Exploring Game Industry Data Using Temporal Visualization

Cyril Focht, Isaac Karth, and Max Kreminski

Department of Computational Media

University of California Santa Cruz

ABSTRACT

In recent years there has been an increasing focus on understanding game development communities and how practices of the industry affect game developers. In this paper we expand on this work by using visualization methods to explore community and occupational trends in the video game development industry over time. For our analysis we use data from the Mobygames database, which contains data on over 100,000 games [1]. We discuss general trends in employment and interesting anomalies in the data.

1 INTRODUCTION

There has been a great deal of concern in recent years about burnout in the game development industry, and we're starting to see more research into the game development community in general [2, 4]. The contribution of this paper is to present visualizations of changes in the game development community. We approach this by using an egocentric collaboration graph to examine collaborations and community structure, multi-line graphs to visualize the career span of a random sampling of developers, and a career length lighting chart to compare the career paths of each year's cohort in a random subsample of the dataset.

In the community structure visualization we will be looking at changes in collaboration over time and influence of individual developers on sub-communities. This visualization depicts the movement of individual developers from game project to game project over time, illustrating trends in the persistence of groups of collaborating developers. Do developers frequently work with other developers with whom they previously worked, as might be expected in the case of (for instance) subsequent releases from the same game development studio? How long, on average, does any given developer remain a member of a single community within the larger graph? Do community structures within the graph ever split into smaller communities and later rejoin into larger ones again? This visualization aims to answer questions of this nature.

We use three visualizations to examine employment trends. The first demonstrates how long developers are staying in industry based on starting years. The second shows how long developers in different roles are staying in industry over time. The third examines collaborations between developers and traces what parts of the industry developers worked in both before and after a particular game.

Our data was collected from the Mobygames database, which contains data entries on games, which include descriptors, such as release date, general descriptions, reviews, images, contributing studios, and a list of credits. Any information pertaining to individual game developers referenced or used in this paper was pulled from that developer's appearance in games credits.

2 RELATED WORKS

Our goal in this paper is to apply visualization methods to the analysis of data pertaining to the
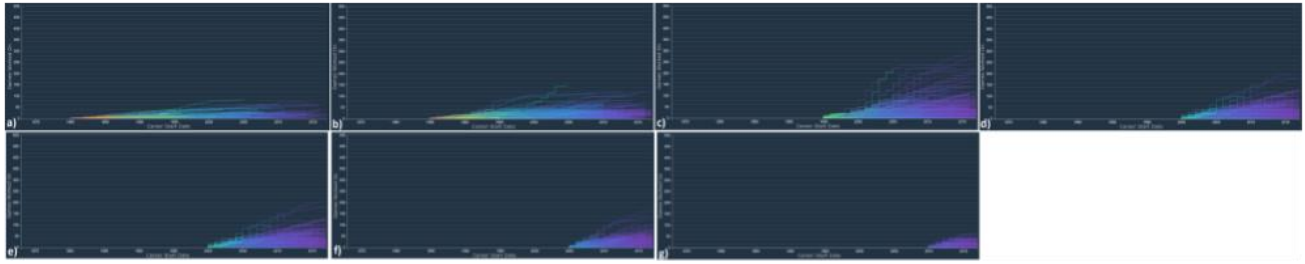
Figure 1. Career span of developers by year of first credited work; (a) 1980-1984; (b) 1985-1989; (c) 1990-1994; (d) 1995-1999; (e) 2000-2004; (f) 2005-2010; (g) 2010-present

game development industry, enabling us to look at general industry trends over time. Since 2014 the International Game Developers Association (IGDA) has published reports of annual satisfaction surveys among individuals in the game development community [2]. Legault and Weststar [3] used the results of IGDA's 2014 satisfaction survey results, combined with two previous quality of life surveys conducted by IGDA in 2004 and 2009, to analyze change in working conditions over a decade. Both of these reports highlight industry issues developers are facing, especially the issue of crunch. While the IGDA reports focus on the experiences of individual developers, Weststar [4] instead delves into understanding the community structure of game developers, investigating both the community as a whole and what effects its fluid organization has on individual developers. In his dissertation Schmalz [6] analyzes innovation in the games industry. In several chapters he analyzes both the characteristics of the industry as a whole and the effects of these characteristics on innovation, with one chapter focused primarily on the roles played by individual developers.

We apply and build on approaches taken by others to similar data. To understand fluidity in community structures, Vehlow et al. [8] introduce a variation of a Sankey diagram that highlights the movement of individual nodes between communities identified in a larger graph. Kobourov and Yee [8] introduce a method for representing time-series network graphs by animating time-slices of a 3-dimensional representation. Herr et al. [9] explore methods to increase the readability of highly complex network

visualizations using data from the Internet Movie Database, a similar dataset similar to the one we analyze in this paper.

Applying visualization methods to games themselves, the products of the game industry, is of particular interest to scholars in the field of game studies. Ryan et al. have made a series of contributions to this space. By embedding textual descriptions of games in a vector space, GameNet [10] enables quantification of the relatedness of games to one another. This, in turn, enables the development of interactive visualizations of this vector space, including GameGlobs, GameTree [11], and GameSpace [12]. Summerville et al. also make use of visualization methods to analyze jump mechanics in games [13].

3 METHODS

Due to both the size of the dataset with which we were working and the fact that we wanted to answer several markedly different questions with the final visualization, we chose to break our visualization down into several discrete tabs rather than attempt to render all the relevant information within a single view of the data.

Broadly speaking, the questions we attempt to answer with our visualization can be divided into two high-level categories: questions pertaining to community structure within the larger graph (especially questions of how individual developers move between communities over time), and questions pertaining to trends in employment of individual developers within the industry over time (especially questions of how long individual developers tend to remain in the industry, and the potential causes of any variation that occurs here).

The careers of all the developers credited on the game Aquaria

Game Release Date
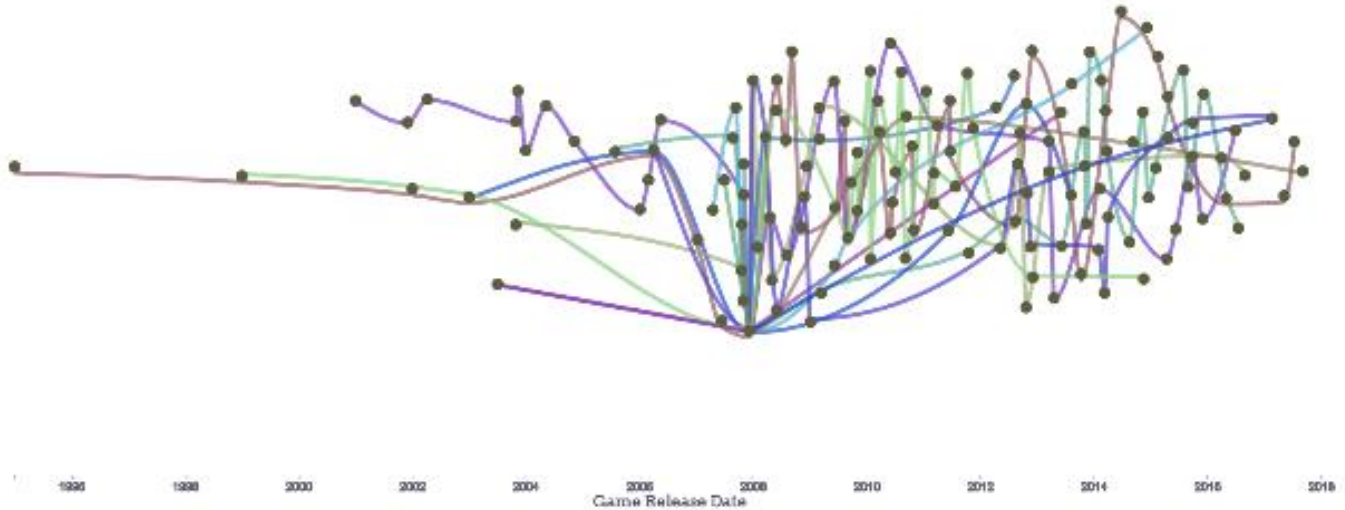
3.1 Community Structure

Due to the project-based nature of video game development, project teams are frequently fluid in structure. Developers are often moved to new teams within the same studio—or laid off, and move to a new studio—upon project completion [4]. We are investigating whether developers tend to make these moves as individuals, or if groups of developers tend to stick together when moving between projects or between studios.

An initial attempt at visualizing community structures applied methods with a Sankey diagram, as used by Vehlow et al. [7] to show amount of movement to and from communities and how that movement is grouped. However, the performance problems posed by attempting to construct a visualization of this nature in the web browser led us to pursue an alternative path.

To visualize community structure within the larger graph, we use an egocentric force-directed node-link visualization that illustrates the movement of individual developers (edges) between game projects (nodes) in the graph over time.

The methods we chose to use for this visualization are not capable of rendering a visualization of the entire dataset in the web browser, as the dataset as a whole contains too many individual nodes and edges to render successfully. As such, we adopt an egocentric visualization method in part as a way of filtering the dataset down to a manageable size. To select this filtered dataset, we begin by picking an initial "seed" game project on which to focus; for demonstration purposes, we chose the indie game Aquaria as our seed. Then, we examine the careers of every developer who worked on this game and bring any other games on which any of these developers worked into the set of games to visualize. Finally, we construct a visualization showing the movement of developers between all of the games in this set.

3.2 Employment

Burnout is a hotly discussed topic in the game industry, and people both inside and outside the industry are becoming increasingly concerned for the wellbeing of individuals within the development community [2, 14] due to exploitative business practices and other toxic cultural norms [15]. There is an abundant amount of work being done to mitigate burnout among game developers, but how effective has it been?

To approach this question we look at the length of time developers have spent working in the game industry and attempt to identify the years during which new developers who first joined the industry were most likely to remain in the industry. We use a random sampling of credits from the Mobygames dataset to form two multi-line time-series graphs to plot the number of games developers are credited with over time. Due to the high density of data displayed in this method, we applied varying levels of opacity to each line to increase readability and achieve a slight heat-map effect so that varying levels of clustering can be more easily distinguished.

First, we look at figure 1, which shows the length of time developers are staying in industry by the year they initially entered the industry. We can see that most developers who entered the industry between 1980-84 began retiring between 2005-09, and continued to 2015. Developers left the industry intermittently from 1995 onward, but the majority retired after 2010. In 1985-89 we see a similar pattern. Developers started to retire as the year approached 2015, with most leaving in 2016, as well as an increase in developers who worked on a large number of projects. In 1995-99 we see a large spike in developers who have contributed to large numbers of games. We don't see very many developers from these years having started to retire yet from here on, which suggests that the average developer stays in the industry for about 30 years before retiring. From 2000 onward we see the same trends continue. Not many people appear to be leaving the industry, and the same patterns of high-productivity developers are emerging.

Next, we look at figure 2. In most of these graphs we can see a considerable amount of clustering of developers who have contributed to fewer than fifty games between 1995 and 2010. Looking at a) we can see that employment picked up between 1985 and 1990 and continued to grow after that. Looking at b) we see that between 1980 and 1985 the industry consisted mostly of programmers. Between these graphs we see that there was an increase in employment between 1985-90 and another between 1995-00. Looking at d) we see that management positions started becoming more popular between 1990-95 and became wide-spread between 2000-05. Management positions also see many developers who are credited on large numbers of games, which one might expect since they often oversee multiple projects at a time. Programmers, designers, and artists have seen similar employment trends, with a few productive people between 1985-90, shifting to many developers contributing to fewer projects after 1995. Audio production saw a similar shift, but much more pronounced, having a few extremely prolific developers until 1995, and shifting to a larger number of employees contributing to fewer projects than previously.

4 DISCUSSION

In general, many industry trends – especially those pertaining to employment – are difficult to make out in these visualizations. The significant overall growth of the industry since the 1980s often masks smaller or lower-level trends in the employment of individual developers. Possibly as a result of this masking effect, there do not appear to be any significant drops in employment over time. If burnout does have an effect on the industry, it is possible that the effect of burnout has remained
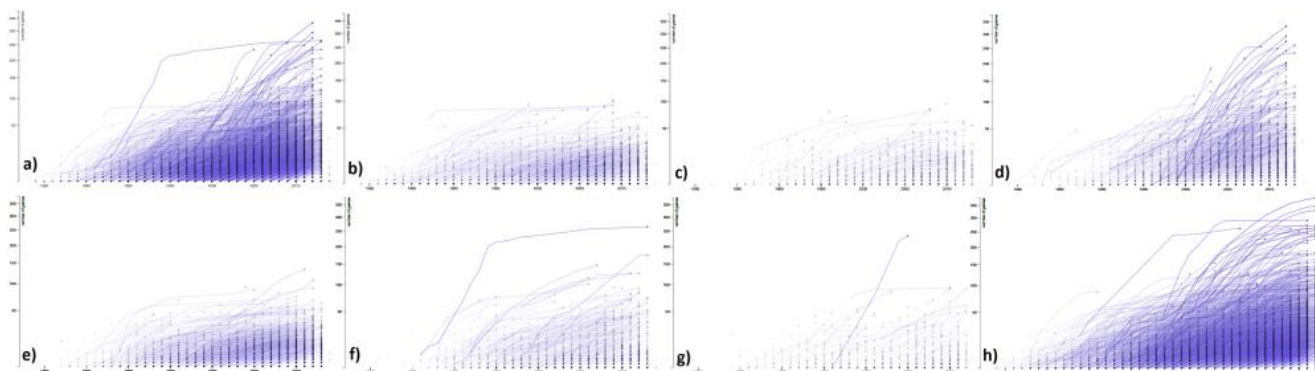

Figure 2. Career span of developers, arranged by role; (a) overlay of b, c, d, e, f, and g; (b) programming; (c) design; (d) management; (e) art; (f) audio; (g) writing; (h) miscellaneous and uncategorizable roles.

constant over time.

During the process of developing these visualizations, we initially restricted our analysis to a limited subset of the entire dataset to validate our output. Specifically, we focused exclusively on games released for the DOS platform, operating under the assumption that no new games would have been released for such an outdated platform in several years at least. At first, when we noticed that nodes were nevertheless being output for 2016, we assumed this was caused by a bug. We were later surprised to discover that this was not in fact an error: some developers continue to create DOS games to this day, and new DOS games have been released as recently as 2016.

In figure 2 we were curious about the nodes being drawn at the bottom of each graph, representing developers who were only credited with making one game. We graphed this data in figure 3 and found that it follows similar trends as some of the graphs in figure 2. This could be in large part due to individuals entering the game industry and leaving after a single project. If this occurs with a consistent proportion of developers over time, that would explain this trend.

Last, we want to point out interesting outlier data that we found. We see that there were many developers working in audio who were credited for working on very large numbers of projects, including one developer who was credited for contributing audio to nearly 300 games. There were many cases where a developer would produce many games in a short period of time and not produce anything else until 10-20 years later. In many cases this might be caused by working on games that enter "development hell": a state in which a single game project spends many more

years in development than initially planned, sometimes jeopardizing any possibility that the project will ever be finished and released. There were also cases of individuals of individuals contributing to large numbers of projects at one time, such as one developer who was credited for work on nearly forty games in one year.

5 CONCLUSION AND FUTURE WORK

In this paper, we discussed the application of visualization methods to data from the Mobygames database in order to answer questions pertaining to the experience of developers within the game industry. We constructed three discrete web browser-based visualizations offering different views of the same underlying dataset and combined these into a single data exploration application.

Our visualizations permitted us to answer questions pertaining to two topics. First, by treating the game industry as a single large graph structure (in which nodes represented individual game projects and edges between pairs of games represented developers who worked on both games in question), we were able to develop a visualization of changes in community structure within this large graph over time. Second, by visualizing in aggregate the career paths of many individual developers, we were able to answer questions pertaining to the duration of employment for individual developers who entered the industry at different times; the rate at which individual developers in different roles contributed to new game projects over time; and the rate at which developers left the industry over time due to a variety of factors (including, at minimum, burnout, retirement, and adoption of jobs in other industries or fields).
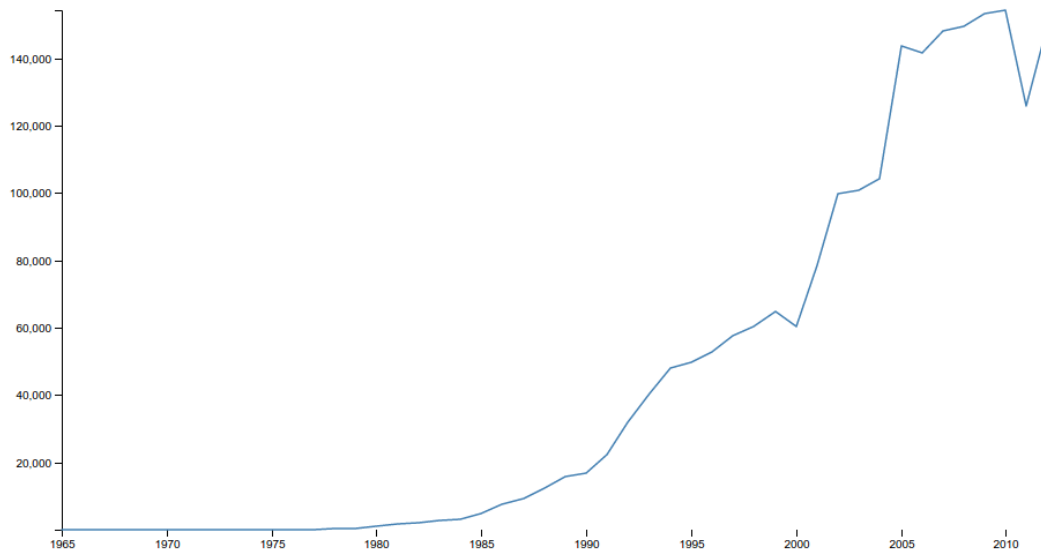
Figure 3. Developers credited on a single game

One potentially fruitful direction for future work involves the analysis and visualization of "second-order" emergent data from the "first-order" data we examined in this paper. For instance, although the visualizations we have already constructed are in many ways sufficient to give us a high-level overview of trends in the industry as a whole, we remain unable to answer questions about, for instance, the shape and duration of the "median" game developer's career. Developing visualizations of this derived data could make the data exploration application we developed here significantly more useful for examining trends in the entire dataset.

REFERENCES

[1] Mobygames. 2017. MobyStats. Retrieved from http://www.mobygames.com/moby_stats.

[2] IGDA. 2016. Developer Satisfaction Survey. Retrieved from https://c.ymcdn.com/sites/www.igda.org/resource/resmgr/files_2016_dss/IGDA_DSS_2016_Summary_Report.pdf

[3] Marie-Josée Legault, Johanna Weststar. 2015. Working Time among Videogame Developers, 2004-14. Retrieved from https://docs.google.com/viewer?url=http%3A%2F%2Fr-libre.teluq.ca%2F697%2F1%2FWorkingTimeSummaryReport-Final-15.pdf

[4] Johanna Weststar. 2015. Understanding video game developers as an occupational community. Information, Communication & Society, 18,10 (April 2015), 1238-1252, DOI:10.1080/1369118X.2015.1036094

[5] Marie-Josée Legault, Johanna Weststar. 2015. The Capacity for Mobilization in Project-Based Cultural Work: A Case of the Video Game Industry. Canadian Journal of Communication 40, 2 (2015), 203–221.

[6] Michael Schmalz. 2016. Limitation to Innovation in the North American Console Video Game Industry 2001-2013: A Critical Analysis. Ph.D Dissertation. The University of Western Ontario, London, ON, CA.

[7] C. Vehlow, F. Beck, P. Auwärter, and D. Weiskopf. 2014. Visualizing the Evolution of Communities in Dynamic Graphs. Computer Graphics Forum 34, 1 (Feb. 2015). 277–288. DOI:10.1111/cgf.12512

[8] C. Erten, P. J. Harding, S. G. Kobourov, K. Wampler and G. Yee. 2003. Exploring the Computing Literature Using Temporal Graph Visualization. Electronic Imaging (July 2003) 45-56. DOI: 10.1117/12.539245

[9] Bruce W. Herr, Weimao Ke, Elisha Hardy & Katy Börner. 2007. Movies and Actors: Mapping the Internet Movie Database. Conference Proceedings of 11th Annual Information Visualization International Conference (2007), 465-469.

[10] James Ryan, Eric Kaltman, Timothy Hong, Katherine Isbister, Michael Mateas, and Noah Wardrip-Fruin. 2016. DiGRA/FDG

[11] James Owen Ryan, Eric Kaltman, Andrew Max Fisher, Timothy Hong, Taylor Owen-Milner, Michael Mateas, and Noah Wardrip-Fruin. 2015. Large-Scale Interactive Visualizations of Nearly 12,000 Digital Games. In Proc. Foundations of Digital Games.

[12] James Ryan, Eric Kaltman, Andrew Max Fisher, Taylor Owen-Milner, Michael Mateas, and Noah Wardrip-Fruin. 2017. GameSpace: An Explorable Visualization of the Videogame Medium. UCSC technical report UCSC-SOE-17-14. University of California Santa Cruz, Santa Cruz, CA.

[13] Adam Summerville, Joseph C. Osborn, Christoffer Holmgard, and Daniel W. Zhang. 2017. Mechanics Automatically Recognized via

Interactive Observation: Jumping. FDG'17. DOI: 10.1145/3102071.3102104

[14] Tanya Short. Crunch is Failure. Retrieved November 23, 2017 from https://docs.google.com/forms/d/e/1FAIpQLSc2LGt1HOZN5G9 dPS81qPxmY2f3TaqmBTMeVUvter50n5u_Lg/viewform?fbzx=- 8544976915473095303

[15] Nick Dyer-Witheford and Greig de Peuter. 2006. "EA Spouse" and the Crisis of Video Game Labour: Enjoyment, Exclusion, Exploitation, Exodus. Canadian Journal of Communication 31, 3, (2006).