An Empirical Study of Open Source Virtual Reality Software Projects

Irving Rodriguez and Xiaoyin Wang
Department of Computer Science, University of Texas at San Antonio

***Abstract*—In this paper, we present an empirical study of 1,156 open source virtual reality (VR) projects from Unity List. Our study shows that the number of open source VR software projects are steadily growing, and some large projects attracting many developers are emerging. The most popular topic of VR software is still games. We also found that VR developers often face miss-commit of automatically generated files.**

I. Introduction

VR is an emerging technique that may reshape the world to a large extent in near future. A recent report from Deloitte Global [1] mentioned that the market of VR has reached 1 billion in 2016. Both VR device providers (e.g., Samsung, Facebook, Google, HTC) and traditional game engine companies (e.g., Unity, Unreal Engine, Cry Engine) have released software development frameworks. In year 2015, Google Play market has observed more than 25 million downloads of VR apps for Google Cardboard [2], a low-end VR device.

Despite the rapid growth of VR software projects, there has not been empirical studies on open source VR software projects. To better understand the state of the practice, in this paper, we present an empirical study on all 1,156 open source virtual-reality software projects collected from Unity List [3] on Apr. 1st, 2017, an on-line VR software repository holding open source software projects based on Unity 3D framework.

II. The Study

In our study, we try to answer the following four research questions.

- **RQ1:** How fast does the number of open-source VR software projects grows?
- **RQ2:** How many developers are involved and interested in open source VR software projects?
- **RQ3:** What are the most popular topics covered by open-source virtual-reality software projects?
- **RQ4:** What are the common files committed in open-source virtual-reality software projects?

A. Growing Trends

To answer **RQ1**, we studied how the frequency of new-project addition and code commits changes over time. The earliest project in Unity List was added to Github in May 2013, which is four year ago. Figure 1 and Figure 2 present the trends of quarterly newly added projects and code commits for each quarter since Q2-2013, respectively. From the figures, we have the following observations. First, the quarterly newly added projects and code commits have been growing steadily since the end of 2015. In the first quarter of 2017, there were 252 newly added projects and 5,954 code commits, which are about 4 times of the number in the third quarter of 2015. Second, there was a short burst of newly added projects and code commits at the beginning of 2015, which we believe is due to the acquisition of Oculus by Facebook in 2014.

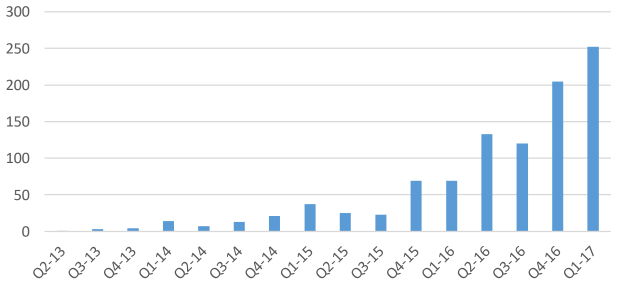

Fig. 1: Growth of Quarterly Newly Added Projects

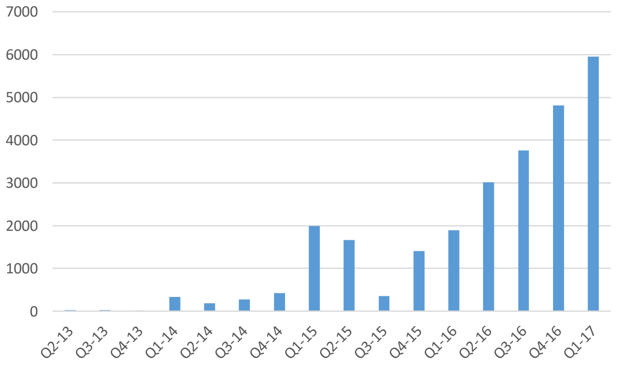

Fig. 2: Growth of Quarterly Total Code Commits

B. Developer Involvement

To answer **RQ2**, we studied how many developers are involved in Unity List projects as contributor. Furthermore, we studied how many developers (or users) are watching, or have starred or forked the projects. Figure 3 shows the proportion of projects with certain number of contributors, watchers, or being starred / forked for certain times. From the figures, we can see that, most projects have zero or one watchers, stars and forks, and one contributor. At the same time, there are non-trivial number of projects that involve and attract multiple

[1] https://www2.deloitte.com/global/en/pages/technology-media-and-telecommunications/articles/tmt-pred16-media-virtual-reality-billion-dollar-niche.html

[2] http://mashable.com/2016/01/27/google-cardboard-user-numbers/#3GvShGu4Nkq2

[3] www.unitylist.com

978-1-5090-4039-1/17 $31.00 © 2017 IEEE
DOI 10.1109/ESEM.2017.65

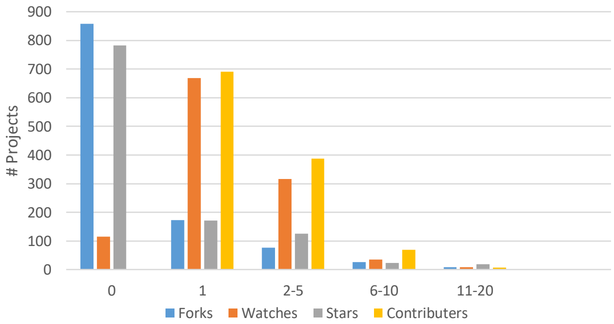

Fig. 3: Statistics on Github Meta-data

TABLE I: Top Keywords in Project Descriptions

Content Keywords		Technique Keywords	
game	206	oculus	129
demo	33	google	91
tutorial	29	hololens	71
leap	23	cardboard	71
maze	22	vive	68
scene	22	rift	67
motion	21	3d	67
shooting	15	htc	50
course	11	video	12
basketball	11	daydream	10

developers. Specifically, there are 48 projects with more than 5 forks (with maximum of 853), 57 projects with more than 5 watchers (with maximum of 339), 80 projects with more than 5 stars (with maximum of 1,835), and 77 projects with more than 5 contributors (with maximum of 49).

C. Popular Keywords

To answer **RQ3**, we collected the brief description of each project on their entrance page at Github. Then, we studied the popularity of keywords. To avoid the result being dominated by a project with very long description, for a keyword, we count at most one appearance for each project. We further removed the stop words and common words in Github such as "projects", "git", etc. Table I shows the most popular keywords on topics and techniques. From the table, we can see that, the most popular topics of VR software are games, followed by demos, tutorials (potentially for the Unity 3D framework). Other popular topic keywords include "leap", "maze", "scene", "motion", etc., which are likely to be type of games. As for techniques, the most popular ones are Oculus, Google (Cardboard and Daydream), Hololens, and HTC vive.

D. Changed Files

To answer **RQ4**, we studied the distribution of newly added and modified files with different file extensions. Figure 4 (1) and (2) show the most commonly modified and added file types, respectively. From the figures, we can see that the most commonly modified files are `.cs` files (source files for C#, one major programming language supported by Unity), followed by `.meta` files (meta data of assets such as pictures, audio media, etc.), `.info` files (information of assets in libraries), and `bin` files (cached shading effects of objects). By contrast, the most added files are also of these four types, but the added `.cs` files are least, which show that they are most actively revised. Actually, besides `.cs` files, all 3 other file types are files automatically generated by Unity framework, which shows that developers often have difficulties in deciding which files should be ignored in git commits.

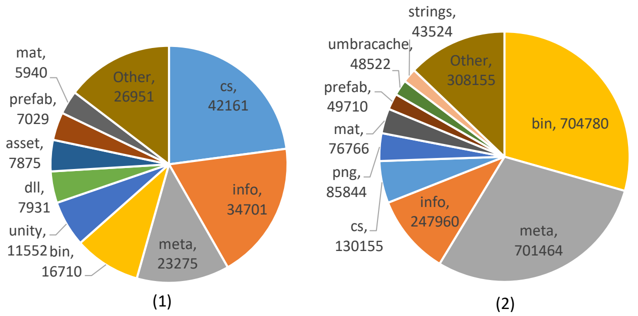

Fig. 4: Breakdown of Changed Files in Code Commits

III. Related Works

Recently, Murphy-Hill et al. [1] studied the specialty of video-game development, which is relevant as there are many similarity between video games and VR software, but they did not mention the status of open source VR software projects. There are also general studies on the meta data [2] [3] and source / test code [4] [5] of open-source projects, as well as analysis of call-back centric code [6] which are popular in software UI [7]. For VR software development, Bierbaum et al. [8] proposed *VR Juggler*, a platform for virtual reality software development. Reitmayr and Schmalstieg [9] proposed a software architecture for virtual reality interaction.

IV. Conclusion

In this paper, we present an empirical study on 1,156 open source VR software projects from Unity List. We studied the growing trends, development involvement, popular topics, and frequent file changes in these projects.

References

[1] E. Murphy-Hill, T. Zimmermann, and N. Nagappan, "Cowboys, ankle sprains, and keepers of quality: How is video game development different from software development?" in *ICSE*, 2014, pp. 1–11.

[2] H. Borges, A. Hora, and M. T. Valente, "Understanding the factors that impact the popularity of github repositories," in *ICSME*, 2016, pp. 334–344.

[3] B. Vasilescu, A. Serebrenik, M. Goeminne, and T. Mens, "On the variation and specialisation of workload—a case study of the gnome ecosystem community," *ESE*, vol. 19, no. 4, pp. 955–1008, Aug 2014.

[4] S. Mostafa and X. Wang, "An empirical study on the usage of mocking frameworks in software testing," in *QSIC*. IEEE, 2014, pp. 127–132.

[5] X. Wang, L. Zhang, and P. Tanofsky, "Experience report: How is dynamic symbolic execution different from manual testing? a study on klee," in *ISSTA*. ACM, 2015, pp. 199–210.

[6] H. Tang, X. Wang, L. Zhang, B. Xie, L. Zhang, and H. Mei, "Summary-based context-sensitive data-dependence analysis in presence of callbacks," in *POPL*, 2015, pp. 83–95.

[7] X. Wang, L. Zhang, T. Xie, Y. Xiong, and H. Mei, "Automating presentation changes in dynamic web applications via collaborative hybrid analysis," in *FSE*, 2012.

[8] A. Bierbaum, C. Just, P. Hartling, K. Meinert, A. Baker, and C. Cruz-Neira, "Vr juggler: a virtual platform for virtual reality application development," in *IEEE VR*, 2001, pp. 89–96.

[9] G. Reitmayr and D. Schmalstieg, "An open software architecture for virtual reality interaction," in *VRST*, 2001, pp. 47–54.