On the Role of Gamification and Localization in an Open Online Learning Environment: Javala Experiences

Timo Lehtonen and Timo Aho
Solita Plc.
Åkerlundinkatu 11
FI-33100 Tampere, Finland
firstname.lastname@solita.fi

Essi Isohanni and Tommi Mikkonen
Tampere University of Technology
Korkeakoulunkatu 1
FI-33720 Tampere, Finland
firstname.lastname@tut.fi

ABSTRACT

Massive Open Online Courses (MOOCs) have rapidly become an important tool for educational institutes in teaching programming. Nevertheless, high drop-out rates have always been a problem in online learning. As MOOCs have become an important part of modern education, reducing the drop-out rate has become a more and more relevant research problem. This work studies a nine-year-long period of maintaining an open, online learning environment of programming. The aim is to find out how the implementation of the learning environment could engage the students to learning and this way affect the drop-out rate. We provide an insight to experiences stemming from nine years of data collected with Javala, an online system created to help shifting from C++ to Java programming. The paper also discusses two key properties of Javala, gamification, and localization, together with data to assess their significance.

CCS Concepts

•**Applied computing** → **E-learning;** *Interactive learning environments; Collaborative learning;*

Keywords

Online programming courses, MOOCs, learning analytics, educational data mining, learning outcomes, learner profiles, localization, gamification

1. INTRODUCTION

With open online courses, learners can participate in classes and complete exercises on their own terms, and learning results can be tracked by the educators with ease. Moreover, these systems support both individual learners as well as institutional participants, who take part in courses because it has been listed as a part of a curricula. In fact, at times it can be difficult to differentiate between these two groups, in particular if institutional use is not based on eventual certificates but simply to promote learning at students' own time.

Koli Calling 2015, November 19 - 22, 2015, Koli, Finland

ISBN 978-1-4503-4020-5/15/11…$15.00
DOI: http://dx.doi.org/10.1145/2828959.2828973

Despite their recent immense popularity, characteristics of successful open online courses are largely unknown. Data regarding creation, participation, and use of such courses is available, but there is little long-term data regarding learner profiles that participate in online courses. Even less appears to be known about learning outcomes among the learners.

In this paper, we provide an insight to experiences stemming from nine years of data collected with Javala, an online system created at <withheld for anonymity> to help the students to shift from C++, the language used in elementary programming courses, to using Java, which was expected to be familiar to students in later courses. In particular, we will focus on two key properties of Javala, gamification and localization, which we believe were genuinely unique at the time of the introduction of the system in 2004. The exact research question we wish to study is *how usage patterns of the system changed when gamification or localization was either enabled or disabled*. In addition, we analyze learner data to assess the significance of these two aspects. Towards the end of the paper, we provide a concluding discussion, where we go through the lessons we have learned and provide some potential directions for future work.

The rest of this paper is structured as follows. In Section 2, we discuss the background of this paper, and in Section 3 we introduce Javala, our online learning environment, which has been in open use since its creation in 2004. In Section 4, we perform a statistical analysis on gathered user data, and in Section 5 we list our main learnings, address limitations of our study, and point out some directions for future work. In Section 6, we draw some final conclusions.

2. BACKGROUND

2.1 MOOCs

Massive Open Online Courses (MOOCs) have rapidly become an important tool for educational institutes in teaching programming. MOOCs offer an opportunity for students to participate in courses worldwide. They have gained huge popularity during last few years. The first MOOCs were offered by Coursera and Udacity [12], and since then numerous others have emerged.

Low engagement is characteristic to MOOCs. While some MOOCs, such as classes from Stanford University, edX, Coursera, and Udacity have attracted tens of thousands of participants [15], completion rates below 10% are typical, although some courses — for example on functional programming — have gained nearly a 20% rate of completion [10]. Another example is a course on circuits and electronics [1], which had 155 000 participants out of which 7 100 (5%) passed.

The ability of MOOCs to generate a tremendous amount of data opens up new opportunities for research [1]. For example, a MOOC on machine learning, with its over 40 000 student code submissions, contains a lot of information on how participants study and learn in this kind of open and massive environment [8]. Analysis through this kind of educational data on different learning styles on MOOCs brings the pedagogical models to the focus and creates suggestions on how different learning styles should be supported [6].

2.2 Gamification

During the last few years, gamification has been an element that has been successfully used in many web based businesses to increase user engagement [4]. A common definition for gamification is "the use of game design elements in non game contexts" [3]. Gamification can usually be seen as a set of elements like points, badges, levels and leaderboards [3]. The most elementary gamification element consists of a rewarding mechanism that rewards people in response of accomplishment of certain activities [5].

A literature review by Hamari et al. [7] with a high-level research question *Does gamification work?* examined altogether 24 empirical studies. The motivational affordances tested in this research were points, leaderboards, achievements/badges, levels, story/theme, clear goals, feedback, rewards, progress and challenge. According to majority of reviewed studies, gamification does produce positive effects and benefits.

There is an increasing number of online services that are focused on adding a gamified layer to the core activity. Khan Academy as described in article *One man, one computer, 10 million students: how Khan Academy is reinventing education* [11] has included also gamification elements and achieved a huge success worldwide. Codecademy [14] that initially covered only JavaScript was launched in October 2011 [12]. As an interactive online programming environment, it covers many topics, for example regarding the Python programming language.

Pex4Fun from Microsoft Research [13] brings programming with to a user's web browser in a game-like way. The user can write, compile, and run code in order to learn programming concepts, practice coding skills, and analyze the behavior of code interactively. The system was released in June 2010 and it collected one million attempts in two years [14]. Pex4Fun includes "coding duels" which are games within the platform. In a coding duel, a player's task is to implement a method that has similar functionality as a secret method behind the scenes. When a new user enters Pex4Fun, the threshold to start programming is low as the starting screen of the system contains a code snippet to start with.

2.3 Localization

Localization is a practice where computer software, tools and services are adapted to different languages, regional differences, and technical requirements of a target market. Often associated with internationalization — the practice of designing software so that it can be adapted to different regions and languages without engineering changes — localization is the process of adapting internationalized software for a specific region or language. Commonly needed actions include the introduction of by adding components and translating text, often by using parameters, for instance.

For end users, the benefits of localization are many. To begin with, in many cases services are simply easier to consume, when they are localized. This is the case even when the users could use a unified, international version of the tool. Moreover, services can also be made available for larger user group, where language skills might be a burden otherwise, especially when special vocabulary is needed for successful communication. We believe that the latter is particularly important in educational setting, where the translation takes a part of the focus from the learner.

3. JAVALA LEARNING ENVIRONMENT

3.1 Description of the Tool

Javala was an open learning environment that was published in September 2004 [9]. At that time, Java version 1.4 was the official version and thus later additions, like generics, were not covered in the content of Javala. The programming language used on the CS1 courses at <withheld for anonymity> was C++, but still some courses stated Java as a required preliminary skill for the course. There were no official courses available on Java programming at that time, and the main purpose of Javala was to help students in shifting from C++ to Java. Later on, during 2007, also an English version of Javala was introduced and released. The system was run successfully without human interaction for over nine years, at which point changes in curricula made the need for such transitional exercises less acute. The final blow for shutting Javala down took place in October 2013, when security support for the OS version Javala was running on came to an end.

From the very beginning, Javala was built to be open, which in this context means free, non-commercial and easily available (in seconds) anywhere, anytime. When a new student entered Javala, the only thing needed was to give a nickname and then start coding. No registration was needed, but later the user could register the nickname if he or she wanted to. In this sense, Javala was similar to Pex4Fun [13] mentioned earlier.

There were altogether 15 categories of theory material which contained the total number of 41 exercises. The student first read the theory part as shown in Figure 1, and then opened the embedded exercise to a new pop-up window shown in Figure 2. The exercise window contained Java source code, which was partly hard coded and partly modifiable. The student modified the source and pressed the "execute"-button. The system then compiled and executed the code on the server side, and gave feedback in the same pop-up window. A correct solution produced some points for the user.

In addition to points as such, badges were achieved by reaching certain amount of points. The user first started as a "Java Tourist" and advanced through a few levels to "Java King". The advancements of users were also announced publicly to other users on the "Javala Happenings" board, which was shown constantly in the user interface.

There was also a Top 100-board that listed best users of the day, of the week and of all-time. When a new user entered Javala, and when the user had solved all exercises, the nickname of the user was shown as the best user of all time users until another user came and solved all the exercises.

Figure 1: The layout of a Javala page – theory and exercises embedded. The current user has zero points and is thus a "Java Tourist".

3.2 Example Sessions

Several typical usage patterns can be identified in the data. In the following, we present some of them to demonstrate with examples how learners used Javala.

A sample session of a user that spent 13 minutes to solve an exercise related to converting a string to lower case is shown in the following. The correct solution to the exercise was simply `return s.toLowerCase();`.

```
Execution #1:
> toLowerCase(s);
> return s;

Execution #17:
> s = "SAABASDF";
> s = s.toLowerCase();
> return s;

Execution #27:
> s = s.toLowerCase
> return s;

Execution #37:
> s = s.toLowerCase();
> return s;
```

The user executed code altogether 37 times. It is noteworthy that the purpose of this exercise was just to demonstrate how to call a method of an object and thus no null comparison was required. The user first executed a solution that is syntactically very near to the correct solution. The user then ended up to solution #17. Then, some seven months later, the user came back and tried 11 different solutions with a correct one on execution #37. In this case, the same user spent over 2.5 hours trying to solve exercise `StringPalindrome`, with the obvious goal, but failed to solve it.

A sample session of another user solving exercise `StaticMethod` is shown below[1]. This user represents the group of users that

[1]Indentation has been modified for this paper to represent the

Figure 2: A sample exercise. When the user solves the exercise, one point is achieved.

seems to disappear when gamification is turned off.

```
4:18:15 PM:
<<< The user takes a look at the exercise. >>>

4:24:58 PM:
public static boolean isLicenseNumber(String s) {
  if (s.length == 7 && charAt(3) == "-")  {
    return true;
  }
  return false;
  }
}
<<< 16 other solutions that do not compile >>>

4:41:21 PM:
// isLicenseNumber(String s)
public static boolean isLicenseNumber(String s){
  if (s.length() == 7 && s.charAt(3) == "-"){
    return true;
  }
 return false;
}

4:43:06 PM:
<<< The user takes a look at another  >>>
<<< exercise "StringPalindrome"       >>>

4:50:07 PM:
// isLicenseNumber(String s)
public static boolean isLicenseNumber(String s){
  if (s.length() == 7 &&
      String.valueOf(s.charAt(3)) == "-"){
    return true;
  }
  return false;
```

code adequately.

```
}
<<< 35 solutions; most compile, but are wrong >>>
5:11:53 PM:
// isLicenseNumber(String s)
public static boolean isLicenseNumber(String s){
  if (( s != null ) &&
      ( s.length() == 7 ) &&
      ( "-".equals(String.valueOf(s.charAt(3))) ))
  {
    return true;
  }
  return false;
}
<<< The exercise was solved correctly >>>
```

The user had a total session length in Javala of 4 hours and 22 minutes. In the above exercise, the user spends 34 minutes with a five minute threshold to solve the exercise. The actual calendar time spent is approximately 50 minutes, but because of breaks and other browsing activity is not counted to the session length, the effective session length is shorter. It is noteworthy that the first execution that compiles is with timestamp 4:50:07, after half an hour of coding. The earlier executions did not compile because of incorrect way of comparing a string to a character. The user takes a look at other exercises related to string comparison — in this case, exercise `StringPalindrome` — and perhaps the breaks in the session are used for searching help from other sources.

An expert user solves exercise `StaticMethod` in 90 seconds. The sample session shown below[2].

```
2:11:36 PM:
<<< The user takes a look at the exercise >>>

2:12:36 PM:
public static isLicenseNumber(String s)
  if (s.length() != 7) return false;
  if (s.charAt(3) != '-') return false;
  return true;
}

2:12:42 PM:
public static isLicenseNumber(String s) {
  if (s.length() != 7) return false;
  if (s.charAt(3) != '-') return false;
  return true;
}

2:12:49 PM:
public static boolean isLicenseNumber(String s) {
  if (s.length() != 7) return false;
  if (s.charAt(3) != '-') return false;
  return true;
}

2:13:03 PM:
public static boolean isLicenseNumber(String s) {
  if (s == null) return false;
  if (s.length() != 7) return false;
  if (s.charAt(3) != '-') return false;
```

[2]Again, indentation has been modified for this paper to represent the code adequately.

```
    return true;
}
<<< The exercise was solved correctly >>>
```

First, the expert user opens the exercise and consumes 60 seconds on formulating a solution that is almost right but does not compile because of a missing curly brace. The user then targets to a compiling solution and finds one in 13 seconds. The last thing missing in the second last solution is a null comparison and the exercise was solved in 90 seconds. The user was shortly interviewed after completing all exercises. He said that his strategy was to solve all exercises by executing code as much as possible even without really reading the exercise descriptions. Anyway, the first task of this single expert user was to formulate a rather good solution with only a missing curly brace and null comparison as a first solution, so he also spent time thinking before executing code. The last two executions are a good example of effectively using the system as a helper in testing. If the user would have thought a bit more, the second last execution would have contained the null comparison. Anyway, as the user said, he took the benefit of the system as a helper and did not invest thinking capacity to null comparison before it was actually needed.

3.3 Role of Gamification

Gamification was an important part of Javala. Achieving points and badges for the own nickname and getting the advancement publicly announced gave more motivation for learning to program in Java.

We tested the role of gamification by removing every gamification element from the system for a three month period in the beginning of year 2013. The gamification elements consisted of points, badges, achievement announcements and a list of top users. A statistical summary of the importance of gamification is described in detail in Section IV.

3.4 Role of Localization

Javala was originally created for a limited set of users, although we never restricted participation in any way. However, since our primary goal was to support the transition of Finnish students from C++ to Java, we originally introduced only Finnish version of the system. Nevertheless, we bore in mind though that over time we would also need an English version to support student exchange and visiting international students as well. Consequently, we ended up creating a localized version of Javala, with particular attention invested in the localization aspects.

While partly accidental, we feel that the ability to create localized learning experiences for students increases user experience for both the educator and the learner. Consequently, both the educator and the student can promote the use of the system more easily on their own language. Based on the data that has been gathered on usage of the system, it can be deduced that in Finland there has been a lot of institutional use of Javala, including for instance high schools where using Finnish can be considered an asset or even a requirement. In contrast, the English version has been used extensively from other countries (e.g. Romania) as well as continents (e.g. Asia), but apparently mostly by individuals.

4. JAVALA IN NUMBERS

4.1 Use Statistics

There were 34 124 logged in users in Javala during the nine year period. A total number of 13 577 users (40%) did not complete any exercise. This number may also include users that are web crawlers or bots. Furthermore, as the login procedure was to simply enter a nickname, one person may have used more than one nickname.

The user logs for the nine-year period were analyzed using R statistical computing environment[3]. The nicknames of the users are hidden in this paper because of privacy concerns. We instead identify the users with a username with a running sequence number. The sequence number started from 10000 for the Finnish users and 80000 for the English users.

During the nine-year period a single exercise was executed 1 006 835 times. The code worked correctly in 197 618 executions. Compiler errors occurred 463 155 times and the code worked incorrectly 268 623 times. Rest of the solutions ended up to exceptions or to an endless loop. Logged in users also took a look at Top 100 list 24 691 times in total.

Figure 3 shows the number of new users that started the usage of Javala (approximately from 500 to 2 500 users per quarter of a year). Some seasonal change seems to occur as the sums for different quarters are 11 469 (Jan-Mar), 6 100 (Apr-Jun), 7 114 (Jul-Sept), and 9 336 (Oct-Dec). Apparently, the beginning of the year was the most popular arrival time for new users.

When the English version of Javala was published on q1 of 2007, there was a total number of 2 400 logged in new users starting the usage of Javala. Of them, only 147 users (6%) completed at least half of the exercises which accompanies the high drop-out rates of MOOCs [1].

Since the publication of the English version of Javala, an exercise was run altogether 589 254 times (215 918 times or 37% for the English and 373 336 times or 63% for the Finnish).

4.2 Geolocation of the Users

The IP addresses of each request was logged to a separate log file since the beginning of year 2007. Unfortunately, there was an error in the logger settings and parts of the data were overwritten randomly per day. The number of run exercise requests collected with the broken log settings was 66 018 requests while the real number of run exercise requests during the years 2007-2013 was 589 439. Thus, only around 10% of the IP addresses of the requests were captured correctly. Moreover, because the domain names were resolved in 2014, some IP addresses may point to another domain name than before. However, because parts of the log were lost randomly, it can be assumed that the captured data represents the actual data adequately well. Thus, the partly saved data can be used to draw some guidelines of the geolocation properties of use.

The three most popular top level domains that Javala was used from were the Finnish telecommunication operators' dynamic IP addresses. Thus, Javala was apparently mostly used from home computers or mobile devices in Finland. The first ten recognizable domains or telecom operators in the list were a Finnish high school, a university and its cam-

[3] http://www.r-project.org

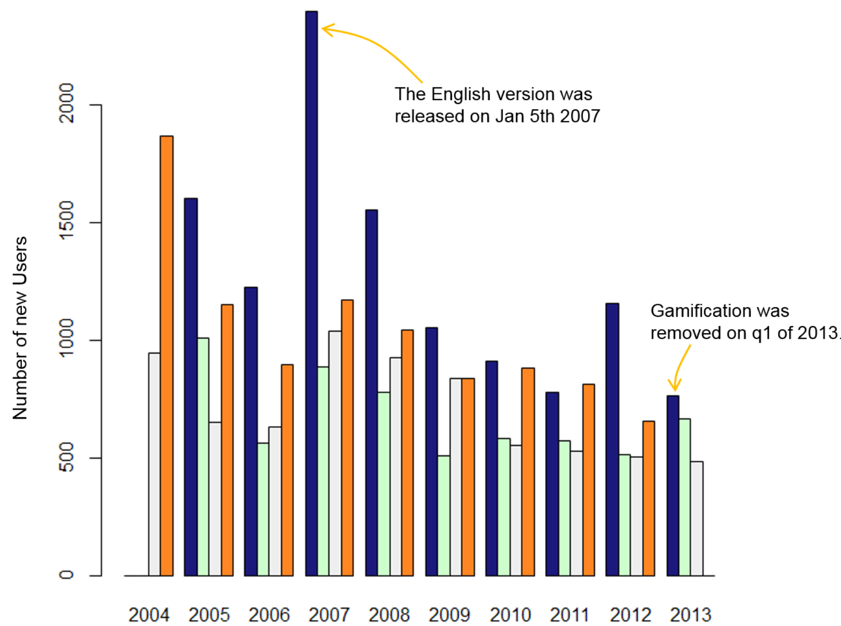


Figure 3: Number of new users per quarter of a year.

pus area, a large Finnish ICT company, two other Finnish universities, and telecom operators from Germany, Spain, Mexico, Japan and Romania. It is noteworthy that the location where Javala was used most was a high school in Western Finland, not the institute where the system originated from.

4.3 Sessions of Usage

The threshold used for measuring the session length was five minutes. If the user was inactive for more than five minutes, the session was interpreted to be ended. For example, if the log contains events at 12:00:00, 12:05:00, 12:10:00 and 12:15:00, the session length is 15 minutes. However, such an interpretation on session length may lead to shorter total session lengths than the length apparently really is. For instance, if the log contains events at 12:00:00, 12:03:55, 12:10:00, 12:10:04 and 12:20:00, the session length is interpreted to be only 3:59, because there is no evidence that the user actually used the system between the breaks longer than five minutes.

There were 5 754 users (17%) that spent at least 45 minutes in Javala during the nine-year period. They spent altogether 16 461 hours in Javala. The number of users that spent more than 7.5 hours (a working day) was 285. The four longest total session lengths in Javala were 67, 44, 27 and 26 hours and the total session lengths seem to fit the exponential distribution.

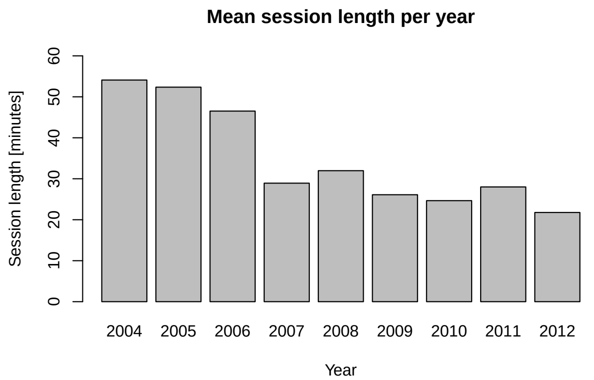
The shortest time to complete all exercises in Javala was 33 minutes. The best 2% of users (12 users with shortest total session length) solved all exercises in less that 56 minutes. It is worth noting that user may have practiced the exercises with one nickname and then completed them quickly with another nickname. All except two of the top 12 users used the English version of Javala. One explanation to this is that many technically oriented people in Finland may have English language in their browser settings as the default language and in such case, the English version of Javala is shown to the user first when entering the site. One more fact of one of the top 10 users is known. One of the authors worked in a Finnish ICT company where he organized a Javala coding competition in Fall 2011. The event was advertised in advance as a Java coding competition and then the task was then to solve all exercises in Javala as quickly as possible. The user was a skilled programmer, who solved all exercises in 49 minutes. He was a Finnish person who used the English version because of the default language setting of the browser.

Next, we filtered a set of users called learners to get a realistic picture of the data. The learners had a total session length between 45 minutes and 15 hours. Thus, 28 370 inactive users and 27 outliers with a total session length more than 15 hours were filtered out. The total number of learners was 5 727 users. Their average total session length was 2 hours 45 minutes while the median was 2 hours. The average time spent to solve the `StringToLower` exercise was 6 minutes. For `StringPalindrome` it was 16 minutes and for `StaticMethod` it was 24 minutes. In average, they completed 20 out of 41 exercises while the median was 16. The mean total time for solving all exercises was 4 hours. The number of new users during the first quarter of years 2010-2013 was 914, 782, 1159 and 768 accordingly, while number of learners was 114, 118, 158 and 56 accordingly. Removing gamification seems to approximately halve the number of learners.

4.4 Evolution of Usage

The data set covers a nine-year long period. Figure 4 illustrates the long term and seasonal changes in the data. In the figure, a separate user was recognized by a nickname that completed at least a single exercise.

There was clear evolution in the usage of the system during the nine-year period of being online in figure 4. For



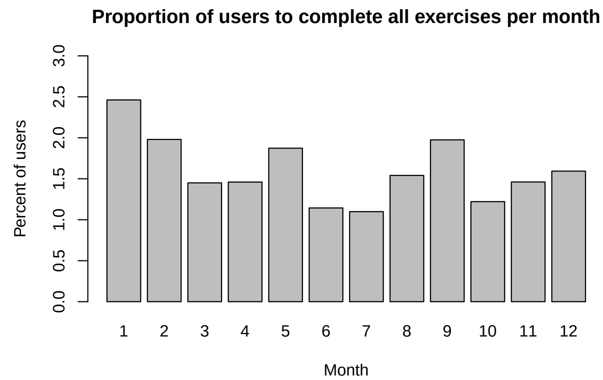


Figure 4: Long term and seasonal changes in the data.

Table 1: Number of users per session length

	1 exercise	60 min	120 min	240 min
2010/Q1 (on)	526	98	45	17
2010/Q2 (on)	322	53	33	12
2011/Q1 (on)	488	100	65	31
2011/Q2 (on)	352	81	47	17
2012/Q1 (on)	649	132	77	16
2012/Q2 (on)	326	53	36	12
2013/Q1 (off)	438	45	19	8
2013/Q2 (on)	463	69	45	14

Table 2: Gamification and usage

Gamification	On	Off
N	5 771	438
Means:		
Total time (min)	41	26
Number of exercises solved	5	3
Standard deviations:		
Total time (min)	90	57
Number of exercises solved	9	6

example, the mean session length lowers during years from over 40 minutes during years 2004-2006 to 22 minutes in year 2012. Also the proportion of users to complete all exercises during different months seems to variate. The users that started using Javala in January (2.5%) or September (2%), seem to complete all exercise more likely. During summer months, the eagerness to complete all exercises is lower (1%). This is important to note in the analysis of the next subsections as we compare users that started using Javala during last few years during certain quarters of a year.

4.5 Role of Gamification

To analyze the role of gamification, we conducted statistical tests for the numerical data collected in Javala. Because of the evolution of the data, it is not reasonable to test the removal of gamification effect in the beginning of the 2013 to all the previous data. Instead we compared the time interval to the same time intervals in the previous years.

Figure 5 presents the proportions of users per session length during the first and second quarters of a period from year 2010 to 2013. The gray curves present the users per session length during years 2010-2012. The red curve presents the users while gamification was turned off during quarter 1 of year 2013. The green curve presents quarter 2 of 2013 when gamification was turned back on. While gamification is off, the proportion of users that have session length of 60 to 180 minutes seems to be remarkably lower. As the session lengths seem to be exponentially distributed, we applied the Kolmogorov-Smirnov test to test if the distribution of session lengths differs statistically significantly. A p-value<0.001 denotes that the difference between the gamificated and non-gamificated session lengths is statistically significant.

Table 1 presents the number of users per session length for each quarter of a year. The first column presents the number of users that entered Javala and completed at least a single exercise. When gamification was turned off during Q1 of 2013 (red curve in figure 5), the number of users with at least a 120 minute long session lowers to 19 users while the mean of number of users during other quarters is 50 users. When gamification was turned back on in Q2 of 2013 (green curve in figure 5), the amount of users with at least a 120 minute long session was restored to 45 users. When the session length is four hours or longer, there seems to be no remarkable difference in the number of users. We applied the Chi-squared test of independence to the number of users with different session lengths. The test with p-value<0.001 indicates that session lengths of 60, 120 and 180 minutes are statistically significantly different between the periods while gamification was enabled/disabled, but the number of users with a session length of 240 minutes is not. A visual perception in figure 5 supports this finding.

Next, we investigated the mean session lengths and number of exercises solved when gamification was enabled or disabled during years 2010-2013. Table 2 presents the means of total session length and the number of exercises solved for the users who used Javala with the gamification features and the users who did not have the gamification in Javala. When the gamification features were on, the users spent more time there and completed more exercises.

Because the data seems to be not normally but possibly exponentially distributed, we used the non-parametric Mann-Whitney U test for analyzing the statistical differ-

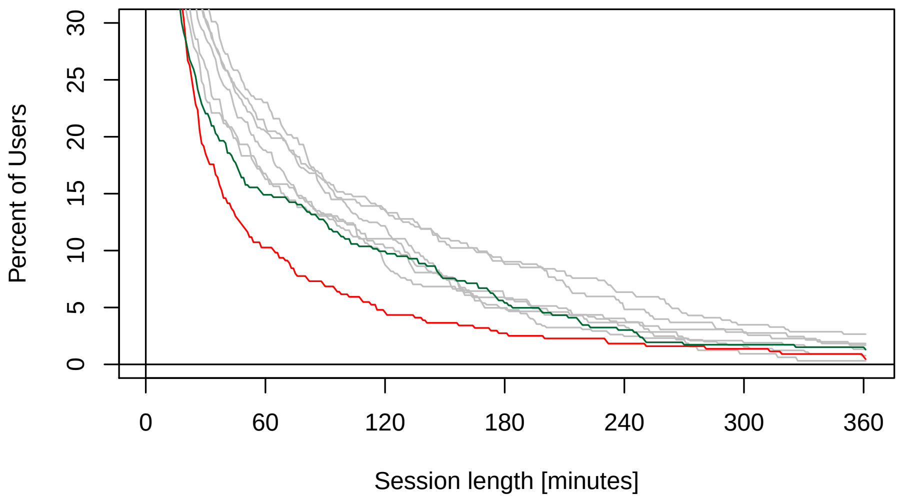


Figure 5: Proportion of users per session length in minutes.

ences in the data as it does not make any assumption on the underlying distribution of the data. The total time usage was statistically significantly smaller with gamification off (with p-value<0.01). Also the number of solved exercises was statistically highly significantly smaller with p-value<0.001. Moreover, When gamification was on, a total number of 579 (2%) of users completed all exercises. Amount of users to complete all exercises during the first quarters of 2010, 2011 and 2012 were 11, 21 and 16. When gamification was turned off, no user completed all exercises.

Finally, we investigated the proportion of long and immersed usage sessions. We measured the proportion of users that had a continuous session that lasted over one hour without breaks no longer than five minutes. The proportion of users that had a long and immersed usage session, dropped from 2.5% to 1.0%.

4.6 Role of Localization

Similarly, we investigated the localization of Javala. Table 3 presents the same variables as Table 2, but the grouping is done according to the language version of Javala during years 2007-2013 when both versions were available. The users of the Finnish version of Javala spent more time and completed more exercises than the users of the English version. The total time usage and number of solved exercises were both statistically significantly smaller with English version (with p-value<0.001).

Table 3: Localization and usage

Language version	Finnish	English
N	6 328	7 200
Means:		
Total time (min)	56	36
Number of exercises solved	7	6
Standard deviations:		
Total time (min)	117	82
Number of exercises solved	10	10

5. DISCUSSION

5.1 Analysis

The users who had the gamification features in Javala used Javala for a longer time in average and completed more exercises in average than the users who did not have the gamification. This difference was statistically significant, and it shows that adding gamification in a learning environment plays an important role in keeping the students engaged with the learning environment. This supports the conclusion of [7] that gamification does produce positive effects and benefits also in MOOCs. Moreover, the proportion of users that were immersed to use Javala continuously for a period of longer than one hour without breaks longer than five minutes, was over two times higher when gamification was on. The flow by Csikszentmihaly [2] "is a state of peak enjoyment, energetic focus, and creative concentration experienced by people engaged in adult play". A usage session lasting over one hour could be interpreted as a flow state where the user is fully immersed to programming activities.

In the same way, there was a statistically significant difference: the users of the Finnish version used the system for a longer time and completed more exercises in average than the users of the English version. The reasons, why the Finnish version was more popular than the English one, are not known exactly. One reason may be that a localized Finnish version of a learning environment is easier to approach. A foreign language may be a burden especially for younger students. A Finnish version for Finnish people may also feel safe and it may introduce positive feelings. Another reason may be that the visibility of the Finnish version of Javala in search engines was good because of the Finnish language.

Because starting using Javala was so easy, it may have raised the dropout rate, too. The procedure for login was very simple: the user just entered a nickname and then started coding the exercises. This produced a large number of users that entered Javala just for curiosity but completed no exercises.

It was also possible to register the nickname with password protection at any time, but this was totally optional. If the nickname was not registered, anyone could reuse it accidentally by entering the same nickname. When a user entered a nickname that was already in use and which did not have password protection, the system showed a notification that the nickname was already in use. Some users may have ignored this notification which leads to a bias in the user data where two totally different real world persons may have merged into one Javala user.

5.2 Limitations

The test for gamification lasted for only a limited time of three months; however we believe this is extensive enough and therefore only a small threat to the validity of this research.

There were only two localizations, with one being truly local and other more or less international. The versions were not introduced at the same time, but they were running simultaneously for six years, so the comparison of the different language versions is reasonable.

Experiments regarding gamification were based on interleaved, not parallel, use of the system. A/B testing would have been a better method for real testing, although it would have its own limitations, e.g., the problem that users in a same physical classroom could encounter different kind of versions of the system, which could cause confusion and affect the results gathered with this testing strategy.

5.3 Future Work

A future version of Javala should include better possibilities for interaction between the novice and expert users. One possibility could be to include gamification elements in the process when an expert helps a novice and grant some points or badges to a student that helps another student.

A more advanced learning environment should also give better feedback on the correct solution the user manages to construct. For example, if the solution differs from all the solutions made by the expert users, the system could give feedback to the user that the solution is exotic, because it differs from the mainstream solutions.

One important new feature for this kind of learning environment would be to enable the user to get help in a situation where the user gets stuck on some irrelevant problem for a long time. The system could help the user to advance in this kind of hard situations. This way the quality of learning or the ratio between time invested and new skills learned could be better.

6. CONCLUSIONS

In this paper we presented the long-term statistical data of usage of an open online course. We showed there was a statistically significant difference between the users who used the gamificated version and the non gamificated version of the system. This supports the argument in other studies that gamification does work, also in context of open, online courses. Gamification improves the usage of the learning environment and incites the users to strain to the limits of their skills more eagerly, which may lead to learning more about the subject. Localization is another important aspect. Localizing the system to the users own locale also has a positive impact on the eagerness of the user to use the learning environment.

A smooth user experience is also essential. A system that can be used anonymously and concentrates to the subject itself during the first seconds of usage is safe and also easy for the teacher to embed into other teaching activities.

Gamification, localization, and an easy but safe way of embedding the course to other contexts forms a cost efficient way to generate thousands of days of learning possibilities for everyone.

7. REFERENCES

[1] L. Breslow, D. E. Pritchard, J. DeBoer, G. S. Stump, A. D. Ho, and D. Seaton. Studying learning in the worldwide classroom: Research into edxâĂŹs first mooc. *Research & Practice in Assessment*, 8:13–25, 2013.

[2] M. Csikszentmihalyi. *Beyond boredom and anxiety.* Jossey-Bass, 2000.

[3] S. Deterding, D. Dixon, R. Khaled, and L. Nacke. From game design elements to gamefulness: Defining gamification. In *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*, pages 9–15. ACM, 2011.

[4] A. Domínguez, J. Saenz-de Navarrete, L. De-Marcos, L. Fernández-Sanz, C. Pagés, and J.-J. Martínez-Herráiz. Gamifying learning experiences: Practical implications and outcomes. *Computers & Education*, 63:380–392, 2013.

[5] D. J. Dubois and G. Tamburrelli. Understanding gamification mechanisms for software development. In *Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering*, pages 659–662. ACM, 2013.

[6] F. Grünewald, C. Meinel, M. Totschnig, and C. Willems. Designing moocs for the support of multiple learning styles. In *Scaling up Learning for Sustained Impact*, pages 371–382. Springer, 2013.

[7] J. Hamari, J. Koivisto, and H. Sarsa. Does gamification work?–a literature review of empirical studies on gamification. In *System Sciences (HICSS), 2014 47th Hawaii International Conference on*, pages 3025–3034. IEEE, 2014.

[8] J. Huang, C. Piech, A. Nguyen, and L. Guibas. Syntactic and functional variability of a million code submissions in a machine learning mooc. In *AIED 2013 Workshops Proceedings Volume*, page 25. Citeseer, 2013.

[9] T. Lehtonen. Javala–addictive e-learning of the java programming language. In *Proceedings of Kolin Kolistelut/Koli Calling–Fifth Annual Baltic Conference on Computer Science Education. Joensuu, Finland*, pages 41–48, 2005.

[10] H. Miller, P. Haller, L. Rytz, and M. Odersky. Functional programming for all! scaling a mooc for students and professionals alike. In *ICSE Companion*, pages 256–263, 2014.

[11] M. Noer. One man, one computer, 10 million students: how khan academy is reinventing education. *Forbes*, pages 398–407, 2012.

[12] D. Pritchard and T. Vasiga. Cs circles: an in-browser python course for beginners. In *Proceeding of the 44th ACM technical symposium on Computer science education*, pages 591–596. ACM, 2013.

[13] N. Tillmann, J. de Halleux, and T. Xie. Pex for fun: Engineering an automated testing tool for serious games in computer science. Technical report, MSR-TR-2011-41, March, 2011.

[14] N. Tillmann, J. De Halleux, T. Xie, S. Gulwani, and J. Bishop. Teaching and learning programming and software engineering via interactive gaming. In *Software Engineering (ICSE), 2013 35th International Conference on*, pages 1117–1126. IEEE, 2013.

[15] A. Vihavainen, M. Luukkainen, and J. Kurhila. Multi-faceted support for mooc in programming. In *Proceedings of the 13th annual conference on Information technology education*, pages 171–176. ACM, 2012.