Chapter 13
Grand Challenges in Software Engineering for Games in Serious Contexts

Antonio Bucchiarone

Abstract The potential benefits of using the engaging and interactive nature of games to achieve specific objectives have been recognized by researchers and professionals from various domains. Serious games have been developed to impart knowledge, skills, and awareness in areas such as education, healthcare, and the environment, while gamification has been applied to enhance the engagement, motivation, and participation of users in non-game activities such as sustainability and learning. As a result, the fields of game design, software engineering, and user experience are increasingly converging to create innovative solutions that blend the strengths of games with real-world applications.

The main goal of this book has been to foster an environment of collaboration that unites experts from both the software engineering and game development communities. The primary aim has been to facilitate knowledge sharing, exchange of experiences, and interdisciplinary perspectives to explore the latest opportunities, challenges, costs, and benefits associated with games in serious contexts. Additionally, the book seeks to establish a fresh research agenda that aligns with the emerging trends and issues in the field.

The aim of this chapter is to provide an overview of the major challenges that must be addressed by the software engineering and game development communities to fully realize the potential of serious games and gamification in various domains.

Keywords Software engineering · Games in serious context · Grand challenges · Research roadmap

---

A. Bucchiarone (✉)
Motivational Digital Systems (MoDiS), Fondazione Bruno Kessler (FBK), Trento, Italy
e-mail: bucchiarone@fbk.eu

K. M. L. Cooper, A. Bucchiarone (eds.), *Software Engineering for Games in Serious Contexts*, https://doi.org/10.1007/978-3-031-33338-5_13

13.1 Introduction

This chapter attempts to summarize the discussions of the chapters presented in this book capturing a vision of the grand challenges facing the two communities of software engineering and games. This analysis has the unique objective to provide useful context for future research challenges and directions. We start with a summary of the key challenges presented in the overall chapters of the book, and we conclude with a brief summary of where we believe the field of software engineering for games in serious context (GSC) is going.

As we have seen from the concrete application scenario introduced in this book, GSC is gaining popularity in all those domains that would benefit from the increased engagement of their target users [1]. Thus, these applications are found in disparate contexts, such as education and training [2–6], health and environmental awareness [7–10], e-banking [11], software engineering [12], everyday challenges [13], and so forth.

The growing adoption of GSC experiences make their design and development increasingly complex due to, for example, the number and variety of users, and their potential mission criticality. This complexity is nurtured, among other factors, by a lack of theoretical grounding and adequate frameworks to engineer the intended solutions. One of the main challenge in this context is to bring the attention of interdisciplinary researchers and practitioners to the opportunities and challenges involved in the new trends and issues related to the development of GSC applications.

13.2 Grand Challenges

This section describes the challenges that have emerged in the various chapters of this book. We start from the set of challenges more related to the engineering of games in serious context (GSC), and we conclude with a set of challenges where the use of GSC could improve the software engineering aspect.

13.2.1 Design of GSC

The design of GSC is quite complex and requires numerous precautions in order to achieve a well-functioning system. In fact, according to several data in the literature [14, 15], there is little cohesion with respect to theoretical underpinnings and what gamification encompasses, leading to inconsistent results related to the use of such systems. These results can be partly explained by the lack of standardized design methodologies and the extensive use of the shortcoming one-size-fits-all strategy [16]. In other words, they are often designed without taking into account that different categories of users have different interactions with these systems. To face these results' inconstancy, several solutions have been presented. Some authors suggest that the design of GSC should take into account the final users' differences and preferences [15, 17–20], while other authors have presented specific design frameworks in order to properly design GSC [21–23] taking into account specific needs. To summarize, we need to consider to whom the GSC is directed and what the characteristics of the target group are [24]. Indeed, personalized interactive systems are more effective than one-size-fits-all approaches [18].

13.2.2 Context-Awareness in GSC

In GSC, the term *context* is often associated with *user* and *goals*; actually, it is undeniable that there is a link between these various factors. Therefore, the way in which GSC is perceived by users depends on multiple factors, including the individual characteristics of the users, the context in which the GSC is implemented, and the specific task or activity being gamified. These elements all contribute to how users perceive and engage with GSC [15]. Despite that, contextual factors and the importance of the application domain are often underestimated in GSC research and design [15]. Therefore, according to Koivisto and Hamari [15], the lack of theoretical understanding surrounding the importance of the contextual influence on gamification effectiveness might produce results that in reality cannot be generalized to other contexts.

13.2.3 User Experience Evaluation Methodologies and Tools

User experience (UX) is a multifactorial concept that is difficult to be measured [25]. In GSC, we are especially interested in the differences between traditional research and emerging evaluation of UX, such as physiological data (e.g., electroencephalography, electromyography, and facial expression assessment) [26]. Since the purpose of a user experience evaluation is to record and interpret the experience experienced by users while interacting with a digital game, it is imperative that this recording is accurate and reliable in order for its results to have substance and be useful. When measuring and evaluating user experience in GSC, it is best to employ tools from different methodologies, such as quantitative tools combined with qualitative evaluation tools or objective tools combined with qualitative evaluation tools. Taking advantage of each methodology in this way will increase the reliability of the results. Utilizing tools from only one methodology may negatively affect our evaluation efforts if we choose to leverage those tools. Finally, for better understanding and in order to interpret the experience derived from a GSC, the methodology used to evaluate the user experience plays a very important role. Future research will evaluate GSC using different methodologies and tools. These studies will ultimately be aimed at finding the most effective combination of tools and methodologies for measuring the GSC potential.

13.2.4 Software Reuse in GSC

Creating a GSC can be very complicated, with many activities, elements, and team members composing this development, taking a long time to be produced. Reuse is the concept by which it aims to build some artifact from one that has already been produced to save time and money. The gaming community has already been using reuse concepts in an ad hoc manner to create new games from existing ones. Reuse can bring some advantages for the development of GSC, such as greater longevity, lower production costs, and greater diversity of solutions being created in a shorter time. In this context, *componentization* can be used to simplify the complexity of a GSC by clearly identifying and separating the concerns. This allows for easier and longer-lasting revision management. At the same time, the *Product Line* paradigm can be exploited to reuse some GSC features and create several branches from it. Finally *Model-Driven Development (MDD)* can be used to derive the characteristics of a GSC and create models from them [27]. Once the models are already created, transformations can be applied to generate a new model that will generate new adaptations of the GSC in the future.

13.2.5 Quality Design in GSC

The development of GSC implies a learning process like the process of incorporating new learning into memory, as well as retrieving and using it. This requires a software architecture designed to optimize memory and processing resources [28]. The design patterns approach offers holism and efficiency; another important advantage of this approach is that it provides reusable solutions, which benefit the maintainability and evolution of the system [29]. In the case of devices with limited storage, such as mobile phones, they allow the optimization of resources. GSC has a greater fluidity in its operation; the player's experience when interacting with the user interface is a motivation to continue looking for efficient solutions that improve the use of resources and thus the speed of response with which the exercises to be solved are presented, leading to the development of an application agile and efficient to make your learning fun, dynamic, and permanent. For these reasons, the software engineering community must contribute studies and techniques to improve the performance of these applications.

13.2.6 Adaptation in GSC

In addition to enhancing design phases, GSC need a support for monitoring and adaptation of the gameplay. In fact, there exist concrete risks for GSC of triggering and producing undesirable side effects [30]. In most of the current approaches, the design, analysis, and revision of GSC require many development activities often unrelated to each other, with the use of various general-purpose languages (e.g., rule-based). The different actors involved (e.g., domain expert, system developer, and impact managers) use different languages and tools to execute their tasks with a completely different understanding of the game concepts and their relations. In turn, this might lead to managing unexpected game deviations with ad hoc and not reusable solutions, making the monitoring and the revision of game mechanics and dynamics a complicated task. What is really needed is the provision of uniform and clean datalogs of players' game actions. In this way, a desired monitoring framework would also assist and enhance the process of monitoring gameplay, aimed at detecting and resolving upcoming design issues at runtime. Such a tool would allow an iterative, player-centric design, in contrast to a one-size-fits-all strategy, notoriously detrimental [13, 31, 32]. Instead, adaptive content is likely to increase player engagement and motivation when it is aligned to players' preferences [33] and adjusts its difficulty to the players' skills and abilities [34]. Adapting and personalizing the gameplay to the user is more likely to foster intrinsic motivation, challenging to achieve as is notably subjective to the player [35–37]. The development of such an *adaptive by design* GSC and the addition of adaptation to existing GSC are both challenging engineering problems, which require a combination of expertise in the learning domain, game development, software development, and machine learning.

13.2.7 Abstraction and Automation in GSC

A fundamental concern of gameful applications is their tailoring to the target domain and users: if a game is detached from the domain interests, the risk is to promote counterproductive/undesired behaviors; similarly, too easy or too complex games could fail engagement objectives due to loss of interest or discouragement, respectively [38]. A direct consequence of the mentioned tailoring needs is the critical contribution and cooperation of application domain and gamification experts: the former ones provide inputs about the engagement issues and desired outcomes, while the latter ones propose corresponding gamification strategies. Such a cooperation conveys gameful application specifications to be implemented in an appropriate target platform.

In the current state of practice, one available implementation option is to pick up a pre-packaged gamification application from a repository [39]. The advantage would be to have a quick development phase limited to configuration purposes, at the price of very limited customization possibilities, unless manually tuning the existing implementation. Diametrically opposite, a completely new gamified application can be developed from scratch: this solution necessarily entails longer time to market, with the advantage of realizing a fully customized implementation. Regardless of the choice, the realization and deployment phases introduce an abstraction gap between gamification stakeholders, namely, domain and gamification experts, and the gameful application itself. In fact, the target application is typically implemented as a collection of rules matching incoming event notifications with corresponding game status updates. Therefore, developers need to translate game mechanics and other elements into corresponding rules while the other stakeholders are required to backtrack state changes into corresponding gaming events.

With the growing adoption of gamification in disparate application domains and its spreading to a wider range of users, the complexity of gameful software is unavoidably increasing. In this respect, the abstraction gap between design and realization becomes a critical issue: the implementation phase is more tedious and error-prone, due to the number of rules and the customization needs. Moreover, maintenance and evolution activities are harder to manage, due to the disconnection between design and realization.

In order to close the gap between design and implementation of gameful applications, abstraction is a key aspect that should be taken into account. A developer should use a set of domain-specific languages devoted to the specification, implementation, and deployment of gameful applications, and more in general, a software engineering process should consider the following key aspects:

Separation of concerns: a gamification approach can be described by means of several perspectives. When the complexity grows, an effective way to alleviate it is to manage different perspectives as separate points of view that are later on fused into a complete solution;

Correctness by construction: given the growth of gamification employment and range of its potential users, the specification of gameful applications becomes increasingly intricate. In this respect, game rules shall be consistent with mechanisms and elements intended for the target application;

Automation: in order to close the gap between design and implementation, the amount of manually written code shall be reduced as much as possible. Or in the other way around, the degree of automation provided by the process shall be maximized.

13.2.8 GSC for Software Engineering Education and Training

Gamification means creating a game narrative that guides players through increasingly complex challenges, keeping them engaged with social activities such as group work or competitions. It means providing immediate feedback and students taking autonomous choices to progress down the individually decided path. Gamification is not an add-on. Instead, gamification mechanics are fundamental to the learning path personalization process in two ways. Not only do they keep the students engaged, but they can also be used as tools to gain insight into the student's behavior from a different perspective and thus help generate a more personalized and engaging learning path. In order to increase engagement, the gamification mechanics must be calibrated according to the underlying activities. That is why gamification mechanics should be enhanced by AI techniques to make the motivation personalized and contextualized [40].

13.2.9 GSC for Software Quality

Software development projects often fail because of insufficient code quality [41]. It is now well documented that the task of testing software, for example, is perceived as uninteresting and rather boring, leading to poor software quality and major challenges to software development companies. One promising approach to increase the motivation for considering software quality is the use of gamification. Initial research works already investigated the effects of gamification on software developers and come to promising [42]. Nevertheless, a lack of results from field experiments exists, which motivates the need of new research in this field. Preliminary results in this direction [42] show that the introduction of a leaderboard game has a measurable effect on the Code Quality (CQ) in software development projects, while further questions for future research arise. The leaderboard can be used more intensively in teaching. In addition, it needs to be evaluated in a professional context with experienced developers. Furthermore, the degree of gamification needs to be investigated. How much is too much or too little? The optimal degree of gamification is an aspect that should be investigated more closely in future research works. The time spent on gamification can also be considered, which leads to the question of how much time should or can be spent in order to achieve the best possible results in Code Quality. In terms of motivation, it could be analyzed whether competition with others, the own performance, or the feeling of playing as a team contributes the most. In the context of a multiplayer approach, it could be considered how this affects the player motivation and outcome.

13.3 Final Discussion

In this chapter, we presented the grand challenges that cover the two main perspectives covered by this book: (i) software engineering for games in serious contexts and (ii) games in serious context for software engineering. We hope that this analysis not only represents a snapshot of the challenges faced in these research fields but contributes to stimulate researchers, practitioners, and tool developers to tackle and explore some of them. At the same time, it provides a useful context for future research projects, research grant proposals, and new research directions. We hope in a few years we can look back at this list and see many of them crossed out as a sign of the continuous advancement and maturity of these two communities together.

References

1. Koivisto, J., Hamari, J.: The rise of motivational information systems: a review of gamification research. Int. J. Inf. Manag. **45**, 191–210 (2019)
2. Dicheva, D., Dichev, C., Irwin, K., Jones, E.J., (Boots) Cassel, L., Clarke, P.J.: Can game elements make computer science courses more attractive? In: Proceedings of the 50th ACM Technical Symposium on Computer Science Education, SIGCSE 2019, pp. 1245 (2019)
3. Cosentino, V., Gérard, S., Cabot, J.: A model-based approach to gamify the learning of modeling. In: Proceedings of the 5th Symposium on Conceptual Modeling Education and the 2nd International iStar Teaching Workshop Co-located with the 36th International Conference on Conceptual Modeling (ER 2017), Valencia, November 6–9, 2017, pp. 15–24 (2017)
4. Kim, S., Song, K., Lockee, B., Burton, J.: Gamification in Learning and Education: Enjoy Learning Like Gaming. Springer International Publishing, Berlin (2018)
5. Lee, J.J., Hammer, J.: Gamification in education: what, how, why bother? Acad. Exch. Q. **15**(2), 2 (2011)
6. Bucchiarone, A., Cicchetti, A., Bassanelli, S., Marconi, A.: How to merge gamification efforts for programming and modelling: a tool implementation perspective. In: 2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C), pp. 721–726. IEEE, Piscataway (2021)
7. Johnson, D., Deterding, S., Kuhn, K.-A., Staneva, A., Stoyanov, S., Hides, L.: Gamification for health and wellbeing: a systematic review of the literature. Int. Intervent. **6**, 89–106 (2016)
8. Marconi, A., Schiavo, G., Zancanaro, M., Valetto, G., Pistore, M.: Exploring the world through small green steps: improving sustainable school transportation with a game-based learning interface. In: Proceedings of the 2018 International Conference on Advanced Visual Interfaces, AVI 2018, pp. 24:1–24:9 (2018)
9. Rajani, N.B., Mastellos, N., Filippidis, F.T.: Impact of gamification on the self-efficacy and motivation to quit of smokers: observational study of two gamified smoking cessation mobile apps. JMIR Serious Games **9**(2), e27290 (2021)
10. Vieira, V., Fialho, A., Martinez, V., Brito, J., Brito, L., Duran, A.: An exploratory study on the use of collaborative riding based on gamification as a support to public transportation. In: 2012 Brazilian Symposium on Collaborative Systems, pp. 84–93. IEEE, Piscataway (2012)
11. Rodrigues, L.F., Costa, C.J., Oliveira, A.: Gamification: a framework for designing software in e-banking. Comput. Hum. Behav. **62**, 620–634 (2016)
12. Pedreira, O., García, F., Brisaboa, N., Piattini, M.: Gamification in software engineering – a systematic mapping. Inf. Softw. Technol. **57**, 157–168 (2015)
13. Vassileva, J.: Motivating participation in social computing applications: a user modeling perspective. User Model. User Adapt. Interact. **22**(1), 177–201 (2012)
14. Seaborn, K., Fels, D.I.: Gamification in theory and action: a survey. Int. J. Hum. Comput. Stud. **74**, 14–31 (2015)
15. Koivisto, J., Hamari, J.: The rise of motivational information systems: a review of gamification research. Int. J. Inf. Manag. **45**, 191–210 (2019)
16. Böckle, M., Micheel, I., Bick, M., Novak, J.: A design framework for adaptive gamification applications. In: Proceedings of the 51st Hawaii International Conference on System Sciences (2018)

17. Bassanelli, S., Bucchiarone, A.: GamiDOC: a tool for designing and evaluating gamified solutions. In: Extended Abstracts of the 2022 Annual Symposium on Computer-Human Interaction in Play, pp. 203–208 (2022)
18. Tondello, G.F., Wehbe, R.R., Diamond, L., Busch, M., Marczewski, A., Nacke, L.E.: The gamification user types hexad scale. In: Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play, pp. 229–243 (2016)
19. Oliveira, W., Hamari, J., Shi, L., Toda, A.M., Rodrigues, L., Palomino, P.T., Isotani, S.: Tailored gamification in education: a literature review and future agenda. Educ. Inf. Technol., pp. 1–34 (2022)
20. Codish, D., Ravid, G.: Gender moderation in gamification: does one size fit all? (2017)
21. Klock, A.C.T., Gasparini, I., Pimenta, M.S.: 5w2h framework: a guide to design, develop and evaluate the user-centered gamification. In: Proceedings of the 15th Brazilian Symposium on Human Factors in Computing Systems, pp. 1–10 (2016)
22. Morschheuser, B., Maedche, A., Walter, D.: Designing cooperative gamification: conceptualization and prototypical implementation. In: Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing, pp. 2410–2421 (2017)
23. Deterding, S.: The lens of intrinsic skill atoms: a method for gameful design. Hum. Comput. Interact. **30**(3–4), 294–335 (2015)
24. Kim, B.: Designing gamification in the right way. Lib. Technol. Rep. **51**(2), 29–35 (2015)
25. Moizer, J., Lean, J., Dell'Aquila, E., Walsh, P., (Alfie) Keary, A., O'Byrne, D., Di Ferdinando, A., Miglino, O., Friedrich, R., Asperges, R., Sica, L.S.: An approach to evaluating the user experience of serious games. Comput. Educ. **136**, 141–151 (2019)
26. Atrash, A., Mower, E., Shams, K., Mataric, M.J.: Recognition of physiological data for a motivational agent. In: Computational Physiology, Papers from the 2011 AAAI Spring Symposium, Technical Report SS-11-04, Stanford, CA, March 21–23, 2011. AAAI, Washington (2011)
27. Di Ruscio, D., Kolovos, D.S., de Lara, J., Pierantonio, A., Tisi, M., Wimmer, M.: Low-code development and model-driven engineering: two sides of the same coin? Softw. Syst. Model. **21**(2), 437–446 (2022)
28. Mizutani, W.K., Daros, V.K., Kon, F.: Software architecture for digital game mechanics: a systematic literature review. Entertain. Comput. **38**, 100421 (2021)
29. Qu, J., Song, Y., Wei, Y.: Design patterns applied for game design patterns. In: Chen, Y. (Ed.) 17th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, SNPD 2016, Shanghai, May 30–June 1, 2016, pp. 351–356. IEEE Computer Society, Washington (2016)
30. Rapp, A., Hopfgartner, F., Hamari, J., Linehan, C., Cena, F.: Strengthening gamification studies: current trends and future opportunities of gamification research (2019)
31. Khaled, R., Fischer, R., Noble, J., Biddle, R.: A qualitative study of culture and persuasion in a smoking cessation game. In: International Conference on Persuasive Technology, pp. 224–236. Springer, Berlin (2008)
32. Orji, R., Mandryk, R.L., Vassileva, J.: Improving the efficacy of games for change using personalization models. ACM Trans. Comput. Hum. Interact. **24**(5), 32 (2017)
33. Lavoué, E., Monterrat, B., Desmarais, M., George, S.: Adaptive gamification for learning environments. IEEE Trans. Learn. Technol. **12**(1), 16–28 (2018)
34. Pastushenko, O., Oliveira, W., Isotani, S., Hruška, T.: A methodology for multimodal learning analytics and flow experience identification within gamified assignments. In: Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems, pp. 1–9 (2020)
35. Deci, E.L., Koestner, R., Ryan, R.M.: A meta-analytic review of experiments examining the effects of extrinsic rewards on intrinsic motivation. Psychol. Bull. **125**(6), 627 (1999)
36. Malone, T.W.: What makes things fun to learn? Heuristics for designing instructional computer games. In: Proceedings of the 3rd ACM SIGSMALL Symposium and the First SIGPC Symposium on Small Systems, pp. 162–169. ACM, New York (1980)
37. Tondello, G.F., Wehbe, R.R., Diamond, L., Busch, M., Marczewski, A., Nacke, L.E., The gamification user types hexad scale. In: Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play, CHI PLAY '16, pp. 229–243. ACM, New York (2016)

38. Hanus, M.D., Fox, J.: Assessing the effects of gamification in the classroom: a longitudinal study on intrinsic motivation, social comparison, satisfaction, effort, and academic performance. Comput. Educ. **80**, 152–161 (2015)
39. TechnologyAdvice.com. Compare 120+ gamification platforms (2019). https://technologyadvice.com/gamification/
40. Bucchiarone, A., Martorella, T., Colombo, D., Cicchetti, A., Marconi, A.: POLYGLOT for gamified education: mixing modelling and programming exercises. In: ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion, MODELS 2021 Companion, Fukuoka, October 10–15, 2021, pp. 605–609. IEEE, Piscataway (2021)
41. Vasileva, A., Schmedding, D.: How to improve code quality by measurement and refactoring. In: Paulk, M.C., Machado, R.J., Brito, M.A., Goulão, M., Amaral, V. (Eds.) 10th International Conference on the Quality of Information and Communications Technology, QUATIC 2016, Lisbon, September 6–9, 2016, pp. 131–136. IEEE Computer Society, Washington (2016)
42. de Paula Porto, D., de Jesus, G.M., Ferrari, F.C., Fabbri, S.C.P.F.: Initiatives and challenges of using gamification in software engineering: a systematic mapping. J. Syst. Softw. **173**, 110870 (2021)