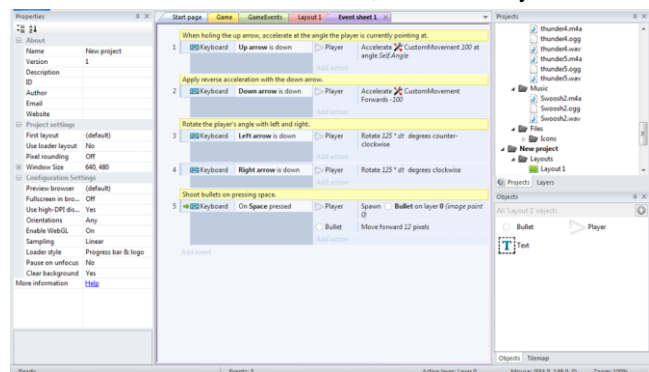


Figure 2. Construct 2 User Interface

has topics ranging from programming to collaborating virtually with teammates and monetization. The major inflection point for emerging HTML5 toolkits and libraries will be when users create their own independent tutorial sites and generate a level of “fandom” that Flash enjoyed through such sites as FlashKit and ActionScript.org and a half dozen other non-Adobe sites. The process of creating a single community around HTML is still in

its infancy in the gaming community, marked by the competing toolsets and the different skills and experience required to work in each. Many gaming toolkits have been targeted at amateur developers, appealing to the expressed desire to not have to code in order to prototype or even realize the games that they want to produce. Between Game Maker and Construct 2, game development environments have been moving to these coding-optional or coding-guided models of game development. "No coding required!" appears in large, rather friendly type on the front page, and a full bouquet of tutorials and helpful documentation await further on in the website.

Figures 1 and 2 juxtapose the interfaces of Flash and Construct 2 in their development set-up. Many elements are similar in graphical feel, and the transition for Flash developers is eased by those connections. However, Construct 2 is a dedicated game environment, with a more limited set of features and a system that eliminates Flash's characteristic timeline. Both interfaces involve the direct manipulation of visual content, which is essential for making it more accessible to casual and artistic users. Other subsets of Flash, such as Flixel, more strongly enable the mechanisms of Construct 2: Flixel reduces some of the difficulty of developing with Flash by providing libraries and features for adding tilemaps, scrolling, and other features of object-oriented games without writing code from scratch (Heaven 2013).

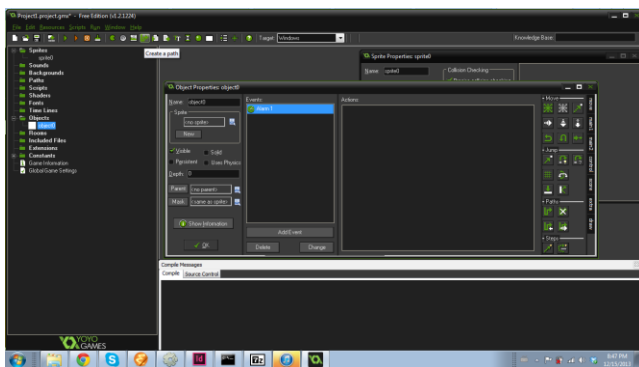


Figure 3. GameMaker User Interface

Construct 2 is the listed development tool for at least half of the current HTML5 games on New Grounds, and is listed along with several Flash alternatives to Adobe tools on the development section of the website. Construct 2's approach to facilitating game design combines the cross-platform appeal (and ease) of Unity with the user-friendly drag and drop of development designed specifically for gaming provided by GameMaker, another major environment often compared with Flash (Guimaraes 2011). Microsoft's Kodu provides another analogue for simple programming models which Construct 2. Figure 3 shows the user interface of Gamemaker, which streamlines the development process while maintaining a similar interface. While Flash has evolved to become a tool for gaming, each of these tools were designed from the beginning with gaming genres and conventions in mind. The idea of moving a sprite around with the arrow keys is fundamental to their vocabulary, and so it is also the easiest task to accomplish. In Construct 2, the development environment treats sprites as objects to which you can assign physics and various constraints without writing textual code. These constraints can include whether a player character can land on them, kill them, or any number of other triggers and pre-defined behaviors.

The difference for authors between creating a game starting from an animated Flash movie starting point is still extreme. Flash began primarily as an animation tool which expanded, step by step, into a platform capable of handling the needs of games and new media. While it was an early target for third party libraries, most of the initial libraries for Flash, including box2d, are also available free. Construct requires an investment on the part of the developer in a premium license to even collect any sort of kickback from hosting a game on Newgrounds. The most basic license is free, but is somewhat hobbled. In order to sell games made with it, or to use it on iOS or with more than a certain complexity, a license is required. The openness which implied freedom on the World Wide Web promised to consumers in a post-Flash world is not the same for the competing HTML frameworks.

3.3 Popularity of subgenres

Flash has been responsible for an explosion of experimental, short form genres which have given the game developers their start. What sets an HTML5 game apart enough that it's identity as an HTML game, either in conjunction or even perhaps alone, becomes desirable to associate with the experience? For instance, despite talk, there never appeared an HTML5 gaming portal for the site, and many of the games which are html are inconsistently labeled – some are labeled construct2, while others are labeled html5. The capabilities of HTML are still a strict subset of that of Flash – so while it is very likely that casual games that take advantage of touch may emerge, it is unlikely a radically new subgenre will emerge solely from the benefits of HTML. The lack of a simple interface for vector-based tweening suggests that sprite-based games are more efficient, and so animation will persist as a video instead of a core tween. But despite the support of HTML as a format to distribute games, the resulting offering is still not growing as fast as it could.

There are several factors which prevent HTML games on New Grounds from being as numerous or as popular as their Flash counterparts. One of these, the ability to submit an HTML game as an archive, has recently been moved from experimental status to a full fledged feature. Newgrounds has pioneered the use of achievements, scoreboards and other methods of integrating the site content with the user profiles. The site also relies on revenue sharing through the use of ads with games that are hosted on the site. These capabilities grew up with the site's relationship with Flash, and so the ability to link a new HTML game into the same APIs isn't there yet. The site's extra features, including achievements and advertisements, are linked through the traditional Flash APIs. This lack of official support, coupled with relatively low number of HTML games that have come out over the last year suggests that the major stakeholders in HTML gaming are in fact initially professional companies that can both afford the licenses or technically proficient amateurs who are willing to implement their own cross-browser code to achieve their desired effect. The sheer number of frameworks and libraries for JavaScript provide fewer clear paths to becoming an expert in making HTML games, at least without becoming an expert in a particular tool such as Construct 2. The Flash community maintains its presence and shared values through the continued activity on the fora dedicated to AS2 and AS3, as the majority of the newcomers are greeted with the same answers about the tools, and the large amount of pre-existing tutorials, especially for AS2, are still available with source code. Construct 2 does not enable

an easy way to share source code, though you can create and share a plugin.

3.4 Example HTML Games on Newgrounds

We will consider closely a few of the most popular HTML5 games on Newgrounds which have been released in the last year and their contribution to the transition. Specifically, how popular were these games, and how much experience did the more successful authors have in a prior medium (such as Flash) before releasing their first HTML5 game? The first of these was one of the very first Construct 2 games on Newgrounds: AirScape. AirScape was released in October 15th, 2012 and has accumulated just over 60,000 impressions. AirScape is a simple platformer which uses gravity as the primary mechanic. The player jumps from one “ground” to another, avoiding obstacles and otherwise progressing through levels. The author also created a tutorial on Construct 2’s web page describing how he used the existing code “blocks” to describe a variable gravity field based on where you were in the level. AirScape provided an advertisement in Unity for a while.

Another popular game is based on customizations of non-gaming JavaScript libraries including jQuery. A Grain of Truth, by the Rudowski brothers, is characterized by an expansive, comic-driven aesthetic that speaks to webcomics more than it speaks to shmups and tower-defense games. The feel is closer to Myst, and the attention to detail ranges from the timing of events and the hand-drawn feel of the vistas to the cursor feedback itself. This game indicates a very different direction than AirScape, one which takes casual gaming into a longer form with the ability to save and restore the game seamlessly. The critical reception from the Newgrounds community mostly commented on the difficulty of its mahjong-inspired puzzle and some issues relating to crashing. The comments were full of possible solutions, most of them using a specific browser and browser version. This game clearly was using the latest capabilities, and as a result, many of the players were frustrated despite loving the design, art and music.

One of the most popular non-Construct 2 games is *Secret Santa* by Nutcasenightmare. The objective of the game is to turn various platforms into ice, thereby reducing their friction and moving the Santa Claus to his goal. This game not only follows simple physics based gameplay familiar to those fond of Angry Birds or Cut The Rope, but also includes a means to create new puzzles. The creators even held a contest for the best user-created levels, of which sixty were submitted. Unlike A Grain of Truth, Secret Santa follows more of a structure traditionally understood as a mobile game, and the interface’s simplicity (which, needing to be mobile, does not rely on mouse hovers) means that it takes advantage of the multi-platform appeal of HTML over Flash. Like the original chrome version of Angry Birds, Secret Santa demonstrates that HTML gaming is capable of high framerates and performance

And finally, Magnetized was designed from the ground up as an entry for the Construct 2 Touch Jam by rocky10529. The gameplay is simplicity itself; the only way to interact with the game is to touch the screen. The exact position, angle or velocity do not matter. Instead, the presence of your finger anywhere on the touch screen (iPad or iPhone) will trigger the same response as if it were touching anything else. This works as the player is informed that only the highlighted magnet would be activated by a touch, but it also has the added effect of reducing the complexity of the gameplay itself, as you are only concerned with predicting

the simulation of forces and aren’t worried about where your finger might fall.

Magnetized echoes the same social structure as many mobile games, creating bite-sized chunks of time and filing them with an addictive game mechanic. The Flash Community has remained divided amongst those who are unwilling to jump onto HTML5 until they are more comfortable with their understanding of the DOM and browser differences. The HTML5 games reviewed here indicate that though they lack some of the aesthetics of the vector based tweens found in Flash, they can demonstrate equally compelling game mechanics and further be designed to take advantage of HTML5’s text rendering, mobile capabilities and even novel game design tools such as Construct 2. These benefits are still riddled by browser compatibility issues, a distinct fragmentation of the game development tools and communities and ultimately the ongoing debate over whether Flash is still alive and kicking or if rumors of its demise have been greatly exaggerated.

4. CONCLUSION

The future of Flash as a platform is tied up with the innards of the World Wide Web, and particularly engrained with communities such as Newgrounds. The lifespan of Newgrounds has been exemplary, and as a dominant web arcade its fate is important to the future of game studies and web gaming. Newgrounds has been no stranger to controversy: its status as an open community has allowed it to be a home to playable content that would not be welcome on controlled platforms such as Apple’s walled garden. For games with violence and sexual content that might not be so welcome on the App Store, the web arcade remains an essential point of contact and opportunity for the truly amateur or independent developer to find an audience.

Flash’s proprietary status didn’t keep it from being one of the cornerstones for a community of remixers, collaborators, and experimenters. However, the open status of HTML5 might be better suited in the long run to the community’s future, and the potential for Bollier’s conceptual “viral spiral” to continue as more genres make their way into HTML5. The increased available of tools for developers along the lines of Construct 2 will be essential to the establishment of an HTML5-driven web arcade, and the lack of consistency in choice of development environment will make it difficult for a culture of knowledge sharing to emerge quickly that can rival the incredible resources surrounding Adobe’s Flash.

Some developers have been quick to dismiss the possibility of Flash’s death in web gaming. Paul Preece, who defined the tower defense genre with his Flash game Desktop Tower Defense, was quoted in Edge magazine defending the future of Flash (2010):

Of course, Flash should never have been required to play video on the web in the first place. But game developers were fortunate that it was needed, as it allowed a sophisticated development platform to be distributed to 99 per cent of computers by piggybacking on the coat tails of video support. Web-based games wouldn’t be anywhere near as pervasive as they are now if HTML had supported video back in 2005 when YouTube launched. Now that

Flash has established its own momentum as the default games platform for the web it will not be affected much by video support moving from the Flash plugin to the browser. The debate is a red-herring in relation to webgames.

The web is not a single platform: if anything, it is an interwoven mess of platforms, with each web browser, server and client-side technology, and client-side hardware impacting its use and capabilities. Apple's ability to effectively ban Flash from the iOS ecosystem is a testament to how much the client hardware and software choices shape the face of the web. While Preece and other Flash developers may be correct that their platform of choice is not leaving quickly, the consequences of increased browsing on touchscreen devices will definitely be felt in game development.

The emergence of HTML5 can feel like a step backward for game development, as even Construct 2 is still far removed from the level of support and community resources that Flash game developers can rely on, and thus many developers are still occupied with inventing the wheel rather than revolutionizing their games and updating genres to reflect the new possibilities and limitations that touch screen devices offer. HTML5 in general suffers from a troubled platform legacy, as it depends heavily on scripting languages that were never intended to be used for the roles they've evolved to fill. However, it does have both the advantage and disadvantage of being an open standard, without any one corporation dedicated to mapping out its evolution over the next decade: like HTML5 games themselves, it is a platform whose future can be guided by communal efforts. But the presence of HTML5 on Newgrounds is growing, and the community can build on this early foundation. Over the next few years, HTML5 gaming looks poised to experience a strong surge, and with it a transformation that can bring the web arcade to a new generation of gamers and platforms.

5. REFERENCES

- [1] Adobe. Flash Professional CC / Toolkit for CreateJS. *Adobe Creative Cloud*. <http://www.adobe.com/products/flash/flash-to-html5.html>
- [2] Ahmadi, N. (2011). Beyond upload and download: enabling game design 2.0. In *End-User Development* (pp. 371-374). Springer Berlin Heidelberg.
- [3] Ahmadi, N., Jazayeri, M., & Repenning, A. (2012, March). Performance evaluation of user-created open-web games. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing* (pp. 730-732). ACM.
- [4] Bollier, D. (2008). *Viral spiral: how the commoners built a digital republic of their own* (p. 96). New York: New Press.
- [5] Browne, K., & Anand, C. (2012). An empirical evaluation of user interfaces for a mobile video game. *Entertainment Computing*, 3(1), 1-10.
- [6] Chambers, M. (2011). Clarifications on Flash Player for Mobile Browsers, the Flash Platform, and the Future of Flash. [Blog post.] <http://www.mikechambers.com/blog/2011/11/11/clarification-s-on-flash-player-for-mobile-browsers-the-flash-platform-and-the-future-of-flash/>
- [7] Coleman, S., & Dyer-Withford, N. (2007). Playing on the digital commons: collectivities, capital and contestation in videogame culture. *Media, culture & society*, 29(6), 934-953.
- [8] Curran, K., & George, C. (2012). The Future of Web and Mobile Game Development. *International Journal of Cloud Computing and Services Science (IJ-CLOSER)*, 1(1), 25-34.
- [9] Deuze, M., Martin, C. B., & Allen, C. (2007). The professional identity of gameworkers. *Convergence: The international journal of research into new media technologies*, 13(4), 335-353.
- [10] Edge Staff. (2010). Flash in the Pan? *Edge Magazine*. <http://www.edge-online.com/features/flash-pan/>
- [11] Elitistinen (2013). The Dangers of HTML5 [Forum post]. *Newgrounds*. <http://elitistinen.newgrounds.com/news/post/841600>
- [12] Fulp, T. (2012 October). HTML5 Games & Pumpkin Carving [Forum post]. *Newgrounds*. <http://www.newgrounds.com/bbs/topic/1321541>
- [13] Fulp, T. (2012 December) Newgrounds Wiki: History. *Newgrounds*. http://www.Newgrounds.com/wiki/about-Newgrounds/history#wiki_toc_5
- [14] Guimaraes, M. A. (2011, March). Game development with Game Maker, Flash and Unity. In *Proceedings of the 49th Annual Southeast Regional Conference* (p. 9). ACM.
- [15] Guttenbrunner, M., Becker, C., & Rauber, A. (2010). Keeping the game alive: Evaluating strategies for the preservation of console video games. *International Journal of Digital Curation*, 5(1), 64-90.
- [16] Harrison, M. (2012). Seriously, What's a Data Scientist? (and The Newgrounds Scrape). *Everyday Analytics*. <http://www.everydayanalytics.ca/2012/12/seriouslywhatsadatascientist.html>
- [17] Heaven, D. (2013). Let there be games. *New Scientist*, 219(2932), 19-20.
- [18] Issac, M. (2011). Jobs Was Right: Adobe Abandons Mobile Flash, Backs HTML5. *Wired: Gadget Lab*. <http://www.wired.com/gadgetlab/2011/11/adobe-kills-mobile-flash/>
- [19] Jazayeri, M., & Ahmadi, N. (2011, June). End-user programming of web-native interactive applications. In *Proceedings of the 12th International Conference on Computer Systems and Technologies* (pp. 11-16). ACM.
- [20] Jenkins, H., Ford, S., Green, J., & Green, J. B. (2012). *Spreadable media: Creating value and meaning in a networked culture*. NYU Press.
- [21] Jobs, S. (2010). Thoughts on Flash. *Apple*. <http://www.apple.com/hotnews/thoughts-on-flash/>
- [22] Juul, J. (2012). *A casual revolution: Reinventing video games and their players*. The MIT Press.
- [23] Kendall L. (2007) Colin Mochrie vs. Jesus H. Christ: Messages about masculinities and fame in online video conversations. *Proceedings of the Fifth International*

Conference on Cultural Attitudes Toward Technology and Communication: 1–20.

- [24] Lee, T. B. (2012). The Death of Flash and the Quiet Triumph of Open Standards. *CATO*.
<http://www.cato.org/publications/commentary/death-flash-quiet-triumph-open-standards?print>
- [25] Luther K, Bruckman A. (2008) Leadership in online creative collaboration. *Conference on Computer Supported Cooperative Work*.
<http://www.cc.gatech.edu/elc/papers/luther/fp592-luther.pdf>
- [26] Luther, K., Caine, K., Ziegler, K., & Bruckman, A. (2010, November). Why it works (when it works): success factors in online creative collaboration. In *Proceedings of the 16th ACM international conference on Supporting group work* (pp. 1-10). ACM.
- [27] Manovich, L. (2005). Generation flash. *Total Interaction*, 67-77.
- [28] Melamed, T., & Clayton, B. (2010). A comparative evaluation of HTML5 as a pervasive media platform. In *Mobile Computing, Applications, and Services* (pp. 307-325). Springer Berlin Heidelberg.
- [29] Montfort, N. and Bogost, I. (2009). *Racing the beam: The Atari video computer system*. Cambridge, MA: MIT Press.
- [30] Paolillo, J. C., Warren, J., & Kunz, B. (2007, January). Social network and genre emergence in amateur flash multimedia. In *System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on* (pp. 70-70). IEEE.
- [31] Paolillo, J. C., Warren, J., & Kunz, B. (2011). Genre emergence in amateur Flash. In *Genres on the Web* (pp. 277-302). Springer Netherlands.
- [32] Plemmons, D. (2012). Will HTML5 ever beat Flash in Games? On *Quora*. <http://www.quora.com/Video-Games/Will-HTML5-ever-beat-Flash-in-games>
- [33] Ruffino, P. (2013). Narratives of independent production in video game culture. Loading... The Journal of the Canadian Game Studies Association 7(11). Retrieved from <http://journals.sfu.ca/loading/index.php/loading/article/view/120/155>.
- [34] Salter, A. and Murray, J. (2014). *Flash: Building the Interactive Web*. Forthcoming, Cambridge, MA: MIT Press.
- [35] Santanchè, A., Boulanger, R., Viana, G., Panaggio, R., Melo, B., & Aboud, H. (2013, November). Mobile development using web technologies focusing on games. In *Proceedings of the 19th Brazilian symposium on Multimedia and the web* (pp. 13-14). ACM.
- [36] Warren, J., Stoerger, S., & Kelley, K. (2012). Longitudinal gender and age bias in a prominent amateur new media community. *new media & society*, 14(1), 7-27.
- [37] Wigdor, D., & Wixon, D. (2011). *Brave NUI world: designing natural user interfaces for touch and gesture*. Elsevier.
- [38] Winget, M. A. (2011). Videogame preservation and massively multiplayer online role-playing games: A review of the literature. *Journal of the American Society for Information Science and Technology*, 62(10), 1869-1883.
- [39] Wolf, M. J. (Ed.). (2008). *The video game explosion: a history from PONG to Playstation and beyond*. ABC-CLIO.
- [40] Yardi, S., Luther, K., Diakopoulos, N., & Bruckman, A. (2008, November). Opening the black box: four views of transparency in remix culture. In *CSCW Workshop on Tinkering, Tailoring, & Mashing: The Social and Collaborative Practices of the Read-Write Web*.
http://qualr.kurtluther.com/pdf/cscw08_w12_yardi.pdf