An Experience Report on Using Gamification in Technical Higher Education

Alexandru Iosup
Parallel and Distributed Systems Group
Delft University of Technology
Mekelweg 4, 2628CD Delft,
The Netherlands
A.Iosup@tudelft.nl

Dick Epema
Parallel and Distributed Systems Group
Delft University of Technology
Mekelweg 4, 2628CD Delft,
The Netherlands
D.H.J.Epema@tudelft.nl

ABSTRACT

Technical universities, especially in Europe, are facing an important challenge in attracting more diverse groups of students, and in keeping the students they attract motivated and engaged in the curriculum. We describe our experience with gamification, which we loosely define as a teaching technique that uses social gaming elements to deliver higher education. Over the past three years, we have applied gamification to undergraduate and graduate courses in a leading technical university in the Netherlands and in Europe. Ours is one of the first long-running attempts to show that gamification can be used to teach technically challenging courses. The two gamification-based courses, the first-year B.Sc. course Computer Organization and an M.Sc.-level course on the emerging technology of Cloud Computing, have been cumulatively followed by over 450 students and passed by over 75% of them, at the first attempt. We find that gamification is correlated with an increase in the percentage of passing students, and in the participation in voluntary activities and challenging assignments. Gamification seems to also foster interaction in the classroom and trigger students to pay more attention to the design of the course. We also observe very positive student assessments and volunteered testimonials, and a Teacher of the Year award.

Categories and Subject Descriptors

K.3.2 [**Computer and Information Science Education**]: Computer science education; C.0 [**Computer Systems Organization**]: General—*System architectures*; C.1.0 [**Computer Systems Organization**]: Processor Architectures—*General*; H.3.4 [**Information Systems**]: Systems and Software—*Distributed systems*

Keywords

CS Ed research; gamification; experience report

1. INTRODUCTION

SIGCSE'14, March 3–8, 2014, Atlanta, GA, USA.

http://dx.doi.org/10.1145/2538862.2538899.

Technical higher education, in Europe and in many other parts of the world, is at a crossroad. The pace and complexity of technological progress, the economic pressure, and the cultural changes occurring over the past century have increased the need for a well-educated body of technical-university graduates. However, from the 2000s and onwards the number of students who enroll has started to decrease and the percentage of students who finish their studies in time has decreased. In the Netherlands, under 40% of the students enrolled in technical universities in the three-year B.Sc. curriculum finish in less than *four* years. In several European countries, including the Netherlands, the quality of education in sciences, especially mathematics, at pre-university levels has decreased significantly over the past decade[1]. Concurrently, students in South Korea, etc., are pressured into completing their courses not only on time, but also with top grades. In this context, an approach that balances between laxity and strictness is desirable. Towards this end, we present in this work an approach to teach computer science courses in technical universities that is based on gamification, that is, on using social gaming elements in the design of undergraduate or graduate courses.

Gamification may have originated in the early-Communist thought and matured in the Soviet era [15], as a substitute for monetary incentives to perform at work, and saw a re-emergence in the U.S. in the early 1980s [14]. More recently, in the 2000s gamification received various definitions [6, 11], and was used with promising results in various curricular [7, 4] and organizational [8, 15] settings.

However much already applied, for example to engage university students in a limited setting [7], gamification is not easy to apply to a new topic. For example, giving students points may foster competition and incentivize them to study harder, but it may also diminish their intrinsic motivation and (when points are not well balanced) make them feel discriminated against. The vast array of gaming elements available to designers [3], the loose theories of fun [13], and the contradictory evidence regarding the use of gamification in practice, make applying gamification in higher education a challenging task. Thus, we have set to investigate the following main research question *Can gamification be effective in teaching higher-education courses?*

[1] Dijsselbloem Committee (2008) End report to Parliamentary investigation (2007–2008), Dutch Government, Second Chamber Report.

We have applied gamification to the design of two courses in university-level computer science, both taught at `GamificationU`, which is one of the leading universities in the Netherlands and in Europe. Chronologically, the first of these courses is a first-year undergraduate course on Computer Organization that matches the Architecture and Organization (AR) knowledge area in the ACM/IEEE CS2013 curriculum [1]. We have taught a gamification-based version of this course for three consecutive years, since the academic year 2010–2011. The second of the two courses is an M.Sc.-level (graduate) course on Cloud Computing, which we have taught during the academic year 2012-2013. In total, we have educated using gamification over 450 students, of which over 75% have completed the course. Although we agree that gamification cannot solve the intrinsic problems of education units, such as poor course content and inadequate teaching skills, we find that it can lead to better course experience for the students and to better overall course outcomes.

With the two gamified courses, we have followed a variety of *instructional goals*. Our primary goal was to develop curriculum elements related to computer science education in systems at undergraduate and graduate level, especially focusing on parallel and distributed systems courses (PDC). Firstly, we have tried to help our students gain experience with system design. Understanding the relationship between abstract architectures and real-world implementation is difficult to acquire at all ages, but, for first-year students, it can be dispiriting and may be a cause for quitting studies. Secondly, we have attempted to foster learning of topics that are challenging technically, and broad and possibly also deep conceptually. The natural consequence, which we have observed first-hand in the Netherlands for the past decade, is a high failure rate among students. Thirdly, scalability and elasticity are some of the most long-lasting, challenging problems in computer science [12, 2, 10], and we are trying to educate students about them early on (the first-year undergraduate course) or in-depth (the M.Sc.-level course). Fourthly, to make the courses more relevant to emerging computer science topics [5], we have added to the undergraduate course elements of GPU architecture and programming, and, to both courses, elements of or a focus on distributed/cloud computing architecture and programming.

Towards answering the main research question, our contribution is fourfold:

1. We introduce a gamification toolbox for academic-level computer science (Section 2).
2. We design two gamification-based courses, which together cover a variety of desirable instructional goals (Section 3).
3. We approach gamification through an empirical method (Section 4). For three years, we have taught hundred of students who succeeded more than usual. We use passing ratios, participation counts, and results of evaluation surveys to quantify the effectiveness of using gamification in teaching our courses.
4. We summarize the lessons we have learned in applying gamification in academic-level CS education (Section 5). Our lessons are personal, but hopefully educative: we made mistakes, encountered student and organizational resistance, and read eye-opening student testimonials of how gamification has changed the life of our students.

2. COURSE GAMIFICATION TOOLBOX

In this section we analyze gamification as an education technique. We see gamification as using social gaming elements, such as team-work, game thinking, and game mechanics, in non-game environments (here, higher education). The main promise of gamification is that it gives the educator a number of powerful and predictable tools for influencing human motivation and behavior and, when done right, to activate various types of students in pursuing learning activities. The main challenge is to make technically and conceptually challenging courses interesting for classroom of students with various personalities and skill-levels. We describe in the following 4 types of students (Section 2.1) and 7 core tools for course gamification (Section 2.2).

2.1 Player Classes

The astute professor needs to understand and cater to different student skill-levels and personalities. We propose that skill can be assessed dynamically, via tests offered during the operation of the course, and that personality (or motivation) can be understood within the framework of four primary player-motivations defined by Bartle [3]:

1. *Explorers* are players who enjoy understanding the world–the student who is curious. Designing courses for these students is challenging for the educator, as students are interested in both the quality and the quantity of the material.
2. *Achievers* enjoy completing most of the challenges they are presented with. These are our ambitious, high-achieving students, who would strive not only to pass the course, but also to achieve at least a grade of 80%.
3. *Socializers* participate in the game mainly because other players, in particular players like them, also do. Passing the course is interesting for them, if it allows them to continue being part of the same social circle.
4. *Winners* (*Killers* in Bartle's taxonomy) want to complete challenges at the expense of other players. For them, a challenge is good if it can only have one winner (preferably themselves). Many of the top-percentile students could belong to this category. Winners may be self-destructive, in that competitiveness may push them into burn-out, depression, or boredom.

Bartle also finds [3] that any long-running game community needs players from every personality type. In particular, he finds that winners and achievers are classes of players without whom game ecosystems cannot survive. In this model, a classroom without the participation of top students quickly collapses, in both level and attendance.

2.2 Seven Core Tools for Course Gamification

Our current approach for gamification of education units fits within the framework of Zichermann and Cunningham [17]. For them, as for many traditional theorists of computer games, games contain mechanics, dynamics, and aesthetics. *Mechanics* define how games operate as systems, that is, the way they convert specific inputs into specific outputs. *Dynamics* guide how players and the game mechanics interact during the runtime of the game. *Aesthetics* refer to the way the game mechanics and dynamics interact with the game designer's artistry, to produce cultural and emotional outcomes.

We propose that mechanics and dynamics are what a course designer can systematically employ and tune to produce desired student and community behaviors; aesthetics, albeit also important, are less predictable and thus not explicitly investigated in this work.

We identify 7 core tools for gamification. The 3 core mechanics are:

1. *Point systems* are managing the acquisition and expenditure of points, that is, of units of value internal to the game, that quantify student performance. Students may be rewarded with points that count towards the course grade; for example, 1,000 points may translate into the maximum course grade of 100%. Students may also earn points that are not directly linked to the course grade; for example, points that can be spent to propose a topic to be discussed in the next course. Points that *expire*, for example after each major component of the course or after one year, may be used to level the playing field or to prevent inflation.
2. *Levels, access, and power* are ways to incentivize students to play, achieve, and excel, respectively. Levels are typically a direct result of accumulating experience (points). For example, the level of a student may be the final course grade of that student. Access describes what players can see and do inside the game system; access may be granted through level restrictions, continued good behavior, etc. For example, students may get access to additional material, one-hour mentoring sessions with the educator, exta lectures, etc. Power refers to what players are entitled to do, including access to and control of course topics. Power may be achieved through active and continuous participation in the course, acing exams, helping other students, etc.
3. *Leaderboards* are routinely used to compare achievements. Anonymous leaderboards, while not as fun as full-disclosure leaderboards, allow individual students to assess their own ranking. This may be demoralizing if actual ranks are displayed; instead, presenting a long general list of ratings and placing low-ranked players always in the middle may be a better approach.

The 4 core dynamics are:

1. *Badges and other status displays* refer to ways to show achievement. Badges quantify achievement through their challenge and scarcity, but may also be offered to surprise the students. For example, an educator may invent fun badges such as "late but smart" badge for late students who can answer a question when they first enter the class.
2. *Onboarding* is the "act of bringing a novice into the system" [17]. Most modern online games employ the simple technique of starting with "tutorial" game tasks, that is, tasks that every user is guaranteed to be able to solve with relative ease and in a short period of time. Once successful, novice players find it more difficult to leave the system.
3. *Social engagement loops* are designs that make players return to the game. Students that are part of a team have a strong social incentive to be present, if missing out diminishes the chances of the team to perform well. In-class interactions between groups of students are also important.
4. *Unlocking content* is a powerful dynamic tool for controlling the evolution of the course. Students may not be allowed access to a course component, prior to completing the core requirements of another. Top students may be allowed access to supplementary course material, tougher assignments, etc.

Table 1: Course topics and curricular reward, in credits. Each credit (EC) require abouts 28 hours of study.

`BSc-CO`, 6EC (168h)	`MSc-CC`, 5EC (140h)
Digital Logic and Data Representation	Overview of cloud computing
Computer Architecture and Organization	Scheduling and Resource Management
Interfacing and I/O Strategies	Data Centers and Energy Efficiency
Memory Architecture	Multi-tenancy concepts, incl.virtualization
Functional Organization	Cloud programming models
Multiprocessing	Case studies
Performance Enhancements	Guest lecturer
Directions in Computing	

3. DESIGN OF GAMIFIED COURSES

To gain experience with the use of gamification in practice, we have implemented two courses using the toolbox for course gamification introduced in the previous section.

The courses are a first-year undergraduate course on Computer Organization (`BSc-CO`) and a graduate course on Cloud Computing (`MSc-CC`). Table 1 summarizes the main topics for each course. `BSc-CO` covers the topics recommended by the ACM/IEEE CS2013. For the topic "Performance Enhancements", `BSc-CO` students learn about pipelining. In "Directions in Computing", `BSc-CO` includes several basic concepts of parallel and distributed computing, such as Amdahl's Law and synchronization; such approaches have been discussed in a recent SIGCSE panel [5]. We also include here extra lectures on the basics of GPU architectures and programming, and on Cloud computing architecture and programming, which are unlocked for the top 20% of the students and do not count for the course grade. `MSc-CC` is an innovative course in cloud computing, in that it focuses on the architectural (systems-building) perspective, rather than courses that focus on MapReduce [2]. Instead of programming in MapReduce, which is one of the many possible big data programming models, we focus on resource provisioning and allocation. Students can choose to deploy MapReduce-based applications, but the course does not prescribe this.

3.1 Supporting Player Classes

We devise the two courses to accommodate the various types of players described in Section 2.1, as follows. `BSc-CO` proposes to students in-class discussion about concepts, weekly tutorials that bridge the gap between concepts and practice, pair-programming in the laboratory to experiment, and self-study in teams of 6 to create opportunities for social learning. `MSc-CC` uses in-class discussion, reviews of articles and team presentations, and pair-programming in the lab.

The match to player-motivations is straightforward. Explorers have multiple paths to explore. Achievers have sufficient paths of advancement, and several options within each path. Socializers benefit from being part of two teams per course. Winners compete with each other for the most challenging aspects of the lab-work and during in-class discussion. Students can pass the course by following the path(s) that suit their intrinsic motivation.

3.2 Using the Course Gamification Toolbox

Table 2: Use of gamification mechanics and dynamics in course design.

	BSc-CO	MSc-CC
Course level	First-year, Bachelors	Masters
Point Systems	Course points, access tokens	
Levels and access	Access to various elements	
Leaderboards	Hall of Fame	
Badges	Various	No
Onboarding	Entry quiz, 5% bonus to final grade	
Social engagement loops	Teams of 2–4 for Lab	
	Teams of 6 for Self-Study	—
Unlocking content	Unlocking Lab bonus assignments	

Table 3: Row B'xy (M'xy) has results for BSc-CO (MSc-CC) in year 20xy. The ⋆ denotes a course without gamification, used as comparison. Percentages without parentheses are relative to In-class participation.

	Participants (Completed)		Bonuses		
Spring,	In-class	Lab	Self-Study	In-class	Lab
B'10⋆	93 (65%)	118 (78%)	—	—	—
B'11	122 (65%)	114 (96%)	—	1.1%	10%
B'12	147 (65%)	130 (95%)	15%	25%	4%
B'13	161 (80%)	118 (97%)	25%	32%	10%
M'13	34 (76%)	26 (92%)	—	88%	59%

To gamify the two courses, we employ the mechanics and dynamics described in Table 2. We set for both courses a points-scale in which 10,000 points are required for a course grade of 10; this large number decouples the reality of the final course grade (from 1 to 10 at GamificationU) from the abstract in-course denomination. Various access tokens challenge competitive and exploring students. Onboarding offers a small reward to students at the start of the course. Various teams offer the needed social engagement loops. For MSc-CC, where students are more mature, we use peer assessment for the presentation component of the course, after preliminary instruction on evaluation criteria.

The highest rewards require multiple steps of unlocking. The toughest *bonus* lab-assignments may ask from students the prior completion of several other course-components. The top 20% of the students are offered additional course-work, which deepen the course topic “Directions in Computing” via lectures and practical work.

To enable the use of gamification in large-enrollment courses, we use game analytics, here, analyzing the behavior and performance of students while the course is ongoing. We collect meaningful information through end-lecture and in-class quizzes, which we analyze before the next lecture. Game analytics allow us to understand, through the use of traditional data mining techniques, what the students are interested in, how each of them performs, and where individuals and the community need more guidance. In particular, they allow us to repeat in our lectures information that many students have not yet assimilated. Updating the results of individual students also keeps achievers and winners motivated.

4. EMPIRICAL EVALUATION

We summarize in this section the results of a multi-year, empirical evaluation we have conducted via the two gamification-based courses at GamificationU (see Section 1).

4.1 Impact of Gamification on Participation, Completion, and Retry On Failure

We report on data collected over 4 course sessions, characterizing over 450 students. We have managed BSc-CO, the first-year undergraduate course, for 3 Spring sessions that accommodate each 100–150 fresh students and 50–60 students who have failed the course in previous years. MSc-CC, the M.Sc.-level course, was given for the first time in Fall 2012 and attracted an unexpectedly high enrollment of over 50 students (only 36 actively participated). For BSc-CO, we also report on results prior to the introduction of gamification, for the Spring 2010 session (the limit of our access to old-course data); however, other elements of the course, such as the lecturer and the lab-work support team, have also been changed.

We summarize the results observed for our student cohorts in Table 3. These results indicate that:

- As indicated by the in-class completion percentages (column “Participants”, “In-class”, between parentheses), courses that use gamification achieve good results. The completion percentage, which only counts first-try completions, is encouraging. In 2013, over 75% of the students completed both courses; for previous years, only 65% have completed in their first try. From student surveys and testimonials, we find completion percentages to be correlated with (although perhaps not caused by) increased student satisfaction, which many students agree is due to gamification. The high completion percentage addresses one of the major problems we have set to solve via gamification.
- For BSc-CO, In-class achievement (“Bonuses, In-class”) has increased every year, although the number of fresh students has not changed significantly. We attribute this to the social dynamic we have added to the course, where interaction and competition between students motivate them to return to class. This is further supported by the increasing number of students who complete in teams the social track (“Bonuses, Self-Study”).
- As indicated by the fraction of students who receive bonuses, the multiple paths of advancement already attract a significant fraction of our students. As new students get to learn course best-practices from the previous, and find class-time useful, we expect that participation will increase. For the first-year undergraduate course, we also conclude that more incentives are needed to attract the students.

We also find that:

- Based on the attendance to extra lectures, to which almost 100% of the students invited show up, we find that top students do like to learn for the sake of learning—they have high attendance even for lectures that do not impact the course grade.
- Mid-term performance characterizes well the top performers in the BSc-CO cohorts. Among the Top-20% at the end of the course, less 15% were not already part of the same top, mid-term. This allows for an early identification of achievers and winners.
- About 10–15% of the students fit the profile of winners or achievers. Less than 5% of the students will try to improve their grade if they have already passed the course in the first try; these correspond to achiever profiles.
- Unlike previous years, when failing students did not return to take the re-sit exam the same year after failing

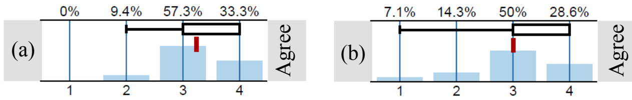

Figure 1: Official survey, BSc-CO: (a) "I have learned much from this course". (N=96) (b) "I found the content of this course interesting". (N=98)

for the first time, for the gamification-based `BSc-CO` over 90% of the students who failed tried again the same year; about 65% of them passed at their second attempt.

4.2 Student Satisfaction

We report on three types of student satisfaction evaluations, official surveys, volunteered testimonials, and in-class participation.

Our university runs detailed *official surveys* for every first-year course. Figure 1 depicts a selection of results for two of the most important questions in the survey, related to learning and interest in the course content. On a scale from 1 to 4, where 4 indicates complete approval and 1 complete disapproval (neutral opinion is not allowed), over 96 students (60-75% of the enrollment) have responded. For the two statements, the study indicates approval, with an average of 3.24 (standard deviation 0.61) and 3 (0.85), respectively.

In our own survey, run in-class and anonymized, we have asked students to evaluate various aspects of gamification. Surprisingly to us, more students thought more carefully about the course due to gamification (over 90%) than felt more motivated by gamification (only 50%–75%); in other words, even students who did not care about the particulars of the teaching technique were pushed by gamification to think more carefully about the advancement paths (the course design). We also find that over 90% of the respondents enjoyed the interactive part of the lectures and enjoyed the exercises at the end of the lectures.

We find that *In-class participation is high.* Lectures are often the bane of modern education, especially for large-enrollment classes. Participation in our lectures is voluntary and not rewarded, and videotapes of the lectures (from previous years) are available to students, yet students participate in large numbers to our In-class activity. Anecdotally, we observe 60–70% of our students in the classroom, even late in the course. For the extra lectures, we observe even higher levels of participation (as reported in Section 4.1). Also anecdotally, these are much higher numbers than the numbers regularly observed in the same faculty.

During and at the end of the course, we have received a large number of *Volunteered testimonials*, some via email. A few examples of the enthusiastic response of our students: "I just would like to say that the extra lectures are a great concept to reward the motivated students." "Lectures were nice, different in a good way from other teachers." and "I really like the rewards [during lectures]".

4.3 The Cost of Gamification

For the two courses we have explored in this work, we estimate the cost of gamification in time spent to create the gamification elements and to adapt the course content, which is in our experience:

- One week to consider gamification elements.
- One day per lecture for adaptation.
- Two hours per lecture, for analyzing end-lecture quizzes and adapting the next lecture accordingly.
- Two days after each major exam, to tabulate results.
- One week to compile final course results, using spreadsheet technology.

5. LESSONS LEARNED

We summarize in this section what we have learned as instructors and designers of two gamification-based courses.

5.1 Educator Experience

Gamification can be personally rewarding (see Section 5.2). In addition to rewarding interaction with the students, one of the authors of this article received, after students voted anonymously, a Teacher of the Year award. The reasons for this award include gamification.

Gamification can also be difficult on the instructor, in addition to the time overheads it incurs (see Section 4.3):

- There is a need for computer-assisted management of bonuses, etc., especially for large-enrollment courses.
- Introducing gamification can be difficult. Among the hurdles, we found: explaining a new system to students, conquering organizational inertia to get initial approval, limited university support before the approach became a success, etc.

From a technical perspective, we believe that:

- Offering points through announced tests leads to class rhythm. Unannounced formative tests, both within a lecture and across a course, may offer strong incentives for students to stay focused.
- Performance-related badges are very popular. Although negative badges exist in the world of gaming, for education we recommend that only positive badges should be used.
- Using short end-of-lecture quizzes and continuously asking for student feedback can provide meaningful information for gaming analytics.

5.2 A Testimonial

Among the testimonials, several have shown us that gamification can have an important impact on (student) lives. The following example is from a student who was not able to complete her B.Sc. studies for several years, in part because of failing to pass BSc-CO; she was helped by our alternative paths of advancement: `I want to thank you for showing that even though I'm not that good at written exams, I still can excel at other points in my study. I'd love to have a copy of my badge, as physical reminder of a course that made me eager to learn about things. Even when some of those things will never really have my interest. This course, and the way it was given, learned me a few things about what motivates me, and only for that reason it was totally worth getting up for every lecture.`

6. RELATED WORK

We discuss in this section related work on instructional methodology and on curriculum content.

Methodologically, our work is one of the first to assess gamification in university settings, and the first long-term study of gamification effects in undergraduate and graduate *courses*. In a recent study [7], gamification was used to present orientation information to university students.

Content-wise, ours is one of the first long-term studies on how to mesh advanced PDC topics (GPUs, cloud computing) into first-year undergraduate courses. Our two years of practice could contribute to the debate on this topic [5]. This work also complements previous studies on the use of programming models for advanced distributed systems, in particular MapReduce [9, 16]. In contrast to these studies, which focus on enabling cloud infrastructure for students, ours focuses on diversifying the content paths within the topic of (distributed) computer systems.

7. CONCLUSION AND ONGOING WORK

Responding to an increasing need to find new techniques for teaching academic-level computer science, we have proposed in this work to use gamification to improve student participation and success in technical higher education.

Drawing from various game studies, theory of fun, and education methodology, we have designed a toolbox for course gamification, which adapts to different types of students via a set of mechanisms and dynamics. Using this toolbox, we have designed two gamified courses, one at undergraduate and one at graduate levels. Key features of our design include enabling various paths of advancement, and fostering social interaction inside and outside the classroom. Our empirical evaluation reports on three years of experience with the undergraduate and one year with the graduate courses, during which we have educated over 450 students. We have found that gamification can help in many ways our students, from increasing passing rates and participation, to high student satisfaction and heart-warming testimonials.

We are currently extending this work via additional surveys. We have collected already over 300 new, more in-depth responses, and are currently analyzing them. We are also trying to help others apply the techniques described in this work, for (semi-)independent validation. We also plan to investigate far-reaching questions about gamification: How to enable longitudinal studies of gamification? How to design a prototypical gamification-based course in computer science? Which type of instructional goal gains most from gamification? Which type of student gains most from gamification? Which gamification element is responsible for the largest improvement? etc.

8. ACKNOWLEDGMENTS

We would like to thank our system administrators, teaching assistants, and student assistants for helping with these two courses. This project was partially funded by a TU Delft Talent Grant, by the NWO/STW Veni grant large (11881), by the EU projects PEDCA (FP7 #320013) and EYE (FP7 #619241), and by the COMMIT.

9. REFERENCES

[1] ACM/IEEE-CS Joint Task Force. Computer Science Curricula 2013. Online reports, Strawman and Ironman., Feb 2012/2013.

[2] M. Armbrust et al. A view of cloud computing. *Commun. ACM*, 53(4):50–58, 2010.

[3] R. Bartle. *Designing Virtual Worlds*. New Riders Games, 2003.

[4] R. Brewer, L. Anthony, Q. Brown, G. Irwin, J. Nias, and B. Tate. Using gamification to motivate children to complete empirical studies in lab environments. In *IDC*, pages 388–391, 2013.

[5] R. A. Brown, J. C. Adams, D. P. Bunde, J. Mache, and E. Shoop. Strategies for adding the emerging pdc curriculum recommendations into cs courses. In *SIGCSE*, pages 109–110, 2013.

[6] S. Deterding, D. Dixon, R. Khaled, and L. Nacke. From game design elements to gamefulness: defining "gamification". In *MindTrek*, pages 9–15, 2011.

[7] Z. Fitz-Walter, D. Tjondronegoro, and P. Wyeth. Orientation passport: using gamification to engage university students. In *OZCHI*, pages 122–125, 2011.

[8] D. R. Flatla, C. Gutwin, L. E. Nacke, S. Bateman, and R. L. Mandryk. Calibration games: making calibration tasks enjoyable by adding motivating game elements. In *UIST*, pages 403–412, 2011.

[9] P. Garrity, T. Yates, R. A. Brown, and E. Shoop. Webmapreduce: an accessible and adaptable tool for teaching map-reduce computing. In *SIGCSE*, pages 183–188, 2011.

[10] S. Hull. 20 obstacles to scalability. *ACM Queue*, 11(7):20, 2013.

[11] K. Huotari and J. Hamari. Defining gamification: a service marketing perspective. In *MindTrek*, pages 17–22, 2012.

[12] Jim Gray. Twelve computer science challenges, 1998. ACM A. M. Turing Award Lecture.

[13] R. Koster. *Theory of Fun for Game Design*. Paraglyph Press, kindle edition, 2010.

[14] T. W. Malone. Heuristics for designing enjoyable user interfaces: Lessons from computer games. In *CHI*, pages 63–68, 1982.

[15] M. J. Nelson. Soviet and american precursors to the gamification of work. In *MindTrek*, pages 23–26, 2012.

[16] A. Rabkin, C. Reiss, R. H. Katz, and D. A. Patterson. Using clouds for mapreduce measurement assignments. *TOCE*, 13(1):2, 2013.

[17] G. Zichermann and C. Cunningham. *Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps*. O'Reilly Media, 2011.