What do serious games developers search online? A study of GameDev StackExchange

Philippe Tamla [1], Thilo Böhm[1], Christian Nawroth[1], Matthias Hemmje[2], Michael Fuchs[3]

[1] FernUniversität in Hagen, Faculty of Multimedia and Computer Science, Hagen, Germany
[2] Forschungsinstitut für Telekommunikation und Kooperation e.V. (FTK), Dortmund, Germany
[3] Wilhelm Büchner University of Applied Science, Pfungstadt, Germany
philippe.tamla@studium.fernunitat-hagen.de

Abstract. The interest in serious games (SG) has grown very fast in the last decade. Yet the research community rarely studies SG and the challenges SG developers are facing while creating games that should educate, train, and inform players. In this research, we use Latent Dirichlet Allocation (LDA) topic model to analyze a popular Q&A site (Game Dev StackExchange) in order to understand the thoughts and needs of SG developers. We analyze the most active and popular topics about SG development. Our study reveals that SG developers' needs are very diverse, ranging widely from simple script code to simulate 3D learning environments, over best practices to design educational games for specific target groups, to features that can increase the player's motivation and learning outcomes.

Keywords: Serious Games Development, Q&A sites; Latent Dirichlet Allocation, Topic Models, Software Search, Software Reuse

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). However, the creation of effective SGs is a difficult and complicated process which requires an appropriate balance between game design and instructional design, including an in-depth understanding of learning, cognition, emotion, and play (Brent and Bill 2014). 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 reused and integrated into the development process (Wim V et al. 2016). However, this requires to understand what SG developers search, and which factors can influence how they locate and select software (Hucka and Graham 2018). Existing research has mainly studied the needs of developers for traditional software engineering, ignoring the challenges of SG developers. But, SG development is different from other traditional software engineering disciplines because of its specific challenges related to the theoretical understanding of learning, cognition, emotion and play (Bellotti et al. 2010). But, to date very little is known about the development and maintenance challenges of SG developers. Therefore, we conducted a study by analyzing discussion threads on Game StackExchange (GDX) site[1] with the aim of understanding search behaviours and practices of (serious) games developers.

We used LDA topic model to identify major topics and areas of interests in SG developers searched for. We discover our topics by using SG terms and the most popular tags used in SG related posts. We rank our topics based on the number of documents. Finally, we analyzed the most popular and active topics by considering the number of documents assigned to each topic and their views. We conduct our study by using the following research questions: **RQ1**: What issues do serious games developers search online? **RQ2**: What are the most popular topics about serious games development?

This paper is organized as follow: We review related works (section 2), highlight our research questions and describe our methodology to detect SG-related posts (section 3). Then, we present and discuss our findings (section 4) and conclude (section 5).

2 Related works

The research most closely related to our work are those investigating SG development and software search.

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[2] 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 platforms, dedicated software architecture (Wim V et al. 2016), reusable SG engines, and advanced technology components (software assets) (Wim V

[1] GDX is a popular Q&A site used by thousands of (serious) game developers from all around the world to post questions and discuss issues about game development.

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

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).

Software Search is a common practice in Software Engineering that aims at discovering suitable software for a given purpose (Hucka and Graham 2018). Existing research has shown that *Software Search* is 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). Studies investigating *Software Search* in the past have tried to understand developers' search practices and challenges (Ko et al. 2006) including in mobile development (Rosen and Shihab 2016). These studies relied on techniques, such as, web survey, interview, search logs, or a combination of those (Sadowski et al. 2015), but the analysis of Q&A discussion sites like StackExchange are becoming very popular due to the increasing number of opensource tools and search engines available online (Barua et al. 2014; Rosen and Shihab 2016). However, to the best of our knowledge, there is no study that has studied SG development. Thus, in this research, we will study what SG developers search and will examine their specific needs and challenges while developing games for serious purposes.

3 Research Methodology

In this section, we describe our methodology to identify SG related posts. We present and motivate our research questions. Then we describe our data collection and analysis process which we use to answer our research questions.

3.1 Research Questions and Motivation

RQ1: What issues do serious games developers search online? – Past research on *Software Search* has mainly studied what developers seek for general purpose software engineering (Hucka and Graham 2018; Singer et al. 2010). But the analysis of SG developers' specific needs has remained almost ignored so far. Surprisingly, the literature often argues that SG development differs from general purpose software engineering (Murphy-Hill et al. 2014) because of the challenges related to the pedagogical effectiveness of games (Westera et al. 2016) which include the development of new techniques and tools for training, educating, and motivating learners (David R. and Sandra L. 2005). Therefore, we are motivated to analyze what SG developers search online and what issues they may face.

RQ2: What are the most popular topics about serious games development? – Discovering the most popular SG discussions can help the SG community to identify major areas of interest for future investigations. For instance, game companies may fine-tune various aspects of their products and SG publishers may point to areas in which practitioners have the most challenges and interest. Also, researchers may discover new research opportunities with a high chance of having an impact on practice.

3.2 Data Extraction and Data Processing

To create our corpus, we first download the *posts.xml* file from the GameDev StackExchange data dump (last updated June 2018) which contains user-generated questions and related answers[3]. We parse the data dump which includes a total of 110,193 posts, from which 42639 are questions. For each extracted post, we maintain a record of its metadata: the post type, timestamp and the total number of views. We then process our corpus by removing all existing code snippets that are present in the posts (i.e. enclosed in <code> HTML tags), as well as common English words which do not help to create meaningful topics as demonstrated in past research (Barua et al. 2014). We then tokenized each word and remove all common English words (adverbs, pronouns, and prepositions), and punctuations using the LingPipe library (Baldwin and Dayanidhi 2014). We finally create our corpus by considering only question posts because each question provides interesting insight about what the game developers have tried. Answer posts oft discuss solutions to existing issues but do not highlight the problem itself. Thus, we extract the *title* of each question, which nicely summarizes the issue being discussed, and the related *body* which gives additional context information about how this problem was initially approached by the asker.

3.3 Identifying SG-related posts

In order to found which posts of our corpus are relevant, we need a way to filter out posts which are not really related to SG development. To achieve this goal, we examined prior work (Barua et al. 2014) and defined an initial set of SG-related keywords. We considered a list of SG synonyms as proposed by (Michael Hoffmann 2016), a set of primary markets for SG (David R. and Sandra L. 2005), and the top ten most popular SG engines (Cowan and Bill 2014) as shown in **Table 1.** Initial keywords used to find SG-related posts. However, with our initial keywords, we couldn't find enough posts that were suitable for our analysis. So, we applied *the tag threshold* $TRT\_tag = \frac{No\ of\ SG\ posts}{\text{Total no.posts}}$ to our collection algorithm (proposed by (Rosen and Shihab 2016)) to identify more SG related posts by using the most popular tags assigned to each post previously found with our initial keywords. **Fig. 1** shows the top 30 most popular tags used in SG-related discussions ranked by their popularity (the number of posts to which a tag is assigned). We experimented different *TRT_tag* values and found that 35% was suitable for our analysis. With this value, we were able to identify 18% more SG-related posts from our corpus. As shown in **Table 2**, our SG-related corpus contains in total 18902 posts. We believe that the list in our corpus contains the most popular posts about serious games which we define here as *SG-related posts*.

SG Synonyms	Applied Game – Applied Gaming – Serious Game – Serious Gaming – Gamification – Games for Change – Games for Good
SG Markets	Military – Government - Education – Corporate-Games – Healthcare – Politics – Religion – Art

[3] To be found at: https://archive.org/download/stackexchange

SG Engines	Unity – Flash – Second life – E-adventure – Unreal – Torque3d – Cocos – Xna

Table 1. Initial keywords used to find SG-related posts

All posts	42639
SG posts found with initial keywords	11262
SG posts found with TRT>=35%	18902
All tags assigned to SG posts	941
Tags relevant to SG posts	337

Table 2. SG related posts and SG related tags

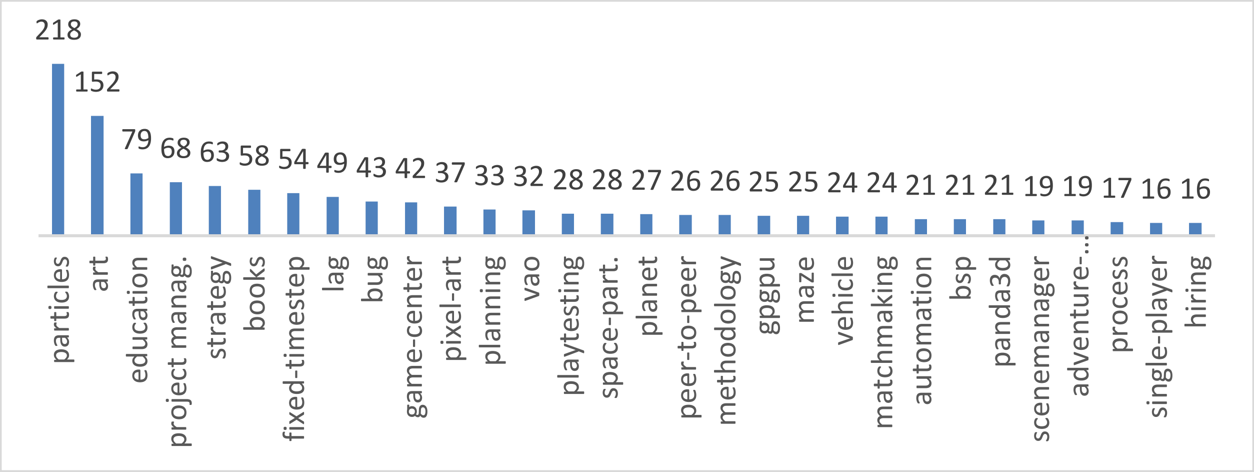

Fig. 1. The 30 most popular SG-related tags

3.4 Identifying the most popular SG-related topics

To discover the most popular SG topics, two metrics were introduced. First, the *topic relevance metric* $R(t) = \frac{Dist(d)}{\max(Dist(d_i))}$ represents the distance of a topic distribution assigned by LDA to a document to the highest distribution assigned to that same document. We use this metric as a threshold to consider only topics with the highest distribution values as the most relevant ones. Usually, a document will have 1 to 5 most dominant topics assigned to it with a distribution of 0,10 or higher (Barua et al. 2014). However, due to the probabilistic nature of LDA, we found some topics with a very small document distribution (e.g. 0.01) which introduced noises in our analysis. By using an R(t) of *35* as our distribution cutoff, we were able to eliminate existing noises in our analysis and could detect only relevant topics to our documents. Second, *the average number of views* $\mathrm{Av(t)} = \frac{SumDocumenstViews(t)}{TotalTopicDocuments(t)}$ was introduced to analyze the popularity of each topic t. *TotalTopicDocuments* is the number of documents assigned to a topic t and *SumDocumentsViews* is the sum of views of all topic documents. Existing research has shown that most of the online discussions are visited by passing (not registered) users (Mamykina et al. 2011) and that the number of views assigned to each post may indicate the importance of a topic (Rosen and Shihab 2016). By using the above-described measurements, each topic can be analyzed using two different dimensions: the number of documents assigned to a topic indicating how active a topic is, and the number of views of topic documents indicating the popularity of each topic in the SG community.

3.5 Topic Modeling using LDA

Topic Modeling is a document clustering technique that uses statistical methods for identifying abstract topics within a document collection. Several topic modelling techniques have been developed and evaluated (such as *Probabilistic Latent Semantic Indexing*, *Non-negative Matrix Factorization*) (Ding et al. 2008). In this research, we use Latent Dirichlet Allocation (LDA) topic model as it was routinely applied to millions of software engineering web documents (Barua et al. 2014; Rosen and Shihab 2016). LDA uses the frequencies of words and topics (groups of words) from documents of a text corpus to discover patterns and build a model based on related words and topics (Rosen and Shihab 2016). The number of topics (K) is defined by the user to control the granularity of discovered topics and there is no standard value of K that can fit every situation and dataset (Grant and Cordy 2010; Wallach et al. 2009). We experimented different K values and found out that 30 yields good results without being too restrictive. We also optimized our Dirichlet hyperparameters by using $\alpha = 0.5$ and $\beta=0.1$ to control how much fuzziness is allowed in the topics' distributions across words and topics respectively. It was shown that optimizing these the facto standard heuristics parameters can dramatically improve the consistency in topic usage as the number of topics increase (Rosen and Shihab 2016). To better classify the issues found in our corpus, we add their dominant topic as determined by the LDA topic model. We labelled our topics manually by considering the top words found in each topic and by examining a sample of posts assigned to each topic.

4 Results

In this section, we present the results of applying our research methodology to answer our research questions. We first examine the percentage of SG-related posts to all existing posts in our corpus. Then, we describe each discovered topic and finally discuss each topic qualitatively.

4.1 Exploring the importance of serious games related posts

Before delving into our research questions, we would like to understand the importance of SG-related posts in our corpus. Thus, after extracting and processing our corpus, we can now examine (as shown in **Fig. 2**) the number of SG-related posts (red line) and their percentage (blue line) to all existing posts per month. Our analysis shows that at least 36% of all existing posts are related to SG development. There is an increase of SG-related posts from August 2010 to March 2013 reaching a maximal value of 300

SG-posts (50%) in July 2012. Then, there is a small decline leading into November 2014 which finally gets stabilized over the following years till 2018 with an average of 27%. Thus, we conclude that the number of SG-related posts is growing and that discussions about SG development are relevant in the game development community.

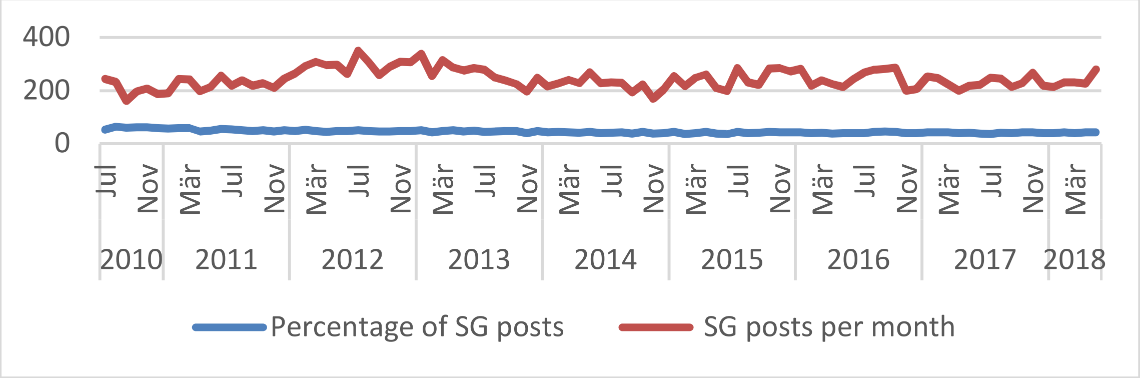

Fig. 2. Percentage and Evolution of SG posts on GameDev StackExchange per Month

4.2 RQ1: What issues do serious games developers search online?

Table 3 shows 14 out of the 30 topics discovered by our LDA methodology from which we merged some topics because of their semantical similarity. For instance, the topics *Collision Detection* and *Pathfinding* are related to *Game Physics* and thus were merged together. We arranged these topics in descending order according to their *share metric* as proposed by (Barua et al. 2014). as proposed in (Rosen and Shihab 2016). In general, we found that SG posts fall into one of the following topics: *Game Design &* Learning Design, *(3D)Modeling/Rendering, Physics, Scripting,* and *Networking*.

Topic	*Top keywords*
Graphics Engine	game java graphics creating render lwjgl library don function simple math framework multiple engines question opengl threads testing people web
Game Physics/Collision Detection/Path Finding	movement physics boxd speed libgdx simulation body path finding velocity projectile bullet gravity distance ball collision detection network
Networking	server multiplayer game client side players network mmo online player networking architecture rts sync prediction browser udp prevent communication clients
Mesh	unity mesh rendering order error performance problem device correct triangle water android dynamic problems udk collider vertices triangles text gui
File&Format	xna file project monogame custom files content culling frustum format screen loading static write running pipeline possible read asset creation
Scripting	unity working script gameobject add properly audio object playing method new level doesn issue editor scaling set child gameobjects isn
Data structure	game data code structure objects implement build libgdx visual studio board designing font node cocosd rts platforms ways gamemaker style
Platform	time canvas game html real games play javascript app google mobile step mode delta devices application run simple services program
Game Design	game games design good pattern idea creating practice making bad management gameplay effects mobile common adventure patterns story book examples
Component	system based component entity turn components systems entities particle architecture event ecs handling design strategy combat battle oriented events types
Learning Design	learning design games learning api starting programming start platform card resources indie software programmer computer developer started language development
Camera	camera screen position object mouse move moving libgdx set target issue coordinate person click shooter transform touch follow environment full
3D Modeling/ Rendering	multiple model models blender textures mesh single sprites objects parts unity rendering meshes building support fbx separate techniques dealing combine
Movement	object point vector rotation direction rotate camera sphere rotating move calculate specific calculating plane axis angle points quaternion find tower
Animation	animation character sprite unity create unityd animations skeletal states behavior flash jump controller bone animate max android trees start animated
Game Engine	game engine open code source legal issues based engines design racing window assets browser small music develop console put projects

Table 3. SG-related topics discovered with our LDA

To get a better insight about our discovered topics, we show a subset of examples related to each group of topics along with some context information (like title, body, tags, #postId) whenever needed, and will discuss their impact on SG Development.

Game Design. We found one topic about game design. Our study reveals that SG developers seek best practices and experts' opinions for taking important design decisions, like choosing a proper design pattern (#32093) or selecting an appropriate platform to create educational games (#8359).

*Title**:** Design Pattern for **Social Game Mission Mechanics** Body**:** When we want to **design a mission sub-system** like in the The Ville or Sims Social, **what kind of design pattern / idea** would fit the best? There may be relation between missions (first do this then this etc...) or not. **What do you think sims social or the ville or any other social games is using for this? I'm looking for a best-practise method to contruct a mission framework** for tha game. How the well-known game do this stuff for large scale social facebook games? [Topic: Game Design, **Tags**: game-design,flash, social. PostId**:** #32093]*

*Title**:** What makes a good educational game? – Body**:** I'm currently creating a game framework/engine for educational games. My hope is that this can be used in elementary schools. **What makes a good educational game? Which platforms should I target?** [Topic: Game Design, Tags: serious-games cross-platform educational-games.. PostId**:** #8359]*

Also, SG developers seek techniques to implement AI for tracking the progression of the player (#129492), assess the player's learning outcomes (#94079) and interest (#8925), and increase his engagement in learning (#149096, #28540).

*Title**:** Machine learning to improve strategy game AI - Body:I am currently working on a simple strategy game as a hobby and I am starting to think about designing an AI to add opponents ….My idea was **to record the player's activity every X amount of time** as well as the current state of some game variables. Then I would use some machine learning algo (as a black box) to find a relationship between game variables and actions to make. I would also record the final score at the end of each game so that the **AI learns more from successful games than unsuccessful ones** (or even better set a score threshold above which to send the data to the AI)… **Any thoughts or directions towards interesting resources /tutorials would be very welcome [Topic**: Level Design, Tags: game-design ai strategy reinforcement-learning, PostId**:** #129492]*

*Title**:** How can I ensure the player learns all skills in a single open level? Body: I am making a game that only has a single level. It is a survival game. … I can have the player learn new skills gradually. For example, level one requires jumping, level 2 requires climbing, level 3 requires both. **Is there a common strategy to force the player to learn all the necessary skills** to succeed? Currently I have 4 directions which the player can travel from the starting point. Each of which…[Topic: Learning Design, Tags: level-design tutorials reinforcement-learning, PostId**:** #94079]*

*Title: Things that make a **game more interesting for kids –** Body: What makes a game more interesting or fun to play… Does the age of consumer play an important role in his/her interest of game?What can I put in a game to make it more fun, educational, and interesting to play? – Topic: Game Design PostId: #8925 Tags: game-design serious games*

*Title: How to calculate players engagement?- Body: **What metrics could be used to know what certain people think** about the game and **how much they liked it**? [Topic: Game Design, PostId: #149096 Tags: game-design -design metrics]*

*Title:What are the downsides of only explicitly informing the player of *success*? – Body: I working on **educational software** and **trying to increase engagement** ("fun") [Topic: Game Design, Tags: game-design, PostId: #28540]*

There are two emerging research fields that focus on tracking and analyzing players' interactions. First, *Game Analytics (GA)* (A. All et al. 2016) focuses on developing techniques to automatically follow users' interactions and increase their engagement (maximizing the time a player stays in the "flow"). Second, *Learning Analytics (LA) (Loh et al. 2015)* aims at analyzing and measuring players' learning outcomes, for purpose of understanding and optimizing an environment in which it occurs. But, there is a lack of standard formats to represent players' interactions (Cooper 2013) and facilitate the processing and analysis of their learning progress (Ángel et al. 2017).

3D Modeling/Rendering. Our study reveals that SG developers seek instructions and efficient tools or programs for creating and rendering (nice-looking) 3D objects even with limited knowledge, skills, and time (#121634, #44168).

Title: Creating a large amount of sprites and animations. Body: For my AP U.S. **Government and Politics class**, I am making a **spinoff of Mortal Kombat**. Because of time constraints, I'm using an open source version of the game and want to **create new characters**. I have about two weeks to do the project.. **I want it to look as nice… What**

*Title**:** Software rendering 3d triangles in the proper order - Body: **I'm implementing a basic 3d rendering engine in software (for education purposes**, please don't mention to use an API). When I project a triangle from 3d to 2d coordinates, I draw the triangle. If I'm sorting all the objects, this is n*log(n)**. Is this***

programs or techniques would be best…? – [Topic: 3D Modeling/Rendering, Tags: sprites animation, PostId: #121634]

***the most efficient way to do this?** [* Topic: 3D Modeling/Rendering, Tags: 3D, rendering, software-rendering PostId**:** #44168]

3D Modeling is the process of using software to generate a representation of 3D characters and objects for use in a game (Dorribo-Camba et al. 2013; Jackson and Buxton 2007), and *Rendering* refers to the process of creating an image from 2D or 3D model (McShaffrey 2005). *Rendering* covers every aspect ranging from texture creation, lighting and shadow to materials, scene graph management (Millington and Funge 2009). Well-designed SGs that can engage and affect learners require vivid characters, realistic models and environment to be able to demonstrate behaviours with specific learning lesson(s) (Kelly et al. 2007). Although existing research has revealed the reuse of existing 3D models can reduce design time by 30% (Dorribo-Camba et al. 2013; Jackson and Buxton 2007), there is still little work in the reuse of existing 3D models. Existing techniques rely on indexing mechanisms to extract metadata (semi-automatically) from models and support reuse (Ioan Marius Bilasco et al. 2006), however, substantial efforts are still required to maintain the extracted metadata.

Physics. Game physics (known as Physics) refers to the process of introducing classical laws of mechanics in games (Millington 2007). Physics introduces the realistic feel to the game and can stimulate real world objects (e.g. lighting, collision detection, water, cloth physics, weather, etc.) (Vasudevamurt and Uskov 2015). Physics engines support the modelling of virtual world objects by means of configuration to avoid manual programming (Söbke and Seicher 2016). But physics calculations are often computationally demanding and thus need to be appropriately integrated into game design concepts in order not to distract the player, but promote immersive and engaging games (Söbke and Seicher 2016). We found that SG developers seek help to create and/or optimize their algorithms for collision detection (#75109) or pathfinding (#108338), and how to generate special characters (like living beings) and to integrate them in specific virtual environments (like world politics) (#53137).

*Title: How to **simulate early politics? –** Body: I'm making a historically accurate game where the player can interact with past times and shape the future. The entire game is scientifically generated with math and **real physics**…**The player can build anything they want** and it will **use physics to simulate each creation based on the materials it's made of.** I was considering just modeling our political system and then applying mathematical regression to the algorithm to regress it 100 million years. However, I'm worried about the limitations of floating point numbers and I think that errors will compound.…P.S. **I'm mostly talking about American dinosaurs, because they had the most impact on world politics**. [Topic: Physics, Tags: mathematics procedural-generation simulations, PostId: #53137]*

Existing game engines provide a variety of features and algorithms for AI, rendering, and simulation that can be used right away by developers (Söbke and Seicher 2016). Facilitating the detection of such features can lead to efficient SG development because developers will only focus on developing domain-specific features instead of writing low-level algorithms from scratch (David R. and Sandra L. 2005).

Scripting. Our study reveals that SG developers seek specific code (like unityscript) to control the behaviour of their games and do animation (#161824, #163929). The literature refers to three levels of coding (Blow 2004): **script code** (for controlling the overall content, rules, and high-level behaviour of games), **gameplay code** (for graphics simulation, physics), **engine code** representing special algorithms for 2D/3D animation, rendering, and programming networking. Coding plays a significant role in the development of SGs and should be completely integrated into existing SG engines in order to optimize the game's efficiency in terms of performance and memory usage (Vasudevamurt and Uskov 2015).

Networking. We found that SG developers seek help to create games that allow multiple users to play together over greater distance and on a distributed environment like the Web (#157707). Networking technologies are responsible for the scalability and security of games and also deal with issues related to bandwidth and latency (Vasudevamurt and Uskov 2015).

Title: Optimization for end-to-end multiplayer pong-like game – Body: I've developing a small pong-like ***game for educational purposes*** *and I've reached the point where I'm looking at implementing network gameplay. I was looking into what optimizations are usually performed for a smooth experience since latency can cause jitterry movements if not handled properly....*
I know that client-side prediction is used and read up on it but it seems to me that this is applies in client-server types of multiplayer. Is this true, and if yes why?
Topic: Networking, Tags: networking, multplayer, optimization, pong PostId: #157707

4.3 RQ2: What are the most popular topics about SG development?

Fig. 3 displays the top 10 most popular SG-related issues on GDX along with the average number of views assigned to each topic. We see that while *Rendering* and *Game Design* are the *most active* topics about SG (those having the largest number of posts), while topics such as *Game Physics, Animation* belongs to *the most popular* topics (having the highest number of views).

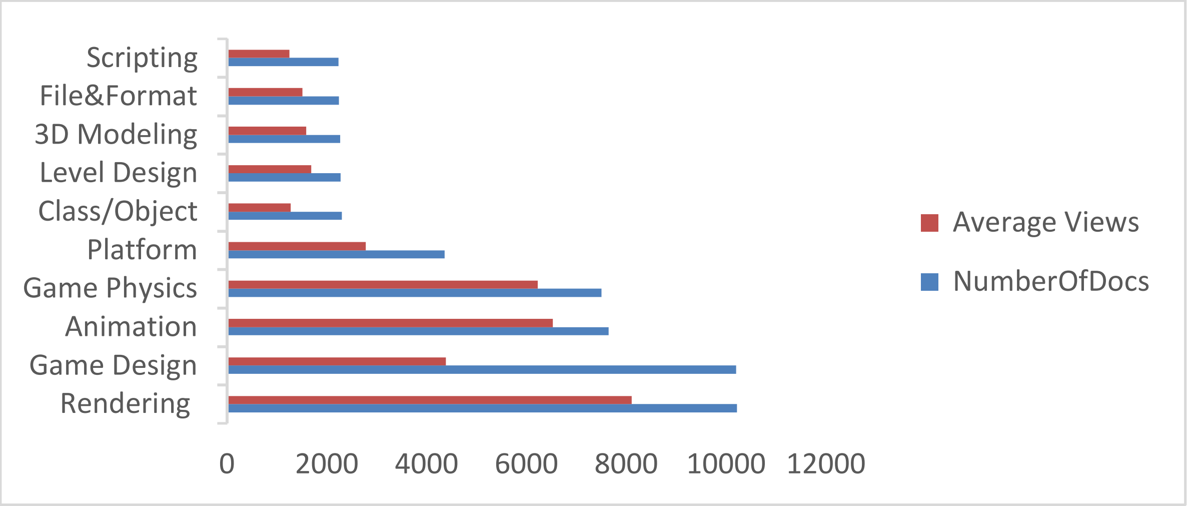

Fig. 3. Top 10 most popular SG-related topics

5 Conclusion & Future Work

In this paper, we have used the LDA topic model on GameDev StackExchange corpus to quantitatively detect and classify the main topics SG developers usually search. For instance, we have found out that *Rendering* and *Game Design* are very ***active topics*** (suggesting that SG developers are seeking best practices and common strategies to do one thing), while discussions about *Physics* and *Animation* belong to the most ***popular topics***, which suggests that they are seeking help to fix their issues. Our study also reveals that SG developers seek different things, ranging from simple scripts for stimulating 3D learning environments, over best practices to design educational games, to specific algorithms for increasing the player's motivation and learning outcomes.

We believe that future research in SG should focus more on standardizing common practices (like design patterns, design concepts) and tools to facilitate the development of specific game genres and the reuse of common features that add a good balance of learning and pedagogical objectives to games. This will motivate developers and gaming companies to easily and efficiently design, develop, and deliver full-scale SGs while being able to reduce the time to market. For this to be successful, a good understanding of SG filed and theories (game design, learning theories, and domain content) is needed.

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. A. All, E. P. Nunez Castellbar, J. Van Looy (2016) Game analytics maximizing the value of player data, 92-93. Springer
2. 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
3. Ángel S-L, Iván M-O, Jason H, Damon R, Andy J, Baltasar (2017) Applying standards to systematize learning analytics in serious games. Computer Standards & Interfaces 50:116–123
4. Baldwin B, Dayanidhi K (2014) Natural language processing with Java and LingPipe Cookbook. Packt Publishing Ltd
5. 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
6. Bellotti F, Berta R, Gloria A de (2010) Designing effective serious games: opportunities and challenges for research. International Journal of Emerging Technologies in Learning (iJET) 5
7. Blow J (2004) Game development: Harder than you think. Queue 1
8. 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
9. Cooper AR (2013) Learning Analytics Interoperability - a survey of current literature and candidate standards. http://blogs.cetis.ac.uk/adam/2013/05/03/learning-analytics-interoperability. Accessed 2 November 2018
10. David R., Sandra L. (2005) Serious games: Games that educate, train, and inform. ACM
11. Ding C, Li T, Peng W (2008) On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing. Computational Statistics & Data Analysis 52:3913–3927
12. Dorribo-Camba J, Alducin-Quintero G, Perona P, Contero M (2013) Enhancing model reuse through 3D annotations: a theoretical proposal for an annotation-centered design intent and design rationale communication. American Society of Mechanical Engineers
13. Grant S, Cordy JR (eds) (2010) Estimating the optimal number of latent concepts in source code analysis. IEEE
14. Hucka M, Graham MJ (2018) Software search is not a science, even among scientists 141:171–191
15. Ioan Marius Bilasco, Jérôme Gensel, Marlène Villanova-Oliver, Hervé Martin (2006) An MPEG-7 framework enhancing the reuse of 3D models:65–74

16. Jackson C, Buxton M (2007) The design reuse benchmark report: Seizing the opportunity to shorten product development
17. Kelly H, Howell K, Glinert E, Holding L, Swain C, Burrowbridge A, Roper M (2007) How to build serious games. Communications of the ACM 50:44–49
18. Ko AJ, Myers BA, Coblenz MJ, Aung HH (2006) An exploratory study of how developers seek, relate, and collect relevant information during software maintenance tasks. IEEE Transactions on software engineering 32:971–987
19. Loh CS, Sheng Y, Ifenthaler D (eds) (2015) Serious games analytics: Theoretical framework. Serious games analytics. Springer
20. Mamykina L, Manoim B, Mittal M, Hripcsak G, Hartmann B (eds) (2011) Design lessons from the fastest q&a site in the west. Proceedings of the SIGCHI conference on Human factors in computing systems. ACM
21. McShaffrey M (2005) Game coding complete. Scene-graphs. Paraglyph Inc
22. 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
23. Millington I (2007) Game Physics Engine Development. CRC Press
24. Millington I, Funge J (2009) Artificial intelligence for games. CRC Press
25. Murphy-Hill E, Zimmermann T, Nagappan N (2014) Cowboys, ankle sprains, and keepers of quality: How is video game development different from software development?:1–11
26. 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
27. Rosen C, Shihab E (2016) What are mobile developers asking about? a large scale study using stack overflow. Empirical Software Engineering:1192–1223
28. 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
29. Singer J, Lethbridge T, Vinson N, Anquetil N (2010) An Examination of Software Engineering Work Practices: CASCON First Decade High Impact Papers
30. Söbke H, Seicher A (eds) (2016) Serious Games Architectures and Engines. Entertainment Computing and Serious Games. Springer
31. van der Vegt W, Nyamsuren E, Westera W (2016) RAGE reusable game software components and their integration into serious game engines:165–180
32. Vasudevamurt VB, Uskov A (2015) Serious Game Engines: Analysis and Application. Electro/Information Technology (EIT), 2015 IEEE International Conference on
33. Wallach HM, Murray I, Salakhutdinov R, Mimno D (eds) (2009) Evaluation methods for topic models. ACM
34. Westera W, van der Vegt W, Bahreini K, Dascalu M, van Lankveld G (2016) Software Components for Serious Game Development
35. Wim V, Westera W, Nyamsuren E, Georgievand A (2016) RAGE architecture for reusable serious gaming technology components 2016:3