Online First
2019

Communication
Design
Quarterly

Published by the Association for Computing Machinery
Special Interest Group for Design of Communication
ISSN: 2166-1642

Game Design Documentation: Four Perspectives from Independent Game Studios

Richard Colby
University of Denver
richard.colby@du.edu

Rebekah Shultz Colby
University of Denver
rschultzc@du.edu

Published Online May 20, 2019
CDQ 10.1145/3321388.3321389
This article will be compiled into the quarterly publication and archived in the ACM Digital Library.

Communication Design Quarterly, Online First
https://sigdoc.acm.org/publication/

SIGDOC
Special Interest Group for Design of Communication

Association for Computing Machinery
Advancing Computing as a Science & Profession

Game Design Documentation: Four Perspectives from Independent Game Studios

Richard Colby
University of Denver
richard.colby@du.edu

Rebekah Shultz Colby
University of Denver
rschultzc@du.edu

ABSTRACT

Changes in technology, development philosophy, and scale have required game designers to change how they communicate and mediate design decisions. Traditional game design studios used an extensive game design document (GDD), a meta-genre that described most of the game before it was developed. Current studies suggest that this is no longer the case. We conducted interviews at four independent game studios in order to share their game design documentation processes, revealing that, while an exhaustive GDD is rare, the meta-genre functions are preserved in a variety of mediated ways.

CCS Concepts

• Software and its engineering~Documentation

Keywords

game design, game design document, genre, workplace studies, documentation, technical communication

Manuscript received September 17, 2018; revised January 24, 2019; accepted January 31, 2019. Date of publication May 20, 2019.

CDQ 10.1145/3321388.3321389

INTRODUCTION

While game developer design processes have received a fair amount of professional and popular attention in online discussion boards and books (a casual search suggests thousands of examples), scholarly study of actual practice is not as widespread. There are myriad of complicating factors that often inhibit such study. For one, changes in technology, scale, and expertise in the maturing industry have led to a constantly shifting landscape (Walinsky, 2016). This is compounded by the fact that many in the industry find it difficult to put processes into words (Kultima, 2010) or "games themselves have become a kind of lingua franca" (O'Donnell, 2014, p. 42), which is to say that comparisons to and interpretations of other games are primarily used when communicating design decisions. This leads to "autobiographical design" (Hagen, 2010), leaving designers with idiosyncratic or subjective experience as the primary way to communicate game ideas.

In addition to the dynamic landscape and game-centric process descriptions is the secrecy under which most game studios operate. Game studios seek not only to protect their intellectual property before it can be fully realized in a game making researcher access difficult (Ruggill, McAllister, Nichols, & Kaufman, 2017), but their design processes are often treated as propriety endeavors (O'Donnell, 2014), protected from other studios that might want to steal the recipe. After a four-year participant observation study of various game studios, O'Donnell (2014) described game development as a labyrinth, writing, "Developers have typically made a headlong plunge in, with no way to get back, or even untangle where they have been" (p. 78).

This headlong plunge is a vexing problem for technical communication research as there is often an absence of persistent artifacts and documentation within a game development process. Greene and Palmer (2014) surveyed nine industry professionals about their design processes and found that, while design documentation was used, there was no industry standard, and their survey method did not provide them access to these documents. Christensen, Cootey, and Moeller (2007) compared a case study of a professional game design studio with one of students, who are not the most accurate representation of professional design practices. O'Donnell (2014) makes only a brief mention of a hypothetical wiki page as a type of design documentation. While not a specific focus of their study, Ruggill, McAllister, Nichols, and Kaufman (2017) named a variety of written documentation types such as "central vision documents" (loc. 646) in programming, "art test documents" (loc. 1641) in art, "milestone documents" (loc. 3881) in production, "design documents" (loc. 4775; loc. 4278) in QA, and various legal documents in business, and yet, these references are the extent of what we hear about how those documents mediated the design processes of their informants. McDaniel and Daer (2016) conducted a three-month, single case study of n-Space, a venerable game studio responsible for over 40 games. In their examination of development discourse at the studio, McDaniel and Daer found a hodgepodge of written and interactive assets at the center of a cross-disciplinary and cross-cultural process responding to technical constraints and conflicts, but they did not share the specifics of those documents.

In studies of game studios, communication between programmers, artists, designers and managers was a common topic (McDaniel & Daer, 2016; O'Donnell, 2014; Greene & Palmer, 2014; Ruggill, et al., 2017; Hagen, 2010). And yet, that communication is often ephemeral, and the resulting game the only lasting manifestation of the collaboration. How game studios communicate internally during design and production processes is a compelling question of importance to technical communicators and those studying workplace communication. While larger studio design practice is of interest, a few studies have offered that perspective (Hagen, 2010; 2009; O'Donnell, 2014; McDaniel & Daer, 2016). Smaller studios better reflect an activity of practice where studio programmers, artists, designers, and managers have more input on the overall game design and, as a result, we might see different methods for communicating design decisions. Furthermore, many game developers start at smaller studios, and increasingly, large studios will often contract with smaller studios to create assets (O'Donnell, 2014), suggesting that both large and small developers will find benefit in such study.

In this article, we begin by charting a brief history of game design processes and documentation. In particular, we describe how game design documentation serves as a type of meta-genre (Giltrow, 2002) that persists in various forms to mediate design decisions even as the genre changes over time. We then turn to four interviews we conducted with small to medium sized independent game studios to show design documentation motives and practices. As we show in these interviews, there are opportunities for technical communicators to both learn from and contribute to design communication practices in game design studios.

The Game Design Document

In the history of game design, we have witnessed two historical shifts, each with concomitant documents mediating game development. The earliest games were often designed by a single author who programmed the game mechanics, the graphics, sound, and story. They designed games mostly for themselves (Meier, 2016), and wrote notes mainly to themselves. Richard Garriott, designer of the Ultima series, preserved a large selection of these notes and design documents that he donated to the special collections at the University of Texas Austin. With the commercial success of games that followed, and the increasingly sophisticated expectations of audiences, larger teams were required, and with it, more audience-focused and formalized documentation. The resulting design documentation earned status as its own genre: the Game Design Document (GDD).

The GDD took on significance as it defined narrative-heavy and puzzle games that larger studios began to develop in this first shift in the history of game development. As Sansone (2014) points out in his analysis of GDDs from this period, they contained scripts, story beats, puzzles, game flow, and visuals, mediating on paper the content that artists and programmers integrated into the final game. The GDD became an important part of game design as it began to give game developers a framework to communicate their design decisions, a descriptive blueprint, with varying levels of specificity depending on the rhetorical exigence. As such, the GDD functions as a meta-genre that directs game design as an activity. Meta-genres provide background knowledge in how to produce and distribute genres within an activity system (Giltrow, 2002; Russell & Yanez, 2002), but they also mediate the uptake and circulation of other genres within its genre system (Bazerman, 1994), specifically genre ecologies, which are "an interrelated group of genres. . . used to jointly mediate the activities that allow people to accomplish complex objectives" (Spinuzzi & Zachry, 2000, p. 172) and include even unofficial genres needed to complete a project such as white board notes or dynamic wikis (Spinuzzi, 2004). The GDD is an amalgamation of situated expressions that function as an antecedent to the game that it ultimately helps define (Giltrow, 2002).

As a meta-genre, there is no consistent set of genre conventions, only typifications that are stabilized through practice (Schryer, 2002), but Tim Ryan (1999, October 19; 1999, December 17) on Gamasutra, the foremost online resource for working game development professionals, wrote a comprehensive breakdown of his approach to the GDD. Ryan describes two main genre functions of the GDD: a game proposal (which acts as a game vision statement for the development team and publishers) and technical specifications (See appendix for a full outline of a traditional GDD, according to Ryan). The game proposal begins with an introduction, which acts as a vividly written one-sentence game pitch, a background that describes similar existing games, a game description, key features of the game that make it unique, an explanation of the game's genre, a list of platforms the game can play on, and concept art for the game (Ryan, 1999, October 19). The technical specifications include game mechanics, which describes the core gameplay, game flow, or a detailed description of what happens when the game is played, characters, physics, which should stay descriptive without including actual code, and a description of how artificial intelligence in the game should act. The technical specifications also cover a description of the game's interface, music, art, and video. This prescriptive top-down approach is intended to mediate the uptake of the situated expressions in the GDD across multiple specialized development teams: the artists need only reference the interface and art, whereas the level designers need only reference game flow and the physics (Ryan, 1999, December 17).

Because the activity rules that meta-genres include are often both tacit and explicit, they are hardly prescriptively static or codified. Instead, the genre conventions and stated rules within meta-genres are constantly and contingently renegotiated by their discourse community as the purposes and needs for different genres within the activity system change over time (Giltrow, 2002; Swales, 1990). Resistance to specific genre forms arises when contexts and social requirements for the genres change. During the early game design period, coding was so laborious and expensive that most software development consisted of the waterfall design process, where developers carefully programmed one section of code at a time before moving on to the next. Within this design model, GDDs needed to carefully document every part of the design process before programming began. To reflect the waterfall design process, GDDs catalogued what Spinuzzi and Zachry (2000) would classify as a closed-system. This design method, however, created a significant problem in game design as it led to game mechanics that limited player agency. Noted game designer Richard Garfield highlights this problem when he compares games to puzzles: "the most a puzzle solver can aspire to in general is what the puzzle designer intended. An excellent game will allow them [players] to surpass and surprise the designer" (quoted in Scholder & Zimmerman, 2003, p. 18). Thus, a "good" game is a complex and dynamic system. Carefully documenting every choice, even those not considered by the designer, would be difficult if not impossible. Thus, the technical specification function of the GDD as an exhaustive catalogue soon fell out of favor for many projects simply because it became so large that "nobody bother[ed] to read it" (Hagen, 2010, n.p.).

In addition to changes in context, diversity in and development of technologies, philosophies, and scale have led to design processes that do not always benefit from exhaustive pre-planning, leading to the next shift and current practice in game development. Two technological developments in particular have changed game design drastically. First, most current games are based on a modular design. Artists do not need to code their visuals—they can use tools to create and design the visual elements of the game and then insert those visuals into the game. Sound and music are similarly modular so that different parts of the process can more easily be added or removed at any stage of design (Kauffer & Butler, 1996). Additionally, game design tools such as the Unreal Engine and Unity have made software design more efficient, allowing for rapid prototyping, and thus eliminating the need for exhaustive pre-planning. As a result, the GDD has shifted mainly to the game proposal function (Ryan, 1999, October 19) or vision document: a way to keep a team working on multiple, specialized modules in sync, as it includes a conceptual plan for graphics, sound, mechanics, and narrative and explanations for how they fit together to achieve the game's central premise (DeAnda & Kocurek, 2016; Hagen, 2010). The GDD has even been used as a rubric, with Carolyn VanEseltine describing how QA testers' responsibility was to determine "whether the product that's been delivered matches the design documents" (quoted in Ruggill et al., 2017, loc. 4275).

Changes in software development philosophy have also led to changes in game design philosophy. When the Manifesto for Agile Software Development was published in 2001, it was a culmination of critique on traditional design that sought to replace careful planning with dynamic and organic collaboration, creation, and prototyping (Beck et al., 2001). One of the technical communicator participants in Zhang and Saari Kitalong's (2015) study found that she was more involved in the creative design processes with Agile development compared to her work "at the end of the product cycle" (p. 208) of waterfall development. No doubt, Agile and other iterative development processes such as Scrum (Takeuchi & Nonaka, 1986) or spiral (Boehm, 1988) have become popular, and books on game design have discouraged GDDs and waterfall design (Schell, 2008) in favor of quickly prototyping game ideas, playtesting the prototype with gaming audiences, and then refining the game further within an iterative, open-system (Spinuzzi & Zachry, 2000) and a reflexive design process (Sansone, 2014; Schell, 2008).

The change in design philosophy from waterfall to more Agile-inspired design processes was also enabled by a shift in how many corporate structures work, especially within areas of design and engineering. Spinuzzi (2015) argues that many corporations have moved from stable bureaucratic structures of controlled hierarchy with rigid rule-systems to a more informal, networked model or adhocracy in order to better innovate. Instead of being defined by set structures, the adhocracy loosely comes together in order to complete a single project and often regroups or disperses completely after project completion. Communication within an adhocracy is flattened and open to all so that the group can quickly adapt and reorganize according to the shifting demands of the project. Because adhocracies can be likened to swarm formations that can take any organizational form around completing a project, members can synthesize multiple design philosophies to meet their own ends, often creating hybrids as we show in our study later.

Other social forces have pushed game development beyond the corporate realm, leading to still other adhocracy formations, and the evolution of the GDD as more of a vision statement within game development. Beyond development, there is currently less need for large corporate backing and distribution of games. User-focused funding sites, such as Kickstarter, and distribution networks like Google Play, the Apple App Store, Steam, and itch.io, allow designers to market and distribute their games without the need for traditional retail space, even if these new systems often reach a smaller niche audience (Jenkins, Ford, & Green, 2013). As a result, there has been an impressive increase in the number of small and independent studios ("Indie Game Development," 2013), and with these small teams comes less of a need for detailed technical documentation.

However, while the GDD has largely shifted from exhaustive record of the waterfall method to a leaner vision statement, especially for smaller, independent design companies, the genre conventions the GDD utilizes still vary widely within game development (Greene & Palmer, 2014) and a more comprehensive GDD is still used within large design companies, especially if they are structured in traditionally corporate ways. For instance, Gamasutra has featured no less than four articles spanning the last decade and a half on writing a GDD (Freeman, 1997; Ryan, 1999; Bakker, 2009; Friesen, 2014; Gonzalez, 2016), with the most recent of these claiming "using a GDD is a thing of the past" while contradicting this statement by going on to describe how to write a GDD, if not a bit shorter than it has been traditionally defined (Gonzalez, 2016). Meanwhile, in 2013, then studio director of the large studio Bioware, Yanick Roy, posted an image of a three-inch binder to Twitter, writing, "First pass at Design Document for the next Mass Effect! #sorryforthetrees." Both of these examples illustrate that while the GDD persists within game development, the genre conventions can markedly differ depending on the demands of the design activity systems they mediate and the ad hoc form that the design team takes.

The multitude of genre permeations for the GDD is to be expected though because, as a meta-genre, it directs diverse and complex game design activity systems that also take on many adhocracy forms (Giltrow, 2002; Spinuzzi, 2015). In fact, Giltrow argues that sharp differences of the same genre form such as the GDD could mean that it is a site of "contest and domination" (p. 199), which may be the case as independent developers rush to inhabit spaces in the game market not filled by major developers, and certainly, a pared down GDD vision statement reflects differences in values and development that suggest flattened communication, loose collaborative structures, and creativity over tight structure and corporate control (Spinuzzi, 2015).

Furthermore, the lack of genre consistency for the GDD is a primary reason that game development would benefit from the expertise of technical communicators (Eyman, 2008; Mason, 2013), particularly as technical communicators understand how to analyze and write for multiple audiences with diverse purposes (Greene & Palmer, 2014). In fact, Eyman (2008) argues that because games follow a similar development process as other types of software, it would follow that technical communicators could facilitate communication within game development in similar ways, and both Eyman (2008) and Mason (2013) specifically mention documentation as one of those ways. Even more specifically, within their study of the GDD, Greene and Palmer (2014) discovered that within game development there is often a disconnect between "concept design and programming-implementation" (p. 24), a gap that the GDD is meant to fill but often fails. They argue that technical communicators could use their rhetorical skills to mediate communication between these two aspects of development.

Our study was also prompted because we saw the GDD as a varied and contested genre. Specifically, we wanted to understand how its various genre conventions influenced the activity of design and how design actively reflexively influenced its various genre conventions within Agile to more traditional work environments. Our research question then asks: how are game design decisions made, communicated, and made manifest in the games produced from small to medium-sized studios? Of particular interest are what genre functions of the GDD mediate current design decisions, and whether such documentation is responsive or dynamic in game design practice?

METHOD

In an IRB reviewed study, we conducted face-to-face interviews with a purposeful sample of four different sized game design studios in the Colorado area: Megan Fox from Glass Bottom Games, Kevin Zhang from Serenity Forge, Jordan Coombs from Warballoon, and John Whitmore from Backflip. Our intent in the research was to show practice in context and not to generalize (Patton, 2002), and our selection scaled studio size, from ostensibly a handful of developers to the much larger Backflip with dozens of artists, designers, and programmers.

With the consent of the participants to reveal their studios and names, we recorded and transcribed semi-structured, in-depth interviews. One author interviewed Zhang, Coombs, and Whitmore, and the other author interviewed Fox. We only were asked to sign a Non-disclosure Agreement (NDA) at one studio, Backflip, but it did not impact our interview. Our interview questions were in four categories: biographical, including past games and inspirations for games; description of game design processes; how game design processes were reflected in a GDD; and attitudes and beliefs about game design. We composed questions in these categories to capture context about participants' life-world (Kvale, 1983), their training, and game ideation experiences (similar to Hagen, 2009 and Kultima, 2010), and their processes (similar to Ruggill et al., 2017 and O'Donnell, 2014). We did not intend to present models of game design as much as we wanted to share what drove these developers to design and the methods they used to communicate that design (Engeström, 1999). While we did make sure to ask at least one question about GDDs, our interview stance remained flexible and intended to capture how game design was conducted and documented in these studios, including the implicit and explicit rhetorical exigencies that determined how design documentation was written or used.

We coded the interview transcripts using a two-cycle content analysis (Saldaña, 2009) approach. On the first cycle, we focused on looking at emergent patterns that developed from our research question (Mayring, 2000). In the second cycle, we compared participant response patterns within the context of their separate game design experiences and sought connections in how meta-genres operated in the workflow of each participant's process. Our conventional content analysis (Hsieh & Shannon, 2005) and comparative coding, in turn, preserves the diversity of the different experience. We did not intend for these four designers to be generalizable, so we did not norm for interrater reliability of our coding scheme. Instead, we coded patterns that emerged from the data to understand how design documentation and design context may influence design processes. We noted a reciprocal relationship, and we organized them under subheadings in the results and discussion section that follows. Thus, the patterns we coded were as follows:

1. *The GDD exists to ensure design consistency as a vision statement.* Here we included any statements that discussed how the GDD was meant to act as a vision document or to ensure consistency.
2. *The GDD exists within a reciprocal relationship to design.* Here we included any statements that discussed how either the GDD or the design evolved together and how they changed each other as a result.
3. *Designer skill sets influence the GDD.* Here we noted any mention of designer skills, paying specific attention to ways that these skill sets changed genre conventions within the GDD.
4. *Game genre differences influence what is included in the GDD.* Here we coded for any mention of how game genre differences changed GDD genre conventions.
5. *Design company size influences GDD genre conventions.* Here we paid attention to how size impacted communication and how the GDD mediated this communication or not.

Studies to date, as well as our own, confirm that game development is heterogeneous (Greene & Palmer, 2014; Ruggill et al., 2017) when studied in context and through the perspective of the developers and designers. Continuing this trend, we share these four perspectives to add to the literature of game design approaches so that technical communicators and game designers might better understanding the meta-genre functions of design documentation in practice.

RESULTS AND DISCUSSION

While neither generalizable nor exhaustive, our interviews revealed five noteworthy findings in the way design documentation was used at independent game studios. GDDs still directed the activity of design, but they did so in a reciprocal way, which is to say that design documentation changed along with design. We also found that design documentation functions and conventions were influenced by designer backgrounds, the game genre, and studio size.

The GDD as Meta-Genre

In our study, we discovered that as a meta-genre that directs the activity of design, the primary purpose of the GDD was to ensure consistency with the overall design vision of the game—a genre purpose that was consistent for all four participants of our study regardless of design studio size or the type of game that was being designed. Game developers, more than any other software developer, recognize that games are fundamentally more than a list of components and assets. If a game is not a fun experience for players, it is not worth playing. Consequently, the GDD as vision document communicates a cohesive experience for the player (Greene & Palmer, 2014; Hagen, 2009; 2010; Schell, 2008). This overall purpose of experiential vision for the GDD is echoed on Gamasutra by Ryan (1999, October 19), who wrote that "the purpose of design documentation is to express the vision for the game." Within our study, Whitmore, a veteran game developer, said about the GDD, "Once you've got the goals established, and you know what the experience is supposed to be at the end, you can test those builds against something that you've already kind of agreed on. In the absence of that kind of vision statement, you end up iterating in a lot of different ways."

As an example of how the GDD ensures design consistency, Fox described the game's themes in the beginning of the GDD, creating a genre convention unique to other GDDs, in order to ensure the rest of her game design—the plot, characters, and other elements—fit well within her game's vision. The prominent theme in Fox's game, *Hot Tin Roof*, is gender as the protagonist is female, but, although queerness is not included in Fox's vision, the protagonist is also a lesbian, making queer identity a major theme as well. She described her writing process for the GDD: "I get down to writing a character story for Suzie. Okay, is it about Suzie the straight girl? It doesn't work. Suzie is a lesbian. Just making sure that all of those fit." The rhetorical exigency of the GDD is to ensure consistency with the design vision, which should be clearly laid out in a descriptive overview within the GDD. This exigence ensures that the daily quotidian rhetoric of game development—the minutia of coding, scripting, or animating characters that designers are often too busy to think of in rhetorical terms—stays consistent with the overall vision.

As a meta-genre that controls the design consistency of the game, the GDD also mediates the expression of a whole ecology of other documents related to the game's design, even when other mediating genres take the place of some of its genre conventions (Spinuzzi & Zachry, 2000). As Giltrow (2002) has described, "meta-genres flourish at the boundaries, at the thresholds of communities of discourse, patrolling or controlling individuals' participation in the collective" (p. 203). For instance, a GDD historically has often had a short one to three paragraph introduction that pitches the vision of the game with a compact description. Ryan (1999, October 19) describes the purpose of the introduction like this:

> The introduction to your game concept contains what are probably the most important words in the document—these words will sell the document to the reader. . . Include the title, genre, direction, setting, edge, platform, and any other meaningful bits of information that cannot wait until the next sentence. The edge is what's going to set this game apart from the other games in the genre. (para. 17)

While the introduction still shares the coherent vision of the game with the rest of the design team, it also pitches the game vision to potential investors and distributors. However, while the other three game designers we interviewed did not mention omitting the introduction, Fox's GDD for *Hot Tin Roof* did. Instead of an introduction, Fox has a vividly descriptive and pithy game pitch memorized and also relies on a multimodal game trailer and a set of slides to communicate the game's vision when pitching to investors. If possible, she also has a short but playable game demo. However, her pitch still introduces the game's focus for *Hot Tin Roof*: "L.A. Noir meets Metroid with an Inspector Gadget gun." Consequently, even though the introduction in Fox's GDD has been largely replaced by other pre-existing genres such as the pitch and multimodal game demo, the purpose of the introduction in her GDD still directs and mediates these genres, even in its absence: visceral descriptions of the game that clearly communicate the game's focus.

The GDD Constitutes a Reciprocal Relationship to Design

While ensuring overall vision consistency throughout the design process, as a meta-genre, the GDD seems to have a reciprocal relationship with the design process as it articulates the activity of design (Christensen, Cootey, & Moeller, 2007; Russell & Yanez, 2002). For all four of our participants, the GDD directs the design as an overarching blueprint, but the design details of the document also evolve throughout the design process as new parts of the game are added or parts are deleted because they are not fun to play, are too difficult to design, or do not enforce the overall vision for the game. Coombs described the GDD as "a living document." He added that "a lot of things change as we go." Fox writes the GDD as she is designing the game, including any additions she programs but also taking anything out of the game or the GDD that conflict with the consistency of her plot. Fox, in fact, wrote her entire GDD for *Hot Tin Roof* as a list, which she frequently added and deleted from as she designed the game. While the GDD's purpose remains constant—that of ensuring plot and character consistency within the game's overall vision—the content and form the GDD takes changes on a contingent basis in order to meet the design needs of the game. Whitmore described the game design process as reflected within the GDD as follows: "What you start with is not what you end with. One of the things that's most difficult to get younger designers to understand is, if you do something the first time, it is never right. You will go back and do it over, and several times. If you don't like doing that, you really can't be a designer."

Designer Strengths Influence GDD Conventions

Because of the GDD's reciprocal relationship with design, we discovered within our study that the genre conventions were often constantly changing and contingent as they depend upon the differences in design processes that individual designers have, design processes which are often, in turn, created by the strengths that each individual designer brings to the process. For two of our participants, these design strengths also influenced the writing of the GDD as well. For instance, Fox has a degree in mathematics, so one of her strengths is developing algorithms for specific game mechanics. To begin *Hot Tin Roof*, she did not do much writing for the GDD beyond describing an initial game idea. Instead, she quickly prototyped the core mechanic of the game in a 48-hour game jam. Afterward, she developed content that included a complex plot-line, characters, and puzzles for her adventure game and recorded everything in list form within the GDD to ensure consistency. However, the game mechanics were not described in much detail because she relied on quickly prototyping them instead. In contrast, Coombs and his brother at Warballoon are graphic designers who first worked in web design and advertising before starting their game design company. Consequently, they approach the process of game design as visual designers, visualizing the look and feel of their games first and describing this in the GDD before prototyping. In fact, Coombs said, "I'd fashion it closer to filmmaking than I would to writing down core mechanics and things like that. It's more like, 'When is the setting? What does it feel like?'" Describing his initial game design process in more depth, he added that to begin a game, he uses "a combination of pen and paper [prototyping] with digital concept art. . . . It's like, 'Okay, here's how this works. Is this right?' Usually, you don't get to [digitally] prototype until you need something to go faster. . . [then] complex math problems start happening." In this way, each designer's design strengths create a specific design process which influences not only what genre conventions look like in the GDD but also the writing process of the GDD as well.

In fact, Coombs and Fox described GDDs that seemed fairly similar in that they only give basic descriptions of game mechanics, but this similarity is due to striking differences in each of their design strengths and resulting design processes. For instance, even though a GDD has been typified traditionally to include a game flow section that describes the gameplay in detail for each major portion of the game (Ryan, 1999, December 17), Coombs' GDD for *Star Command* did not include a game flow section, which is unsurprising as the game has dynamic encounters. Similarly, Fox did not have a game flow section either because she digitally prototypes algorithms and does not describe their flow in her GDD; however, she did use the GDD to describe her puzzles in detail and other play elements such as threats in the game from the perspective of a player.

Game Genre Differences Influence GDD Conventions

Because of the reciprocal relationship between the GDD and the design processes, differences in genre conventions across GDDs were due within our study to basic differences between the types of games being designed as each unique game demands a slightly different design process. Two interview participants discussed how game genre differences changed their design process and subsequent documentation for that design. For instance, while Fox came up with a working prototype for the core game mechanic of *Hot Tin Roof* during her 48-hour game jam, refining the game further took eight months, much longer than she had anticipated: "For *Hot Tin Roof*, I had to make the base game, including that 2-D, 3-D platforming tech. That took a while. . . . I couldn't let anyone play the game until all of that basic systems design was done. And it's a content-driven game, so I had to make content. I had to make levels for it." While she described her game's content in detail in the GDD, the fact that no one could play test the game for so long added to the game's technical woes. *Hot Tin Roof* blends the game genres of first-person melee action with that of a mystery-driven adventure game, creating a new game genre. Consequently, no one had the antecedent genre knowledge (Jamieson, 1975) needed in order to figure out how to play the game without a tutorial. As she herself admitted, this was "dangerous" as during this lengthy amount of time she also did not know if anyone would enjoy the game: "And I'm just counting on my own instincts of 'I think it's fun. The mechanics are fun. It will be fun once the content is done.'" Knowing that she had to dramatically change her design process for future games, working rapid prototyping into her process so that her game could be play tested throughout its design, she decided that her next game would consist of only one genre—a basic melee brawling game called *Spartan Fist*. She decided on this genre because she could quickly design the core fighting game mechanic—a process which was dramatically shortened at two months—and have a playable game that she could also keep refining by adding content to it. This change in design process also led to a shorter GDD. Because a melee brawler consists mostly of a core fighting mechanic, there is not as much content to describe as there is in a mystery-driven adventure game: there are no puzzles, the plot is dramatically shortened, and there are not as many characters.

For Serenity Forge, the game design process and resulting GDD also varied tremendously between games. Zhang described their process similar to testing a hypothesis. Consequently, workflow for each game differed because it consisted of a different set of hypotheses tested differently each time. This varied design process also meant that any documentation, including a GDD, could vary tremendously as well. Zhang stated that "it's been a lot more about forming a hypothesis essentially and then testing it out and realizing it works or it doesn't work, then continuing to do that over and over again for all aspects of game design and game design documentation." Because of Serenity Forge's hypothesis-driven design philosophy, Zhang went on to further explain that "our design process is constantly in flux in terms of experimenting with new types on processes and seeing what works and seeing what doesn't." In fact, he speculated that the company's design process could change so much in the future that they could throw out design documentation entirely if doing so suited the type of game they were designing.

Game Studio Size Influences GDD Conventions

Finally, because as a meta-genre, the GDD directs the activity system of game design, within our study, many genre convention differences within the GDD were also due to the size of that activity system (Russell & Yanez, 2002), which also influenced the type of adhocracy and requisite design philosophies that formed to meet design needs. For the three small, independent game design companies within our study, Glass Bottom Games, Warballoon, and Serenity Forge, this activity system often only included a handful of designers who did coding and artwork themselves. These three small companies tended to follow aspects of Agile philosophy, using pared down or bits of a GDD to describe, often from the perspective of players, the visceral experiences of the game, while the specific technical details of programming such as setting up algorithms, or even creating specific artwork, were left up to the individual creativity of those completing those tasks and were not really included in much detail in the GDD (Beck et al., 2001). However, this Agile-inspired process was dynamic and evolved to meet changing design needs. For instance, there was no need for a formal Scrum design system with a backlog, Scrum master, or Product Owner because members were already in constant communication with each other and often just did what was needed to complete the game (Singhal & Gupta, 2014).

One specific aspect of the GDD where design company size mattered within our study was recording evolving game design changes within the GDD. Even though the GDD exists within a reciprocal relationship to the design process, small design companies did not always record game design changes within the GDD because it was easier for everyone to discuss these changes in person. For instance, Zhang stressed the importance of documenting large changes but downplayed the importance of documenting small changes:

> It's important that everyone's on the same page. I would say [changes to the GDD are] a case by case basis. If the change is so big that it's like oh here's a fundamental mechanic that we're going to change to something else, then it's worth going back and actually jotting it down on the design document. If it's a series of small iterations and we know that even if we jot down these iterations this week by the same time next week they're going to change again, then usually we feel like it's not really worth writing it down.

In fact, Coombs downplayed the importance of updating a GDD even for big changes because it was easy to quickly communicate the changes with his small team, illustrating how meta-genres consist of the attitudes surrounding an activity—an activity which can be implicitly or explicitly stated depending on the nature of these surrounding attitudes (Giltrow, 2002). However, Coombs still documented changes in more ad hoc documents, which the GDD still mediated as the controlling meta-genre (Spinuzzi & Zachry, 2000). Coombs discussed with the team about games' mechanics, iterating both internally and externally, so that the documentation manifested across more than a single GDD: "We're documenting all this, but white boarding it, some Google docs a little bit, a little bit of everything, some notes and stuff like that." The process is messy, and Coombs conceded that, at least in his experience in advertising and in talking to other small studio game developers, "I think most companies do work like this until you get to AAA where it's like, once you have an investor, stakeholders and things like that."

In contrast, large game design companies have diverse teams and usually need the GDD to direct and coordinate many groups, documenting extensively as a result. Within our study, Whitmore, a veteran game designer of 20 years, described how specific genre conventions within a GDD directed the activity system of a large group of designers for a AAA game created within a formal bureaucratic corporate structure:

> Documents became very large. When I was working on something like *Metroid Prime*, we had about a thousand pages of documentation, including one of the level docs. . . . We started to break the document down into a bunch of sub-documents that laid out the systems. Whoever was responsible for coding or creating those systems would have one document they'd look at, and they wouldn't need to worry about the other documents.

For teams of this size, Whitmore explained that the GDD has three different audiences: marketers who need a product description written from the perspective of players early on in the design process, programmers who need to specifically know what to code, and visual artists who need to know what to draw. Consequently, the GDD also had specific sections written for these specific audiences, but each audience would only look at the subsection designated for them and would not look at the rest of the document.

As the one participant in our study who was not part of a small, independent game design studio, Whitmore works in a mid-size studio with 20 to 50 developers at Backflip. Although he tries to use some Agile philosophy in design, he also explained that this is not always possible. The process he described sounded more like an adhocracy, but one that still uses some of the formal organizational structures of a corporate bureaucracy. For instance, he has the formal title and corporate role of Director of Design, although in adhocracy-style, he attempts to be in open communication with his team, avoiding more formalized, bureaucratic chains of communication. As such, he described preferring a fairly detailed GDD to productively direct the activity of his larger team: "We need a comprehensive design document that starts with a kind of overview that lays out the goals of each system, what they're supposed to accomplish in the context of the game, and then breaks that down into very specific rule sets, asset lists, and so on." However, within the adhocracy formation of Backflip, Whitmore has adapted some design philosophies to meet his design needs while abandoning others. For instance, he still favors some Agile-inspired design processes such as a prototyping, playtesting, and redesign. Although he favors full and constant documentation, this documentation exists on a dynamic wiki that can change alongside design changes, reflecting more of an Agile-inspired design process.

Furthermore, as Backflip exists within a type of adhocracy form, Whitmore explained that the GDD was necessary to enable open communication with a large team, communicating even small changes in game design as the game progressed, especially as there is still a reciprocal design relationship between the design process and what is written in the GDD. The GDD communicates a design vision to the artists and programmers but also creates a way for the programmers and artists to communicate their design process in order to shape the GDD.

> There's a lot of spreadsheets there and a lot of rule sets to be written. It's generally fairly dry. It's meant to be read by the programmers and the artists, so that they can kind of figure out what they need to build. That gets reviewed as well, and we'll sit down with the programmers and say, "Hey, is this possible? Can we do this?" Go to the artists and say, "Hey, can we really generate the assets? Do you know how you're going to do this?"

In this way, the programmers and artists still designed with a reciprocal relationship to the GDD. If artists could not create assets for part of the game, they could suggest workarounds that could then contribute to changing the visual aesthetic of the game, which would also lead to changes in the GDD. Similarly for the programmers, if parts of the design did not work well when coded and play tested, they could make suggestions for the design that affected this part of the code, changing that part of the GDD along with it. Even in a larger company then, the GDD articulates the quotidian parts of the design process and helps prevent larger potential exigent crises such as art that is not well rendered or game-ending bugs. Even in larger companies, the GDD both acts upon and is acted on by the designers and artists, existing within a reciprocal relationship to the design process.

CONCLUSIONS

The expectation of technical communicators new to the game development industry might be that rapid software development using Scrum/Agile practice and user-centered iterative design have largely replaced documentation practices like the GDD (Beck et al., 2001; Sansone, 2014). However, our study suggests that the GDD as meta-genre still plays an important role in game development by becoming a vision statement, which in turn forms a reciprocal relationship with design (Christensen, Cootey & Moeller, 2007). Consequently, as a meta-genre that both shapes and is shaped by the activity of design, the genre function of the GDD within our study varied depending on the size of the project, the type of project, the number of people involved, and the experiences and design strengths of the developer as well as the shape of adhocracy the design studio takes (Spinuzzi, 2015). As a result, similar to the findings of Ruggill et al. (2017), McDaniel and Daer (2016), Greene and Palmer (2014), and Hagen (2010), we found no standard documentation practice but a dynamic activity system that borrows bits and pieces of a traditional GDD along with propriety, specialist, and game-focused design documentation. Sansone (2014) has argued that GDDs were no longer used and that, "Documentation…exists, but it is developed in small chunks and in response to iterative needs" (p. 120). However, our interviews suggest a revision of this claim—in fact, some studios, especially studios that are working on larger projects with external IP, still need GDDs that are more than a "concise expressions of theme, tone, and vision communicated simply and emotionally" (p. 120). Furthermore, even some of these concise expressions and small chunks are abandoned in favor of pure prototyping, as we saw in Fox's 48-hour game jam prototype, and potentially in the future at Serenity Forge. Within our study at least, the GDD in its varied forms is a genre mediating a dynamic activity system (Schryer, 2002) that borrows bits and pieces of a traditionally typified GDD but also makes revisions to these genre conventions that are more in line with design strengths of designers and differences in game genres. This finding strongly supports the argument that technical communicators would benefit from understanding the genre functions of the GDD as well as its malleability as a meta-genre, as is evident from the various manifestations design documentation took in the independent studios we describe here.

The role of the technical communicator in these situations is multilayered. For one, as we saw from Coombs at Warballoon, the design process is dynamic, but there is still need of documentation. With smaller studios, having one person in charge of just design documentation is not feasible, but determining processes and systems of common documentation that technical communicators can impart to studios through consultation is viable, a theme echoed in recommendations from Greene and Palmer (2014). Beyond consultation, it is apparent that we should be aware of the practice of game design documentation. While none of our developers used what might be considered a traditional GDD, they all used parts, whether only a vision statement, a list of game features, or a full game encyclopedia complete with art and algorithmic specifications. Understanding the genre, not to dutifully fill in the blanks but to see when a feature is rhetorically effective for a given situation and needed to fulfill a purpose in the design process, is important for technical communicators.

Our study shows technical communicators how the meta-genre can potentially both influence and be influenced by the activity it is directing. Giltrow (2002) argues that the meta-genre is important to study as it often clearly shows how the context and purposes of a rhetorical situation shapes its genre conventions—as well as potentially some of the genre conventions of the genres within its activity system that it mediates (Bazerman, 1994; Russell & Yanez, 2002). Within our study, the GDD existed in a reflexive relationship with design, directing the overall vision for design, but also evolving along with any design changes (Christensen, Cootey, & Moeller, 2007). In this way, the design documentation described within our study shows how the uptake or discarding of specific genre conventions can be created by needs within the design process, existing within a reciprocal relationship to the design activity.

Our study also shows how meta-genres can potentially operate in patrolling the boundaries of genre systems within a particular activity system, and, as a result, direct the uptake of other genres, specifically within genre ecologies, mediating to a certain degree what genre conventions are used within them and how they are used (Spinuzzi & Zachry, 2000). The genre functions within the meta-genre also change as a result to the changes in the surrounding genres. For example, within our study, while traditionally a GDD has opened with a vividly written, compact game introduction that acts as a game pitch for the project (Ryan, 1999, October 19), Fox did not include one in her GDD because she expressed the introduction in the form of a series of multimodal slides, a game trailer, and a memorized one-sentence description. These pre-existing, alternative genres formed a genre ecology mediating the purpose of the introduction within the GDD, even if the introduction was omitted.

Furthermore, our interviews illustrate to technical communicators how genre conventions within the meta-genre can potentially change on an emergently contingent, ad hoc basis depending on the needs of the activity system. Specifically, our study illustrates that genre conventions within the GDD can change depending on differing strengths of the development team, the type of game genre under development, and the size of the development team, especially as it is part of an open design system (Spinuzzi & Zachry, 2000). For instance, because Whitmore was in charge of a large development team, he favored a detailed GDD, including even small changes that did not fundamentally alter the team's vision for the game in a dynamic wiki. He thought noting even small changes to the document was necessary to constantly stay in communication with the entire team and avoid potential bugs. However, Coombs downplayed the importance of careful documentation in the GDD because he was in constant direct communication with his small design group. However, he still noted design changes using other types of media and genres, which existed within an evolving and dynamic genre ecology.

Finally, as a meta-genre, our study illustrates to technical communicators that the genre conventions that the GDD uses can depend on the size of the game design studio and the type of adhocracy that the game development team has formed. For instance, the designers in Warballoon, Serenity Forge, and Glass Bottom Games all favored lightly documented GDDs that served more as vision guides (Schell, 2008). However, they were in small studios that only consisted of a few people, so they could easily stay in constant, direct communication. As Whitmore put it, "If everyone on the team knows where the designer is sitting, and you can go and talk to [him or her] whenever you run into a future question, [light documentation] works pretty easy." In contrast, Whitmore worked in a medium-sized company. The adhocracy that his design studio had formed consisted of still including formal roles such as director of design but was pushing for open communication within the game development team (Spinuzzi, 2015). As such, Whitmore preferred documenting game design changes within the GDD, but using a dynamic wiki, so that his entire design team of developers could stay informed.

Clearly, additional study is warranted. While there are some similarities across our developer interviews with each other and previous studies, we still maintain that game design is a heterogeneous activity. We should continue to study changes in design documentation and processes, including large studios. Both Coombs and Whitmore shared expectations and experience in large studios, but these are just two cases. However, we see hints at changes in large studios. For example, Blizzard Entertainment, developer of *World of Warcraft*, *Diablo*, and *Starcraft*, has been around for well over two decades, and their release schedule has moved at a glacial pace, suggesting a complicated design process with multiple layers of design documentation. However, when "Team 5," a small internal group at Blizzard was asked to put together something new, they created the now popular and quickly iterated *Hearthstone* that started with "literal pieces of paper. . . There were no pipelines to push assets through, no politics, no nonsense: just two guys in a room with a pen and paper and a handful of crazy ideas that might just work" (Serrels, 2014). Understanding how documentation has evolved within larger studios as a primary focus of study rather than the day-to-day operations and attitudes might seem less intrusive to studios with many corporate layers and Non-Disclosure Agreements.

Of course, studying design documentation artifacts becomes a worthy aspiration. No study to date has had access to design documentation during game development; Sansone (2014) looked at older GDDs easily available on the Internet. While some informants in our study were willing to show these documents, we were not allowed to share any part of them in this article. Given the smaller barriers, independent studios offer the most promise to analyze documentation artifacts, but as our studies show, such studios also are less likely to have extensive documentation on hand.

While we are resistant to generalize, we can say with confidence that documentation in game design is heterogeneous. This is not surprising given our understanding of communicative and rhetorical activity systems. Yet, we also saw across our interviews that developers with a few too many years of experience, trained in game design or not, or designing multiple games or only a few, all knew what a GDD was—not a single one asked us to clarify what we meant or what we expected such a document to be. The practice of documenting and communicating game design still exists, sometimes as a vision statement but also as an explicit and maintained document. The GDD is not dead—at least not yet.

REFERENCES

Bakker, J. (2009, June 4). A GDD template for the indie developer. *Gamasutra: The art and business of making games.* Retrieved from http://www.gamasutra.com/blogs/JasonBakker/20090604/84211/

Bazerman, C. (1994). Systems of genres and the enactment of social intentions. In A. Freedman & P. Medway (Eds.), *Genre and the new rhetoric* (pp. 79–101). Melbourne: Taylor & Francis.

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., . . . Thomas, D. (2001). Manifesto for Agile software development. Agile Alliance. Retrieved from http://agilemanifesto.org/

Boehm, B. W. (1988). A spiral model of software development and enhancement. *Computer, 21*(5), 61–72.

Christensen, D. M., Cootey, J. L., & Moeller, R. M. (2007, October). Playing in genre fields: A play theory perspective on genre. In *Proceedings of the 25th annual ACM International Conference on Design of Communication* (pp. 1–8). El Paso, TX. New York, NY: ACM.

DeAnda, M., & Kocurek, C. (2016). Game design as technical communication: Articulating game design through textbooks. *Technical Communication Quarterly, 25*(3), 202–210.

Engeström, Y. (1999). Expansive visibilization of work: An activity-theoretical perspective. *Computer Supported Cooperative Work, 8*(1–2), 63–93.

Eyman, D. (2008). Computer gaming and technical communication. *Technical Communication, 55*(3), 242–250.

Freeman, T. (1997, Sep. 12). Creating a great design document. *Gamasutra: The art & business of making games.* Retrieved from https://www.gamasutra.com/view/feature/131632/

Friesen, A. (2014, Oct. 6). Writing a GDD your team can actually use. *Gamasutra: The art and business of making games.* Retrieved from http://www.gamasutra.com/blogs/AbbyFriesen/20141006/227048/

Greene, J., & Palmer, L. (2014). It's all fun and games until someone pulls out a manual: Finding a role for technical communicators in the game industry. In J. deWinter & R. Moeller (Eds.), *Computer games and technical communication: Critical methods and applications at the intersection* (pp. 17–33). Burlington, VT: Ashgate.

Giltrow, J. (2002). Meta-genre. In R. Coe, L. Lingard, & T. Teslenko (Eds.), *The rhetoric and ideology of genre* (pp. 187–205). Cresskill, NJ: Hampton Press.

Gonzalez, L. (2016, July 26). How to write a game design document. *Gamasutra: The art and business of making games.* Retrieved from http://www.gamasutra.com/blogs/LeandroGonzalez/20160726/277928/

Hagen, U. (2009). Where do game design ideas come from? Innovation and recycling in games developed in Sweden. *Breaking New Ground: Innovation in Games, Play, Practice and Theory. Proceedings of DiGRA 2009.*

Hagen, U. (2010). Designing for player experience. *DiGRA Nordic '10: Proceedings of the 2010 International DiGRA Nordic Conference: Experiencing Games: Games, Play, and Players.*

Hsieh, H. F., & Shannon, S. E. (2005). Three approaches to qualitative content analysis. *Qualitative Health Research, 15*(9), 1277–1288.

Indie game development on the rise in a big way. (2013, Feb. 28). *Gamasutra.* Retrieved from http://www.gamasutra.com/view/news/187540/

Jamieson, K. (1975). Antecedent genre as rhetorical constraint. *Quarterly Journal of Speech, 61*(4), 406–415.

Jenkins, H., Ford, S., & Green, J. (2013). *Spreadable media:*

Creating value and meaning in a networked culture. New York, NY: New York University Press.

Kaufer, D. S., & Butler, B. S. (1996). *Rhetoric and the arts of design.* Mahwah, NJ: Lawrence Erlbaum.

Kultima, A. (2010, May). The organic nature of game ideation: Game ideas arise from solitude and mature by bouncing. In *Proceedings of the International Academic Conference on the Future of Game Design and Technology* (pp. 33–39). ACM.

Kvale, S. (1983). The qualitative research interview: A phenomenological and a hermeneutical mode of understanding. *Journal of Phenomenological Psychology, 14*(2), 171.

Mason, J. (2013). Video games as technical communication ecology. *Technical Communication Quarterly, 22*(3), 219–236.

Mayring, P. (2000, June). Qualitative content analysis. *Forum: Qualitative Social Research, 1*(2). Retrieved from http://www.qualitative-research.net/index.php/fqs/article/view/1089/2386

McDaniel, R., & Daer, A. (2016). Developer discourse: Exploring technical communication practices within video game development. *Technical Communication Quarterly, 25*(3), 155–166.

Meir, S. (Speaker). (2016, Oct. 25). "Episode 23: Sid Meir – part 1" Designer Notes [Audio podcast]. Retrieved from https://www.idlethumbs.net/designernotes/

O'Donnell, C. (2014). *Developer's dilemma: The secret world of videogame creators.* MIT press.

Patton, M. Q. (2002). *Qualitative research & evaluation methods* (3rd ed.). Thousand Oaks, CA: SAGE Publications.

Roy, Yannick. [YanickRRoy]. (2013, Jun 18). First pass at design document for the next Mass Effect! #sorryforthetrees [tweet]. Retrieved from https://twitter.com/yanickrroy/status/347046536934486016?lang=en

Ruggill, J. E., McAllister, K. S., Nichols, R. K., & Kaufman, R. (2017). *Inside the video game industry: Game developers talk about the business of play.* New York: Taylor & Francis.

Russell, D., & Yanez, A. (2002). "Big picture people rarely become historians": Genre systems and the contradictions of general education. In C. Bazerman & D. Russell (Eds.), *Writing selves/writing societies: Research from activity perspectives.* Fort Collins, CO: WAC Clearinghouse and Mind, Culture and Activity.

Ryan, T. (1999, Oct. 19). The anatomy of a game design document, part 1: Documentation guidelines for the game concept and proposal. *Gamasutra: The art and business of making games.* Retrieved July 17, 2016, from http://www.gamasutra.com/view/feature/3384/

Ryan, T. (1999, Dec. 17). The anatomy of a game design document, part 2: Documentation guidelines for functional and technical specifications. *Gamasutra: The art and business of making games.* Retrieved July 17, 2016, from https://www.gamasutra.com/view/feature/131818/the_anatomy_of_a_design_document_.php

Saldaña, J. (2009). *The coding manual for qualitative researchers.* London: SAGE Publications Ltd.

Sansone, A. (2014). Game design document: Changing documents, changing demands. In J. deWinter & R. Moeller (Eds.), *Computer games and technical communication: Critical methods and applications at the intersection* (pp. 109–124). Burlington, VT: Ashgate.

Schell, J. (2008). *The art of game design: A book of lenses.* Burlington, MA: Morgan Kaufmann.

Scholder, A., & Zimmerman, E. (2003). *Replay: Game design and game culture (New Literacies and Digital Epistemologies, V. 18).* New York, NY: Peter Lang Publishing.

Schryer, C. F. (2002). Genre and power: A chronotopic analysis. In R. Coe, L. Lingard, & T. Teslenko (Eds.), *The rhetoric and ideology of genre* (pp. 187–205). Cresskill, NJ: Hampton Press.

Serrels, M. (2014, May 4). Pens, papers, and envelopes: The making of *Hearthstone*. Kotaku. Retrieved from http://kotaku.com/pens-paper-and-envelopes-the-making-of-hearthstone-1571690185

Singhal, A., & Gupta, D. (2014). Scrum: An Agile method. *International Journal of Engineering, Technology, Management, and Applied Science, 2*(6), 182–190.

Spinuzzi, C., & Zachry, M. (2000). Genre ecologies: An open-system approach to understanding and constructing documentation. *ACM Journal of Computer Documentation, 24*(3), 169–181.

Spinuzzi, C. (2004). Describing assemblages: Genre sets, systems, repertoires, and ecologies. *Computer Writing and Research Lab,* 1–9.

Spinuzzi, C. (2015). *All edge: Inside the new workplace networks.* Chicago, IL: The University of Chicago Press.

Swales, J. (1990). *Genre analysis: English in academic and research settings.* Boston, MA: Cambridge University Press.

Takeuchi, H., & Nonaka, I. (1986, January). The new product development game. *Harvard Business Review.* Retrieved from https://hbr.org/1986/01/the-new-new-product-development-game

Walinsky, D. (2016, Nov. 4). The videogame industry's invisible workforce: Part 2. *Medium.* Retrieved from https://medium.com/@davidwolinsky/the-videogame-industrys-invisible-workforce-part-2-8365ecd19940

Zhang, Y., & Saari Kitalong, K. (2015). Influences on creativity in technical communication: Invention, motivation, and constraints. *Technical Communication Quarterly, 24*(3), 199–216.

APPENDIX

Tim Ryan's (1999, October 19; December 17) Typification of the Game Design Document (GDD)

Game Concept:

Introduction: Explains what the game is about in one exciting sentence.

Background (optional): Describes any products, properties, licenses, etc. covered in the introduction

Description: Describes the game from the player's perspective in one to three paragraphs; uses second voice

Key features: Lists game features that make the game different from other games.

Genre: Describes the game genre using pre-existing games as a reference.

Platform(s): Lists the platforms the game will be played on.

Concept art (optional)

Functional Specifications:

Game Mechanics: Describes "the game play in detailed terms, starting with the vision of the core game play, followed by the game flow, which traces the player activity in a typical game. The rest is all the infinite details" (para. 16).

Core game play

Game flow

Characters/Units (if applicable)

Game physics and statistics

Artificial intelligence (if applicable)

Multiplayer (if applicable)

User Interface: "[S]tarts with a flowchart of the screen and window navigation, then breaks down the functional requirements of all the screens and windows" (para. 26).

Flowchart

Functional requirements

Mockup

GUI objects

Art and Video: Lists all art and video assets in the game.

Overall goals

2D art and animation (if applicable)

3D art and animation (if applicable)

Cinematics

Video

Sound and Music: Lists all sound effect and music assets in the game. The overall goals describe the themes and mood the music evokes.

Overall goals

Sound effects

Music

Story (if applicable): Describes the game's story, including backstory and character descriptions.

Level Requirements: Lists level requirements. The level diagram is a chart of all game levels. The asset revelation schedule charts when assets will be revealed to players within each level. Level design seeds includes any paper sketches for level design.

Level diagram

Asset revelation schedule

Level design seeds

ABOUT THE AUTHORS

Richard Colby is a Teaching Professor and Assistant Director of Writing at the University of Denver. He co-edited the collection *Rhetoric/Composition/Play through Video Games*, published by Palgrave Macmillan, and has published numerous articles about games and teaching.

Rebekah Shultz Colby is a University of Denver Teaching Professor. She co-edited the collection *Rhetoric/Composition/Play through Video Games*, published by Palgrave Macmillan, and has published numerous articles analyzing games rhetorically or exploring them pedagogically. She recently published an article in *Computers and Composition* about game-based pedagogy in the writing classroom.