Survey: Software Search in Serious Games Development

Philippe Tamla [1], Thilo Böhm[1], Kerstin Gaisbachgrabner[3], Jana Mertens[2], Matthias Hemmje[2], Michael Fuchs[4]

[1] FernUniversität in Hagen, Faculty of Multimedia and Computer Science, Hagen, Germany
[2] Research Institute for Telekommunikation and Cooperation, Dortmund, Germany
[3] Graz University of Technology, Graz, Austria
[4] Wilhelm Büchner University of Applied Science, Pfungstadt, Germany
philippe.tamla@studium.fernuni-hagen.de

Abstract. The interest in serious games (games that can educate, train, and motivate) has grown very fast in the last decade, yet the research community rarely studies serious game development. For instance, previous work has mainly investigated non-game developers' search practice and has proposed techniques and tools for general purpose software engineering. This paper reports on a pilot user study that was used to understand the thoughts and needs of (serious) game developers. We asked serious game developers "what", "how", and "why" they usually search online to develop their games.
Our study reveals that software search for serious games development is mostly related to finding game-specific tools, documentation, and algorithms for AI, animation, rendering, and learning analytics. Also, specific web platforms (like web tutorials, Q&A sites, and public software repositories) are very relevant for serious games development. Finally, our analysis of serious game developers' search habit informs about the need for more advanced search engines with sophisticated query and filtering facilities.

Keywords: Software Search, Search Engines, Serious Games Development.

1 Introduction

Serious Games (SGs) are video games intended to educate the players instead of pure entertainment (David R. and Sandra L. 2005). The interest in SGs and their application in education and businesses areas has increased very fast in the last decade (Vasudevamurt and Uskov 2015). SGs provide an engaging and self-reinforcing context in which the player is motivated and instructed towards non-game events or processes, including business operations, training, marketing, well-being and advertisement (David R. and Sandra L. 2005). Thus, it is not surprising that many global companies (like Microsoft, IBM, SAP, Adobe, Cisco, Google etc.) have started integrating SGs and gamification techniques in their main business functions, processes, and activities (Anderson and Rainie 2010).

But the efforts, time and costs of serious games development can be reduced if existing software assets and other pedagogical components available on the market can be effectively retrieved (or reused) and integrated into the development process (Wim V et al. 2016). However, this requires to understand what specific actors involved in the development of SGs (like researchers, developers, designers) are usually seeking software and which factors can influence the way they locate and use what they need to develop their games (Hucka and Graham 2018).
This study reports on a deeper exploration of serious games developers' search practices. It was conducted as an online survey (using Google Drive) with professionals and practitioners in SG development. The main goal of this study was to explore the search habits of SG developers in order to understand the factors that can influence the way they *search* and *reuse* software and other related information. The survey was conducted using the following research questions:

- RQ1: What software and other related information do you usually search to develop your serious games, and why?
- RQ2: What are some approaches you usually use to FIND software and related information you need to develop your serious games?
- RQ3: While searching online in the past, what are some obstacles that may have hindered your ability to FIND (RQ3.1) or REUSE (RQ3.2) what you needed to develop your serious game?
- RQ4: Please describe one or more situations when you were trying to find a specific software or any software-related information on the Web…?

Our research questions will be answered by interpreting their results. These questions will be indexed and discussed in the following chapter.

2 Background and Related Work

2.1 Serious Games Development

Nowadays, we observe a growing number of software engineers and researchers focusing on SG development (Brent and Bill 2014; David R. and Sandra L. 2005; Mestadi et al. 2018; Petridis et al. 2012; Söbke and Seicher 2016; van der Vegt et al. 2016). Also, established business companies like Google, IBM, SAP are motivated to design, develop, and implement full-scale SGs in their business functions and processes because of the following advantages: SGs have the ability to 1) solve complex problems collaboratively, 2) improve the efficiency of business processes, 3) support predictive modelling and real-time visualization, 4) increase ROI[1] from processes, time, and resources, and 5) provide more retention of knowledge compared to traditional methods (Vasudevamurt and Uskov 2015). The creation of SGs is also a complex process of game design, programming, content production, and testing (Westera et al. 2016). And its success significantly depends on the quality of external technical gamification

[1] ROI (Return on Investment) - https://www-01.ibm.com/software/rational/rsar/roi/

platforms, dedicated software architecture (Wim V et al. 2016), reusable SG engines, and advanced technology components (software assets) (Wim V et al. 2016). However, there is a lack of standardization, best practices, and tools (Vasudevamurt and Uskov 2015) for supporting the development of SGs and the reuse of game assets which have the ability to preserve and enhance the games' pedagogical effectiveness (Westera et al. 2016). Thus, SG developers often rely on a large set of entertainment-based features and game engines despite the inherent differences between SGs and entertainment-based games (Brent and Bill 2014; Petridis et al. 2012)**.** But, the complexity, time, and effort for creating SGs can be reduced if existing tools and game assets available on the market can be effectively reused (Wim V et al. 2016).

2.2 Software search and related user studies

Software Search is a very common practice in software engineering that aims at discovering suitable software for a given purpose (Hucka and Graham 2018). It has become an important component of real, day-to-day software engineering (Singer et al. 2010) because it can increase work productivity (e.g. through effective reuse). Researchers in software search have tried to understand the search practices and challenges of developers for general software engineering (Barua et al. 2014; Bauer et al. 2014; Hucka and Graham 2018; Xia et al. 2017) and mobile development (Rosen and Shihab 2016). But the study of serious games developers' search practices has remained almost ignored so far.

Techniques used to study software search include *web survey*, *interview*, *search logs*, and *analysis of Q&A sites* (Barua et al. 2014; Rosen and Shihab 2016; Sadowski et al. 2015). (Umarji et al. 2008) surveyed software engineering implementing standard software and discovered that developers were usually looking for blocks of codes (like parsers, wrappers), libraries (for date manipulation, speech processing), stand-alone tools (like an application server or an ERP package). They also discovered that developers were looking for subsystems like data structures, parsers, binary search algorithms that they could reuse in their own implementation without or less modification. (Singer et al. 2010)s' survey informed about the frequency in software search. The study revealed that developers spend 66% of their time reading software documentation, 57% fixing bugs, and 35% making enhancements to their software system. (Singer et al. 2010) observed software engineers at their workplace and discovered that searching was an activity that occurs not only during coding but also while interacting with the hardware (e.g. doing configuration tasks) or debugging. This study also informed about the criteria of selection of software, like working functionality, evaluation of online communities or local experts, and the social characteristics of the software project (like its popularity).
To understand the specific needs and challenges of serious games developers, we recently investigated a popular Q&A site (GameDev Exchange) to find out what SG developers usually search to develop their games (Tamla et al. 2019). We have found that serious games developers usually seek a different kind of help such as best practices, common design patterns, and specific algorithms to develop better serious games that can really train, educate, and motivate players (David R. and Sandra L. 2005).

This survey reports on a deeper exploration of serious games developers' search practices. It was conducted as an online survey (created with Google Drive) with professionals, researchers, and students involved in the development of (serious) games. The study reports on the motivations of serious games developers seeking specific software and related information, how they go about finding help and which challenges they usually encounter to FIND and/or REUSE what they need. The study also informs about potential improvements in next-generation search engines.

3 Pilot User Study

3.1 Method

Nowadays, there is a wide range of algorithms, tools, best practices, and frameworks available online that can be used to facilitate the development of full-scale serious games. To understand the challenges in identifying and reusing such tools and features, an online survey (developed with Google Drive) has been conducted. The main goal of this evaluation was to explore the search habits of serious games developers and to understand the factors that can affect the way they search and reuse software and other related information. So, we used our previously defined research questions RQ1-RQ4. Our research questions will be answered by interpreting their results. These questions will be indexed and discussed in the following chapter.

3.2 Participants

Different consortium partners took part in this survey. We advertised the survey to external gamification companies and educational institutions. Overall, this evaluation was attended by 40 people: 10 software developers (2 software architects, 7 programmers, 1 game designer), 6 scientists, 6 project managers (including 4 CEOs), and 18 students, all involved in the conception, design and implementation of serious games.

3.3 Evaluation Instrument

We used a web-based survey to conduct our evaluation because 1) information can be gathered very easily and quickly from a wide audience, 2) the development effort is very modest and reduced, 3) data can be analyzed quantitatively and qualitatively. The survey consisted of a combination of standardized questionnaires (including free-text and multiple-choice questions) and 1 open question about a specific search experience. We designed the instrument of our analysis iteratively after analyzing previous surveys targeting software search in computing (Rosen and Shihab 2016; Xia et al. 2017). We especially paid attention to the following points:

- *Simplicity*: we created simple and clear questions. We also added explanatory text to some questions that may lead to some ambiguity.

- User experience: we considered only questions that we assumed were within the experiences of our audience
- *Relevance to user's experiences*: we constructed our questions by referring to specific software components and information that we believe are relevant to (serious) game users today.
- *Ethics*: We removed all questions that seemed to be too personal or about proprietary policies (at participants' place of work)

Overall, we defined 3 main categories to collect data on the evaluation:

- *Demographics*: In this section, we collected (prior work (Böhm et al. 2013)) general information about the age, education, and role of the participants.
- *(Serious) Game Development Experience*: This section explores the experience of the participants with the development of (serious) games for specific game genres and target audiences. We also asked questions about general programming experience like experience with programming languages and development paradigms.
- *Search Experience*: Questions in this category were derived from existing research that studied software search in general (Hucka and Graham 2018). We explored (serious) games developers' search habits and tried to explore the problems they may encounter to FIND and REUSE software and other related information. We also defined one open question to ask respondents about their experience with software search.

3.4 Procedure

The evaluation was carried out from August to October 2018. We solicited responses using convenience sampling by sending an invitation to experts, researchers, and students from higher educational, all involved in the development of serious games.

Especially, we advertised the survey via e-mail distribution lists addressing different members of different members of the consortium. RAGE project partners (i.e. UCM, INESC, OUNL, UU, UPB, TUGRAZ). Applied gaming research and development actors: serious-games-solutions.de (The Gamification Expert), oztron.com (Serious Games and Simulations for Education Technology), kastanie-eins.de (Games and Learning), bible.com/kids (Apps and Games to teach the Bible to kids). Educational institutions: the German Institute for Games (Hochschule der Medien Stuttgart (HDM), ifg.hdm-stuttgart.de) and the Institute of computer science of the New York Institute of Technology (NYIT, nyit.edu).

The survey itself was conducted using the online survey platform Google Form. While more users trialled the system, in total 40 participants took part in this evaluation and provided their feedback via the online survey.

3.5 Results

Demographics. The participants aged between 18 and 64. We asked participants about their experience in serious games development: 56% had less than one year, 24%

had between 1 and 3 years of experience, and 20% had between 6 and 20 years of experience. We also asked participants about their roles in their last serious games projects: 46.7% worked as Software Developer, 43% worked as Software Architect (Team Lead), 30% worked as Game Designer, and 10% worked as Test Analyst. The number of years in programming and the roles of participants were routinely used in past research to estimate the experience of developers (Feigenspan et al. 2012).
Additionally, we asked for the highest academic degree: 28.2% of the respondents hold a Bachelor's Degree (or equivalent), 33% a Master's Degree (or equivalent), and 15% a Doctor's Degree (or equivalent). The remaining 20.8% were students before their first academic degree.
Participants were also asked to self-rate their search experience (Böhm et al. 2013). On a five-point Liker scale, 37.8% rated themselves as experts, 32.4% as experienced, 13.5% as moderately experienced, and 2.7% as inexperienced. Results show that in addition to the high level of familiarity in search practices, the group of respondents is characterized by a high degree of education, research and (serious) games development experience.

Search Motivation. To understand *why* serious games developers search specific software and related information, we asked: "What software and other related information do you usually search to develop your serious games, and why?". The question contains a multiple choice box, allowing respondents to complete the list (Other). Other included "Research papers". Figure 1 summarizes the results to this question.

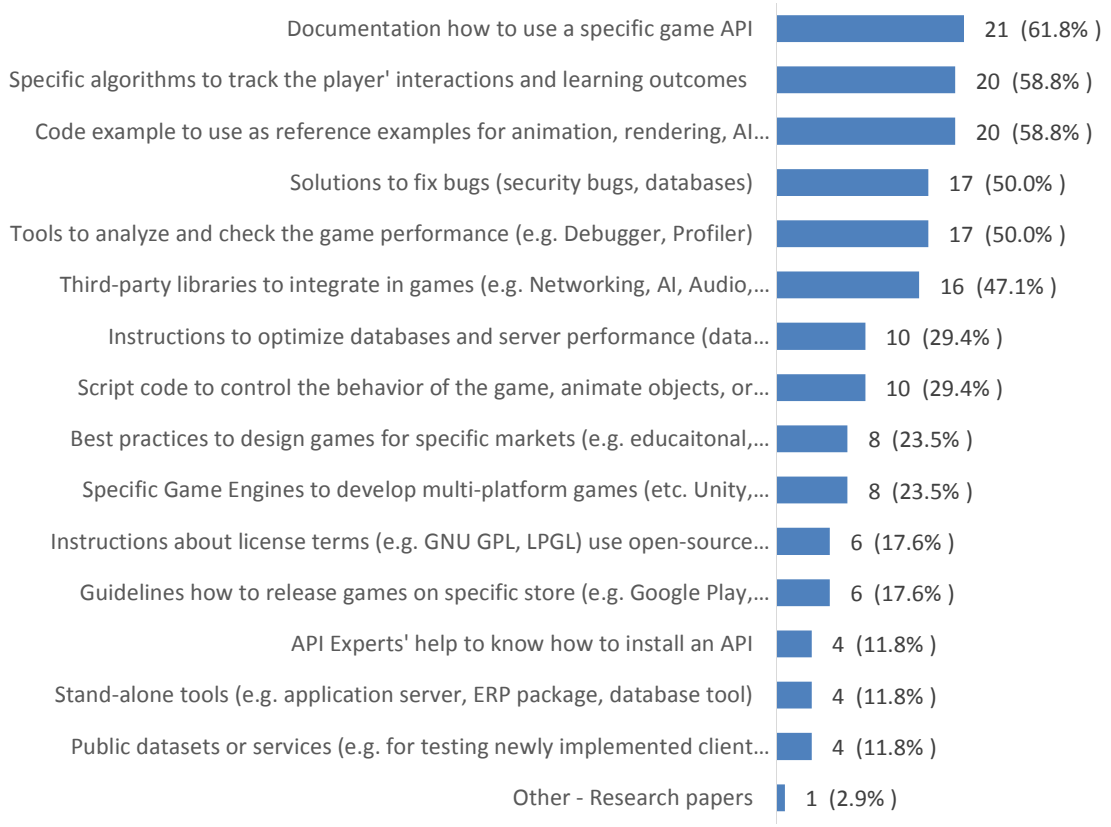

Figure 1: Responses to the question "What software and other related information do you usually search to develop your serious games?". *This question offered an additional slot for free text under "Other". Answer choices were nonexclusive. A total of 34 respondents answered this question.*

We found that looking for *documentation how to use specific game APIs* (61.8% of respondents), *specific algorithms* (to assess players' learning outcomes, increase their motivation, etc.) (58.8%), and *code snippets to use as references examples* (58.8%) were the most common motivations. Followed by *solutions to fix bugs* (50%), *tools for analyzing and checking the performance of games* (50%), and third-party libraries (50%).

Search Approach. To explore *how* serious games developers go about finding help, we asked respondents about the approaches they frequently use to find search software and related information, e.g. asking colleagues. Answer options were nonexclusive multiple choices, including "Other" option with a field for free-text input. Figure 2 provides a summary of our findings. 31 participants answered this question. The result shows that *asking colleagues or fellow students* (67.7% of participants), *using general-purpose search engines* (64.5%), *visiting standard web pages* (54.8%), and s*earching public software repositories* (45.2%) were the most common approach used by the respondents. Others included social network resources (Youtube Tutorials) and specific web site (Asset Store). Figure 3 summarizes the most popular online web pages consulted by 33 respondents. Other include "developers.google.com/games". In total 5 online sites have been named by the respondents, whereby GitHub (69% of respondents) and Stack Overflow (69%) are the most popular, followed by Asset Store sites (for unity and unreal

game engines) (33%). This result shows evidence about the importance of social networks and online communities for serious games development.

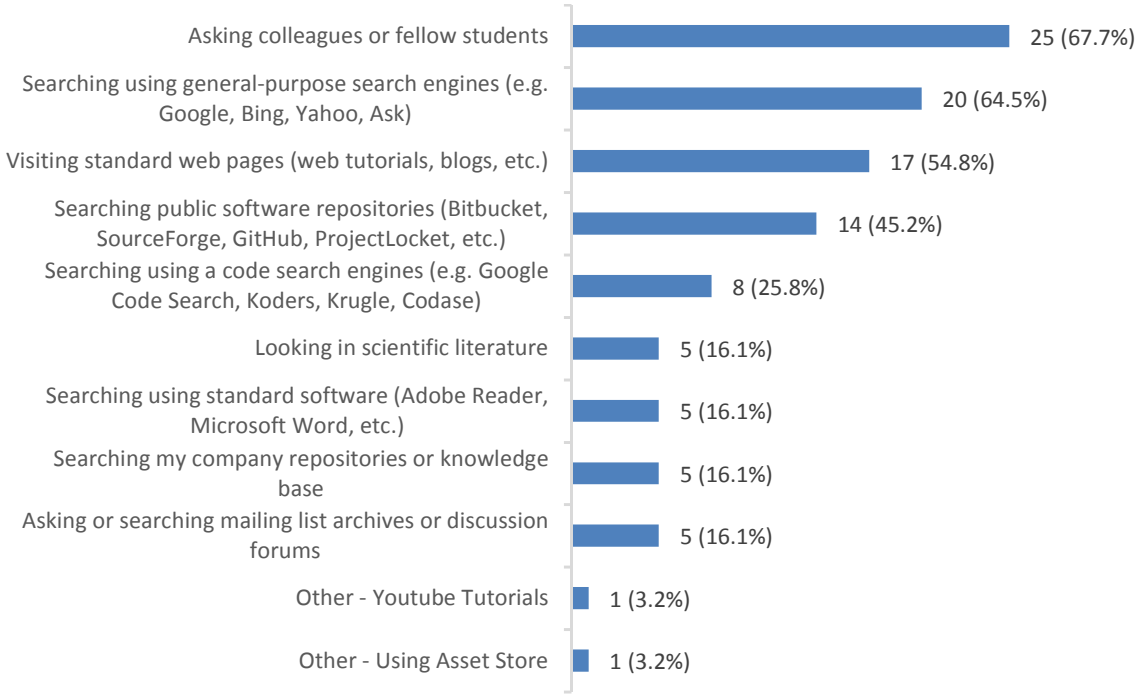

Figure 2: Responses to the question "**What are some approaches you usually use to find software and related information you need to develop your games?**" This question offered an additional slot for free text under "Other". Answer choices were nonexclusive. A total of 31 respondents answered this question.

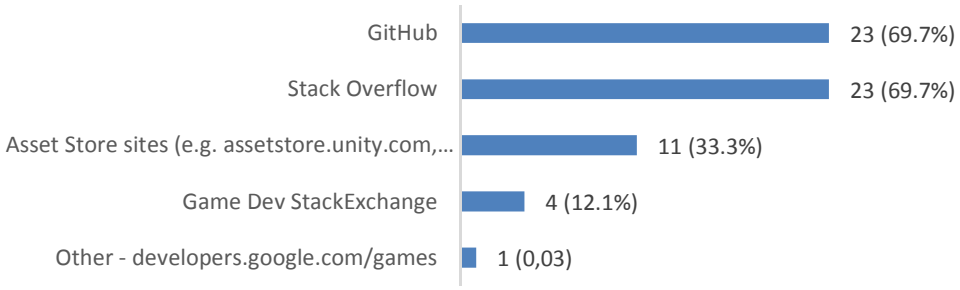

Figure 3: Responses to the question "**Do you also search specific web pages? If yes, please select the ones that match and/or complete the list**". This question offered an additional slot for free text under "Other". Answer choices were nonexclusive. A total of 33 respondents answered this question.

Challenges in Software Search. To understand the different challenges while seeking software and related information, we asked respondents what can prevent them to *find* (Figure 4) and *reuse* (Figure 5) what they found on the Internet. Respondents could select challenges in a multiple-choice box or extend the list with the "Other" option. Figure 4 reveals that 65.6% of our respondents had difficulty to *find* help online because of the following problems: *requirement was too unique* (65.6%) (which

suggest that, in some special context, they might not know what exactly to search for), *unable to locate close match* (code snippet) to use as reference example (40.6%), *wrong search queries formulated* (37.5%), and *too many alternative solutions to choose from* (31.3%). Our survey also reveals that "*poor formatted source code*" (18.8%) could not prevent serious games developers to find software although this may affect the detection of reference examples (close match). Figure 5 summarizes the results about the challenges while seeking software for *reuse*. Incomplete functionality (59.4% of respondents), poor documentation (56.3%) were the most difficult challenges to software reuse. Followed by *too much effort to integrate third-party libraries* (43.8%), *lack of testing instructions* (40.6%), and *incompatibility with the target system* (40.6%).

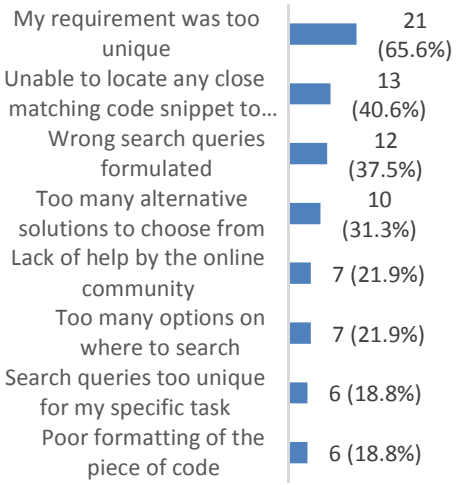

Figure 4: Responses to the question **"While searching online in the past, what are some obstacles that may have hindered your ability to FIND what you needed to develop your game?"** This question offered a multiple choice box with an additional slot for free text under "Other". Answer choices were nonexclusive. A total of 32 survey respondents answered this question. A total of 32 survey respondents answered this question.

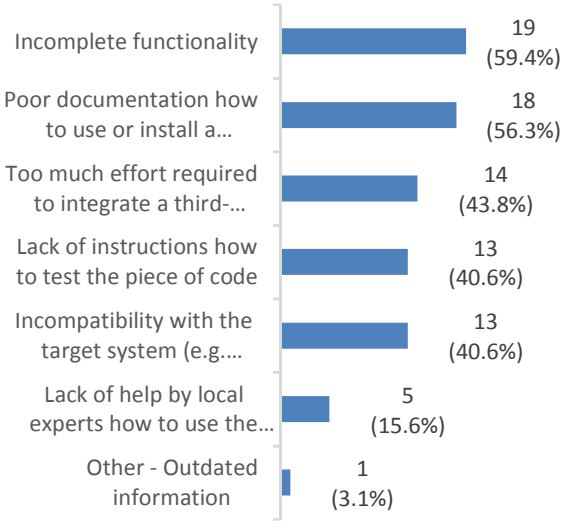

Figure 5: Responses to the question "**What are some obstacles you may have faced in the past to REUSE the software or other related information you found on the Internet**". This question offered a multiple choice box with an additional slot for free text under "Other". Answer choices were nonexclusive. A total of 32 survey respondents answered this question**.** A total of 32 survey respondents answered this question..

Search Case Experience. We wanted to learn more about how respondents seek software, so we sought examples of participants' past experiences. We asked the following questions: "**Please describe one or more situations when you were trying to find a specific software or any software-related information on the Web. What were you trying to find? How did you formulate your search queries? What approaches did you use? What problems did you have to find and/or reuse what you found? And how useful was the search result?"**

The survey form provided a text editing field where participants could write their responses in free-form text. We received a total of 12 responses, of which 8 contained substantial details about past search experience. Table 1 provides three examples taken from among those 8 responses. The analysis of these examples provides evidence that

serious games developers are trying to overcome the permanent *information overload* found in existing search engines (like Google). They seek advanced search features such as filtering by a specific programming language (#User1), or how to optimize search queries by describing the search context with keywords (#User3).

*#User1 – Unity supports multiple script languages (js, C#) and ui options to do one thing. Often this is a problem with normal search engines like google because **you can't filter for a specific language** and **get a lot of code snippets you can't use**. For example when you want to add an gravityfield to an object in c# and search for "unity add gravityfield" you find a good answer in the unity forum for js but nothing for c#.*
*#User 2 – I start with 2 /3 words. If no significant result i add another word . Problem: **too many sponsored responses within the top results***
*#User3 – Once I searched for a tool that could generate JavaScript code for a node server. My main search terms where "swagger", "node", "code gen". Unfortunately **the search engine just returned a lot of bullshit** like a small project called "swagger-node-codegen" (written in JavaScript). Nothing really helpful for me and my purpose. After several days of investigation, I found a code generation tool, written in Java, which also produces/generates JavaScript (NodeJS-Server) code.* *I think, the main problem was, that the search terms I used where to "generic" for this specific search request and even in different conjunction there are too much "possibilities" about what I could have needed. In other words - I was not able to describe my requirements in a unique and distinct search request. What I missed was **the possibility to describe my context**! For example, that **I need the resulted code to be JavaScript**, **not that the generator is written in JavaScript**.*

Table 1. Answers to the open-ended question "Please describe one or more situations when you were trying to find a specific software or related information on the Web (What were you trying to find? How did you formulate your search queries? What problems did you have

Information Desire about Software. To help inform the development of advanced search tools, we sought to determine what kind of features our respondents would like to see implemented in future search systems. We posed the following question to all participants "***What issue(s) do you think needs to be addressed in existing search mechanisms and/or tools to support serious game development?***" (Figure 6). The question was in form of a multiple choice box with an "Other" option as free text. As shown in Figure 6 "Outdated exclusion" and "Content context-sensitive search" were added to the Other option. The analysis of responses to this question reveals that there is a need for more sophisticated filtering and query features that are well integrated with existing search engines, are easy to use and can optimize search results based on the user's context more efficiently.

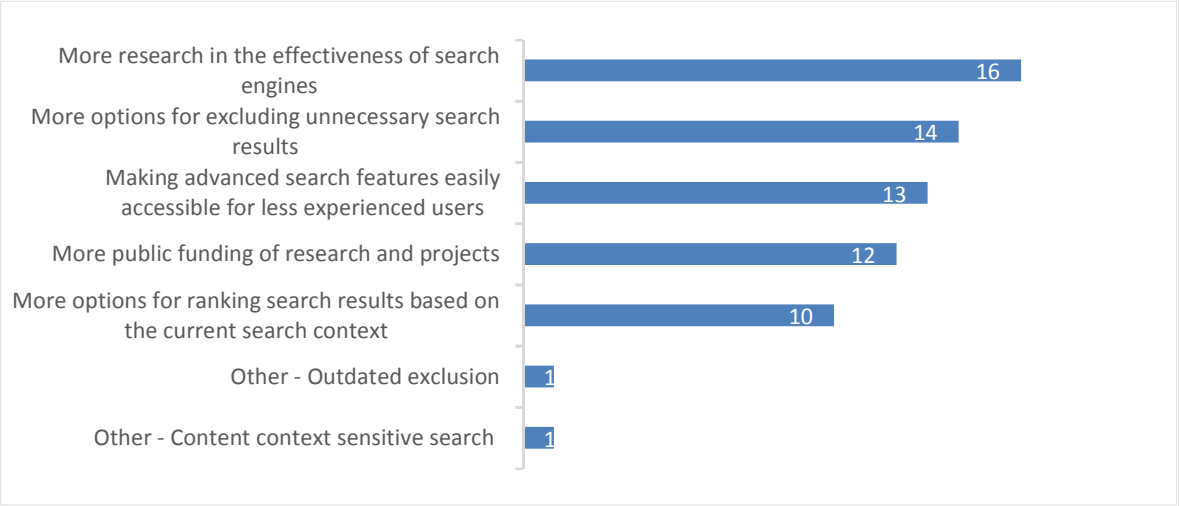

Figure 6: Response to question "**What issue(s) do you think needs to be addressed in existing search mechanisms and/or tools to support serious game development?**" This question offered a multiple choice box with an additional slot for free text under "Other". Answer choices were nonexclusive. A total of 32 survey respondents answered this question.

4 Discussion and Conclusion

Overall, the feedback gathered in this study shows that specific online tools (like search engines, public software repositories, and Q&A sites) are relevant for (serious) games development, because they can provide different kind of help (instructions, algorithms, and tools) that can facilitate and accelerate the development of serious games. Also, this survey provides evidence about the need for more advanced search engines with sophisticated query and filtering facilities.

Acknowledgement

This research was partially funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No 644187, the RAGE project (www.rageproject.eu).

References

1. Anderson JQ, Rainie H (2010) Gamification: Experts expect'game layers' to expand in the future, with positive and negative results. Pew Internet & American Life Project
2. Barua A, W. Thomas S, E. Hassan A (2014) What are developers talking about? An analysis of topics and trends in Stack Overflow 19:619–654
3. Bauer V, Eckhardt J, Hauptmann B, Klimek M (eds) (2014) An exploratory study on reuse at google. ACM
4. Böhm T, Klas C-P, Hemmje M (eds) (2013) Supporting Collaborative Information Seeking and Searching in Distributed Environments. LWA

5. Brent C, Bill K (2014) A Survey of Frameworks and Game Engines for Serious Game Development. 2014 IEEE 14th International Conference on Advanced Learning Technologies
6. David R., Sandra L. (2005) Serious games: Games that educate, train, and inform. ACM
7. Feigenspan J, and Kästner C, Liebig J, Apel S, Hanenberg S (eds) (2012) Measuring programming experience. IEEE
8. Hucka M, Graham MJ (2018) Software search is not a science, even among scientists 141:171–191
9. Mestadi W, Nafil K, Touahni R, Messoussi R (2018) An Assessment of Serious Games Technology: Toward an Architecture for Serious Games Design. International Journal of Computer Games Technology 2018
10. Petridis P, Dunwell I, Panzoli D, Arnab S, Protopsaltis, Aristidis and Hendrix, Maurice, Freitas S de (2012) Game engines selection framework for high-fidelity serious applications. International Journal of Interactive Worlds
11. Rosen C, Shihab E (2016) What are mobile developers asking about? a large scale study using stack overflow. Empirical Software Engineering:1192–1223
12. Sadowski C, Stolee KT, Elbaum S (2015) How developers search for code: a case study. Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering:191–201
13. Singer J, Lethbridge T, Vinson N, Anquetil N (2010) An Examination of Software Engineering Work Practices: CASCON First Decade High Impact Papers
14. Söbke H, Seicher A (eds) (2016) Serious Games Architectures and Engines. Entertainment Computing and Serious Games. Springer
15. Tamla P, Böhm T, Nawroth C, Hemmje M (eds) (2019) What do serious games developers search online? A study of GameDev Stackexchange. IEEE
16. Umarji M, Elliott S, Crista L (eds) (2008) Archetypal internet-scale source code searching. IFIP International Conference on Open Source Systems. Springer
17. van der Vegt W, Nyamsuren E, Westera W (2016) RAGE reusable game software components and their integration into serious game engines:165–180
18. Vasudevamurt VB, Uskov A (2015) Serious Game Engines: Analysis and Application. Electro/Information Technology (EIT), 2015 IEEE International Conference on
19. Westera W, van der Vegt W, Bahreini K, Dascalu M, van Lankveld G (2016) Software Components for Serious Game Development
20. Wim V, Westera W, Nyamsuren E, Georgievand A (2016) RAGE architecture for reusable serious gaming technology components 2016:3
21. Xia X, Bao L, Lo D, Kochhar PS, Hassan AE, Xing Z (2017) What do developers search for on the web. Empirical Software Engineering 22:3149–3185