A Study on the Use of Gameful Approaches in Self-paced "learn to code" (SPL2C) Apps

Joan Arnedo-Moreno
eLearn Center
Universitat Oberta de Catalunya, Spain
jarnedo@uoc.edu

Susanna Tesconi
eLearn Center
Universitat Oberta de Catalunya, Spain
stesconi@uoc.edu

Maria Jesús Marco
eLearn Center
Universitat Oberta de Catalunya, Spain
mmarcog@uoc.edu

David García
eLearn Center
Universitat Oberta de Catalunya, Spain
dgarciaso@uoc.edu

Marta Fondo
eLearn Center
Universitat Oberta de Catalunya, Spain
mfondo@uoc.edu

ABSTRACT

Coding skills have gained popularity beyond the technical programs of university education, leading to the emergence of many free online resources for informal learning. Among them, self-paced "learn to code" (SPL2C) apps are very popular in digital distribution stores. However, the academic literature agrees that acquiring coding skills is a demanding endeavour, with one of the highest dropout rates among all fields in Computer Science. For that reason, such apps are very sensitive to user retention, which they often address by including game-like design principles or elements that nicely mesh with their sandbox "study at your own pace" nature. The present paper studies this category of apps in order to identify the most relevant ones at this moment, while analyzing which are the most popular game elements used in them. This makes it possible to compare the gameful design approaches in this kind of apps with others in the wider educational context.

CCS Concepts

•**Human-centered computing** → **Human computer interaction (HCI);** •**Applied computing** → **Computer-assisted instruction;** •**Social and professional topics** → *Adult education;*

Author Keywords

Programming; Coding skill; Education; Mobile app; Gamification; Gamful element; Testing

In: J. Arnedo-Moreno, C.S. González, A. Mora (eds.): Proceedings of the 3rd International Symposium on Gamification and Games for Learning (GamiLearn'19), Barcelona, Spain, 22-10-2019, published at http://ceur-ws.org

1. INTRODUCTION

Programming competences are now considered very useful in the context of general learning for the near future, and for that reason, several initiatives promote their acquisition [22]. They are aimed at all levels of society, such as children at primary/secondary school level or adults who want to prepare for current or future professional requirements. In recent years, coding skills have gained special relevance beyond the areas associated with technical programs of higher education (HE). This is mainly due to the resurgence of the concept of computational thinking [6, 17], now considered the fifth "C" in the 21st century skills [21], along with critical thinking, creativity, collaboration and communication.

The increased interest in acquiring coding skills has lead to the appearance of a large number of compelling informal resources. Directed at kids, there are LOGO-like games, such as Coddy [18], among many examples, or creative development environments, the main example being Scratch [11]. For an older audience, specialized question and answer sites, such as Stack Overflow [8], and mobile SPL2C apps are especially popular. However, the crux of the matter is that learning to program is difficult, and even in formal studies, programming courses tend to have very high dropout rates. Several theories have been formulated in the educational literature on the topic to find the reasons [14, 19]. Student motivation and dedication is a key factor, as becoming an expert programmer may take up to 10 years [23].

Focusing on this last group of mobile apps, their approach to cope with users' motivation and retention problem is to offer a more flexible and engaging experience in comparison to traditional courses. Usually, as their core approach, they offer a sandbox environment, structuring knowledge into small bits that can be consumed step-by-step, with a very clear learning path that allows students to slowly advance at their own pace, without the constraints of rigorous study terms. Often, this scaffolded learning approach is complemented with the use of game-like design principles or elements (without becom-

ing full fledged games themselves) [15], such as challenges, concrete goals or extrinsic incentives [24]. As a result, users often report that they prefer these approaches when learning to code [1], thus providing a reason for its popularity.

This paper focuses on these so called SPL2C apps, not targeted to children, which can be found in numbers in mobile digital stores. The goal is to identify the most relevant apps to learn how to code that include gameful approaches to engage users, but are not games themselves. Then, to provide a thorough analysis on the most popular game elements in these apps, and to compare them to other such approaches in the general educational context. This study provides an insight on this mobile app segment and the particularities of their chosen gameful approaches in the specific context of the informal acquisition of programming skills.

The papers is structured as follows. First of all, a brief background on the study of gameful approaches in the context of learning is laid out in Section 2. Following, Section 3 presents the methods of research, how the data was compiled and the main body of the analysis of the selected apps. Section 4 provides reflection on on the results and how they relate to other works on gameful approaches and learning. Finally, Section 5 concludes this paper and outlines further work.

2. BACKGROUND AND RELATED RESEARCH

The applications of gameful approaches in the context of education at large have been extensively studied in several literature reviews or surveys, all of them published after the term "gamification" had reached the peak of Gartner's Hype Cycle for Emerging Technologies [9]. These reviews are very useful to assess which dimensions are worth to consider when analyzing this kind of proposals and, in particular, how to categorize and compare their gameful elements.

Some of such reviews focus on aggregating large amounts of demographic data, such as the ones by Caponetto [2] and Martí-Parreño [13]. The former study encompasses 119 papers from 2011 to early 2014, providing information about year of publication, country and target population. The latter considers 139 different papers on the topic, from 2010 to 2014 and the main aspects taken into consideration in the categorization process are publication journal, institution of origin and authors. There is also a brief survey on content constructs (e.g. Effectiveness, Attitudes, Engagement, Cognitive, etc.), but it is not based on an analysis of the proposals themselves or their game elements, but through word frequency analysis. In addition, a much smaller study, 26 papers from 2011 to 2013, can be found by De Sousa's [3]. This one goes a bit beyond simple data aggregation, presenting also a categorization by research objective and enumerating all the studied papers. Nevertheless, even though these surveys gather data from a large number of publications and provide a good overall idea of the context in which gamification is used in education, they do not provide insightful information about how gamification is implemented and which game elements are the most popular and/or effective.

There are some literature reviews that do categorize papers by game elements. The one by Nah [16] includes 15 papers from 2012 to 2013 and mainly focuses on analyzing publications according to game design elements (classified into Points, Levels/Stages, Badges, Prizes/Rewards, Progress bar, Storyline and Feedback), as well as expected outcomes. Dichev and Dicheva also presented a quite thorough survey in 2017 [4], the final result of successive incremental surveys also published the previous years (nevertheless, data is only up to year 2015). It includes 51 papers, categorized according to many different dimensions, such as subject of study, type of learning activity or learning and behavioral outcomes. The many game design elements are identified by the authors to categorize works, but a heavy emphasis is put on Points, Badges, Levels, Leaderboards and Progress bars. Additionally, each included paper is also individually categorized based on its Gamified elements, Subject, Sample size, Duration, Data Collection and Outcome. Both reviews reference every individual paper included in the study and how each one can be traced to each chosen category. Therefore, they provide the reader with a good starting point for further analysis of the articles included in the reviews.

The latest review found at the time of this writing is from Koivisto, published in 2018 [10]. This is the biggest study that considers a categorization by game elements (although the term "affordances" is used in this paper instead), including 128 papers up to 06/2015. Game elements are split into 32 different categories divided between Achievement/Progression, Social, Immersion, Non-digital or Miscellaneous. Psychological and behavioral outcomes are also taken into consideration. However, probably because of the high number of papers, the results are aggregated and papers are not referenced. Hence, the study presents a good general overview of the most popular game elements found in the literature but cannot be used to further explore the topic, as it does not provides references to the subjects of the study.

Focusing on gameful approaches in the specific subject of learning programming, a survey was published in 2017 [5]. However, it only includes 7 papers published from 2012 to 2014, and only 4 of them are actually about coding (the other three are about related topics such as Software engineering or databases). The range of gameful elements taken into consideration for the analysis is also very limited, as it is only centered on the well-known PBL template, based on Points, Badges and Leaderboards. As a result, the utility of this study is extremely marginal.

3. RESEARCH METHOD

The purpose of this study was to find out which kind of gameful approaches are the most popular in "learn to code" mobile apps. On that regard, it is worth noting that all of the previous literature reviews (see Section 2) studied published papers, but did not directly evaluate technological platforms *per se*.The data gathering procedure followed a systematic process devised to identify as many apps as possible that could then become candidates for a more in-depth personalized study of their characteristics.

3.1 Research Questions

The research questions for this study were:

RQ1 - How common is the use of gameful approaches among the most relevant SPL2C apps?

RQ2a - What are the most common gameful elements in SPL2C apps?

RQ2b - Are the gameful approaches in this context different from the ones found in education at large?

RQ3 - How are the gameful elements implemented and which are the idiosyncrasies of the "app" context?

3.2 Identification of candidate apps

Candidate apps were selected through a multi-step process that relied mostly on the Google Play Store. The Play Store was chosen given Android's mobile market share (about 75%), and its simplicity in the app publication process, which makes it a popular default choice for developers. Its public API also allows the automation of some tasks, opening the possibility to create an initial very large pool of apps and process their meta-data, something that would be extremely time consuming using a fully manual approach.

Step 1) A very broad search was executed on Google's Play Store [7]. An automated process was developed to perform a batch of queries according to different keywords, taking advantage of the store's API. The following Context-Free Grammar (CFG) defines the universe of queries (*QUERY*) used.

```
<A> --> learn | practice
<B> --> to code | to program
<C> --> coding | programming
<L> --> (list of programming language names)
<P> --> <L> | <C> | <B> | <L><C>
<QUERY> --> <A><P> | <L><C> | <C>
```

The list of programming language names to be used in this study was extracted from the most popular ones according to Stack Overflow's 2019 survey "Most Popular Technologies - Programming, Scripting, and Markup Languages"[1]. Stack Overflow is one of the most renowned question and answer sites for professional and enthusiast programmers on the Internet (usually ranked among the top 50 in the Global Alexa ratings[2]), and about 90.000 people participated in the survey. However, when configuring the list, it was found that the "C" language name, lacking expressivity by being a single letter, and other relatives such as C++ or C#, having special characters, caused queries to behave erratically in the Play Store. Therefore, they were not included. At the end, the list was composed by 13 of the 16 most popular programming languages, the excluded 3 being those from the C language family.

Using this approach, a total of 114 different queries were executed, obtaining a total of 2214 individual results. Once repetitions across separate queries were taken into account (612), the number of starting app candidates was set at 1602. A preliminary inspection showed that, given the broad search terms used and the inclusion of other more generic keywords, C language apps were sufficiently included nevertheless.

Step 2) The list of candidates was pruned according to the following set of strict and basic criteria which would ensure some degree of popularity and relevance of the chosen apps. It should be taken into account that the uncurated nature of the Play Store allows anybody to publish an app, regardless of its quality or following. Many apps are hardly ever known by the public, or downloaded by very few people. The criteria used to prune the list of candidates from Step 1 were:

a) Market status as "Published" (i.e. an active app).

b) Category set as "Education".

c) A rating of at least 4.0 (i.e. 4 stars out of 5).

d) Minimum number of times rated, 1.000.

e) Minimum number of downloads, 10.000.

To provide some perspective, a very popular learning app such as Duolingo, ranked 100 in the Play Store at the time of this writing, has been rated 7.327.126 times at 4,7 stars, and downloaded more that 100 million times.

Again, this was a fully automated process, from the app meta-data also obtained during Step 1. The chosen values were fine-tuned several times until an approachable number of results was obtained. Under these final criteria, 594 apps did not fulfill b), 763 did not fulfill c), 1552 did not fulfill d), and 1291 did not fulfill e), for a total of 1565 discarded apps. It must be noted that many apps did not fulfill several criteria at once, and all apps fulfilled criteria a). After the filtering process, the number of app candidates was reduced to only 37.

Step 3) From this step on, the process moved from an automated system to a manual one. The resulting apps were installed on a Pixel Android phone. In order to identify whether gameful approaches were used for user engagement, for each app, its description was read, and then executed and manually inspected. The main reasons for discarding apps during this step follows, with the total number of discarded apps between parentheses at the end:

- Directed at children or games. Not about professional programming languages (2).
- No meaningful gameful approaches included (6).
- Actually, instances of the same app, each for a different programming language (9).
- Programming tools, but not really learning apps. E.g. An integrated Development Environment (IDE) or plain API references (10).
- Not really about learning how to program at all (1).

The final number of core candidate apps was thus reduced to 9.

Step 4) As an additional step to fine tune the selection process, the list of "similar apps" provided by the Play Store for each of the 9 apps was then taken into consideration, searching for

[1] https://insights.stackoverflow.com/survey/2019

[2] https://www.alexa.com/topsites

Table 1: Final list of selected SPL2C apps

App name	Short	Stars	Ratings	Downloads
Datacamp	DaCa	4,7	3.038	100k+
Codemurai	Code	4,7	2.000	50k+
Enki	Enki	4,6	14.238	500k+
Mimo	Mimo	4,5	58.658	1M+
Grasshopper	Grass	4,7	14.628	1M+
Programming Hero	PrHe	4,7	582	10k+
Programmer Hub	PrHu	4,3	66.188	1M+
SoloLearn	SoLe	4,8	331.864	3M+
Code Academy Go	CoAc	2,8	206	50k+
Py	Py	4,7	10.856	50k+
Learn Programming[3]	LePr	4,1	12.813	1M+
Master Android	MaAn	4,7	9.966	500k+
Java Programming[4]	JaPr	4,4	9.118	1M+

additional suitable candidates, somehow missed during the automated processes. During this step, interesting apps were taken into consideration even when they did not strictly meet all criteria applied during the automated process of Step 2 (mostly, minimum number of times rated). Four additional apps were considered eligible, for a total of 13 final candidates.

3.3 Analysis

The final list of apps considered for study is shown in Table 1. A summarized name, that is used in some parts of the rest of this paper, is written in column "Short". Some general stats such as store rating (Stars), number of times rated (Ratings) and approximate minimum number of Downloads are also included, to provide some insights about their popularity. It should be noted that the Play Store does not provide an exact number of downloads, just a minimum-maximum range. In addition, a footnote with the exact package name has been included in the table for those apps with a very generic name, broadly repeated in the store, to better identify them.

The analysis was based on interaction with each application as a standard user for one week. During this time, different gameful elements were identified and classified accordingly. Some possible categorizations were presented in Section 2. However, the chosen approach was based on *"The Periodic Table of Gamification Elements"* proposed by Marczewski in his Gamification Hexad User Types framework, a widely accepted and popular approach [12]. This table encompasses 52 different elements, divided in eight categories. Of these, six are mapped to specific user preferences towards different motivations in non-leisure contexts (also know as "user types"), and the remaining two are considered generic across all kinds of users, labelled as *"General"* (GEN) and *"Reward Schedule"* (SCH). This framework was validated by Tondello et al. [20], presenting a standard scale to score users' preferences, and their findings demonstrating the usefulness of the Hexad User Types model as a measure of preferred design elements. The six user types, each providing an insight of the kind of elements included, can be briefly described as follows:

- *Socializers* (SOC): Motivated by relatedness. They want to interact with others and create social connections.
- *Free Spirits* (FRS): Motivated by autonomy and self-expression. They want to create and explore.
- *Achievers* (ACH): Motivated by mastery. They are looking to learn new things and improve themselves. They want challenges to overcome.
- *Philanthropists* (PHI): Motivated by purpose and meaning. They are altruistic, enjoying when giving to other people and enriching the lives of others in some way with no expectation of reward.
- *Players* (PLA): Motivated by rewards. They will do what is needed to collect rewards from a system.
- *Disruptors* (DIS): Motivated by change. They want to disrupt the system, either directly or through other users, to force change that suits their interests.

It must be noted that, even though the list is divided by user type, in this study, the eight categories defined by Marczewski are only used to better organize game elements. Studying the relationship between the identified elements and user type preferences was out of the scope of the study.

4. RESULTS AND DISCUSSION

The summary of the results provided by the in-depth analysis is shown in Table 3, located in Appendix I because of its size. Columns describe apps, listed using its summarized name (see Section 3.2) and rows enumerate gameful approaches, using Marczewski's classification (see Section 3.3). Even though Marczewski's list includes up to 52 gamification elements, only those that were actually identified in any app during the analysis are listed, for a total of 25. Elements included in a given app are marked as "Y", and those not included as "N" (i.e. Yes/No). The end column shows how many times a particular gameful element appeared across each of the different apps, whereas the end row shows the total sum of game elements included in each individual app.

From the information in the last row of the summary table, it is possible to provide an answer to research question RQ1, "How common is the use of gameful approaches among the most relevant "learn to code" apps?". The number of elements included range between 2 and 18 (mean=7.615, std=4.61). In 5 out of all 13 studied apps the use of gamification is very marginal. 2 elements only, in three of them: "Programming Hub", "Master Android" and "Java Programming". In the rest, the use of gameful approaches could be considered common and consistent with the app itself. A special mention is worth for "Programming Hero" and "SoloLearn", both at the top with 15 and 18 elements, respectively. Therefore, these two could be considered the most relevant ones in the context of gameful approaches in SPL2C apps.

4.1 Comparison with other gameful approaches

Regarding differences among gameful approaches in apps and other educational contexts, the last column from Table 3 allows clearly answering research question RQ2a, "What are

[3]tursky.jan.nauc.sa.html5

[4]ab.java.programming

the most common gameful elements in SPL2C apps?". The top five most common gameful elements in this kind of apps are, by order: "Progress/Feedback", "Points", "Badges", "Fixed Rewards", and "Challenges". Its is noteworthy that in those applications with a very limited number of gameful elements, Points and Badges (both among the most common) are usually the ones of choice. Also, since there are many instances of different kinds of reward elements, Marczewski defines three different subtypes (fixed, time-dependent and random), they appear in 8 out of 13 different apps.

In order to answer RQ2b, "Are the chosen approaches in this context different from the ones found in education at large?", the top five elements from each review that did perform a categorization (see Section 2), are also listed in Table 2. For a better comparison, the first column shows the results from RQ2a, ordered by rank. On that regard, it is worth noting that there is a difference between the concept of "Level" as defined by Nah, and the rest of the literature (including this paper). In the former, its meaning is equivalent to "Stage", whereas in the rest of the publications it is used to define some kind of character growth or power. Therefore, it has been considered that Nah's "Levels" are similar to Marczewski's "Challenges" or Koivisto's "Quests". Also, Marczewski considers "Progress/Feedback" as a single element, whereas other studies often differentiate between "Feedback" and "Progress" (usually just labelled as "Progress bars"). Therefore, for a fair comparison, both kind of elements are aggregated in the final results.

From this ranking, it can be seen that elements such as "Points", "Badges" And "Leaderboards" are very common and appear in almost all of them, one might say unsurprisingly. This triad of elements is very common in the literature, usually used in conjunction and labelled as PBL (referring to the initials of each element). However, it is an interesting finding that "Leaderboards" are not actually that common in SPL2C apps in particular (2 out of 13 apps use them). In addition, the top element in this kind of apps is "Progress/Feedback", which only tops the rankings, 5th position but very close to the 6th, in Koivisto's ssurvey. Therefore, two key differences have been detected between approaches in education at large and SPL2C apps.

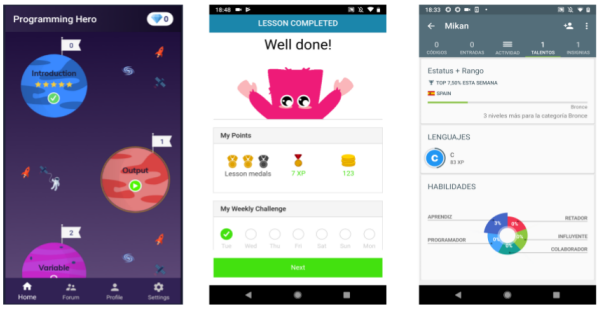
Table 2: Common gameful elements rank by study

Element	SPL2C	Nah	Dichev	Koivisto
Progress/feedback	1st	-	-	5th
Points	2nd	1st	2nd	1st
Badges	3rd	2nd	1st	3rd
Prizes/Rewards	4th	3rd	-	-
Challenges/Quests	5th	-	-	2nd
Levels/Stages	-	4th	4th	-
Leaderboards	-	5th	3rd	4th
Avatars	-	-	5th	-

4.2 Relevant properties of chosen gameful elements

Finally, the answer to research question RQ3, "How are this gameful elements implemented and which are the idiosyncrasies of the app context?", requires going beyond the summarized data, and using a more qualitative approach, extracted from the interaction with the apps as users during the analysis. Answering this provides better insights on how game elements work.

As seen in the answer to RQ2b, some very common gameful elements in education at large are also popular in SPL2C apps, such as points, badges or level ups, and their implementation is not too different. However, the top element is "Progress/Feedback", usually implemented also in a very straightforward manner with progress bars, showing how far into a lesson or course the student has progressed. However, sometimes the preferred metaphor is a semicircular gauge and a percentage. There are also some more elaborate variations, such as an adventure map that an avatar representing the student traverses (Grasshopper, Programming Hero; see Figure 1a). On regards to the "feedback" part of this element, lesson completion is often assigns stars or medals, in a similar manner to levels in some videogames, depending on exercise accuracy (Codemurai, Py, Programming Hero; ; see Figure 1a and 1b). Only SoloLearn provides some extra aggregate feedback and global student stats by given different categories, quite similar to gamification player-types: learner, programmer, challenger, influencer and collaborator (see Figure 1c).



(a) (b) (c)

Figure 1: Progress/feedback examples

The next kind of gameful element at the the top five in this kind of apps are Reward related ones, almost always implemented using a Fixed Schedule or Time Dependent strategy (using Marczewski's nomenclature). The former type is very common, always taking the form of daily or weekly "streaks". The app registers, and clearly shows, the number of consecutive days a student has logged in and advanced in a lesson. After some number of consecutive days, usually a week, the student is rewarded, mostly, with some variation of points. A few apps also use a Time Dependent strategy (DataCamp, Codemurai, Mimo, SoloLearn), on top. In addition, points are also automatically awarded every day the user logs in, or there is a daily challenge (i.e. an exercise or test) that is only available for 24 hours, mimicking the behaviour of many Free-to-Play mobile games. In one case (Codemurai), this system also incorporates a Consequence element when exercises are failed, using a videogame-like "3 lives" mechanism, that goes beyond simple scoring.

It is also worth noting that, even though several different gameful elements related to socialization (Marczewski's Philatropist

or Socializer player types) have been identified, few apps take them into consideration. They are seldom integrated in the app itself, and mostly part of the dynamics of an external discussion forum. The only exception is the Sharing Knowledge element, based on the ability to publish coding personal projects and process comments/voting mechanisms from other users. In fact, SoloLearn is the only app that makes a relevant use of this kind of elements, of special note being the ability to duel with other students in a player-vs-player competition.

On regards to the rest of gameful elements, even though the analysis shows a wide breadth of different ones, up to the 25 identified, their use is marginal (about half of them make 1-3 appearances at most). However, among them, the use of narrative or themes in two apps, an often underused element, deserve some attention. In Grasshopper, the student is guided through the lessons by this animal, as a kind of pet and companion during the coding exercises. In Programming Hero, the student becomes an astronaut and uses the metaphor of space travel through different planets (lessons) to advance through the content (see Figure 1a). Therefore, the use of a metaphor goes beyond the "Progress/Feedback" element, as previously mentioned, and permeates the whole user experience.

5. CONCLUSIONS

This paper has analyzed the most relevant SPL2C apps in order to assess their use of gameful approaches for user retention, and how they are implemented. The goal was to compare them to approaches in education at large, to try to find the particularities of the mobile app educational context, if any. Also, this study has shown that gameful approaches are also used in this kind of apps, and in some cases make an extensive use of them.

From the answers to the proposed research questions, it is shown that this kind of apps have their own idiosyncrasies. Even though some very popular gameful elements in education at large are included, such as points and badges, some others are not, as is the case of leaderboards. Furthermore, they especially emphasize elements related to progress/feedback, which play a minor role in most proposals in the literature related to gamification in education. Timed challenges are also found to be quite relevant, but not usual in education at large.

Expected future work is twofold. On one hand, a more user-centric analysis would be suitable, measuring up to which degree the chosen gameful elements are actually engaging, and taking into account player types (or any other approach to categorize the different kinds of users). On the other hand, and moving beyond the topic of gamification itself, the chosen apps will be further analyzed in order to classify their preferred educational resources and tools. This will allow to assess how common these resources are in the educational literature and their perceived usefulness in learning programming.

6. ACKNOWLEDGEMENTS

This work was partly funded by the eLearn Center (Universitat Oberta de Catalunya) through the *Xtrem Project* programme.

7. REFERENCES

1. J. Boustedt, A. Eckerdal, and R. et al McCartney. 2011. Students' Perceptions of the Differences Between Formal and Informal Learning. In *Proceedings of the Seventh International Workshop on Computing Education Research (ICER '11)*. ACM, 61–68.
2. I. Caponetto, J. Earp, and M Ott. 2014. Gamification and education: A literature review. In *In Proceedings of the European Conference on Games-based Learning*. 50–57.
3. S. de Sousa Borges, V.H.S. Durelli, H.M. Reis, and S. Isotani. 2014. A Systematic Mapping on Gamification Applied to Education. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing (SAC '14)*. ACM, 216–222.
4. C. Dichev and D. Dicheva. 2017. Gamifying education: what is known, what is believed and what remains uncertain: a critical review. *International Journal of Educational Technology in Higher Education* 14, 1 (Feb 2017), 9.
5. R. Elshiekh and L. Butgerit. 2017. Using Gamification to Teach Students Programming Concepts. *Open Access Library Journal* 4, 8 (August 2017).
6. F. J. García-Peñalvo. 2018. Editorial Computational Thinking. *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje* 13, 1 (2018), 17–19.
7. Google. 2008. *Google Play*. `https://play.google.com`. (22 October 2008).
8. Stack Exchange Inc. 2008. *Stack Overflow*. `https://stackoverflow.com`. (15 September 2008).
9. Fenn J. and LeHong H. 2013. Hype Cycle for Emerging Technologies. (2013).
10. J. Majuri, J. Koivisto, and J. Hamari. 2018. Gamification of education and learning: A review of empirical literature. In *Proceedings of the GamiFIN Conference 2018*. CEUR Workshops, 11–19.
11. J. Maloney, M. Resnick, and N. et al Rusk. 2010. The Scratch Programming Language and Environment. *ACM Transactions on Computing Education* 10, 4 (November 2010).
12. A.C. Marczewski. 2015. *Even Ninja Monkeys Like to Play: Gamification, Game Thinking and Motivational Design*. CreateSpace Independent Publishing Platform.
13. J. Martí-Parreño, E. Méndez-Ibáñez, and A. Alonso-Arroyo. 2016. The use of gamification in education: a bibliometric and text mining analysis. *Journal of Computer Assisted Learning* 32, 6 (2016), 663–676.
14. I. Milne and G. Rowe. 2002. Difficulties in Learning and Teaching Programming—Views of Students and Tutors. *Education and Information Technologies* 7, 1 (2002), 55–66.
15. A. Mora, D. Riera, C. González, and J. Arnedo-Moreno. 2017. Gamification: a systematic review of design frameworks. *Journal of Computing in Higher Education* 29, 3 (01 Dec 2017), 516–548.

16. F.F Nah, Q. Zeng, and V.R. et al Telaprolu. 2014. Gamification of Education: A Review of Literature. In *HCI in Business*. Springer International Publishing, 401–409.

17. S. Papert. 1996. An exploration in the space of mathematics educations. *International Journal of Computers for Mathematical Learning* 1, 1 (1996), 95–123.

18. Simply Projects. 2017. *Coddy: World on Algorithm*. Google Play. (14 February 2017).

19. A.V. Robins, J. Rountree, and N. Rountree. 2003. Learning and Teaching Programming: A Review and Discussion. *Computer Science Education* 13 (2003), 137–172.

20. Gustavo F. Tondello, Rina R. Wehbe, Lisa Diamond, Marc Busch, Andrzej Marczewski, and Lennart E. Nacke. 2016. The Gamification User Types Hexad Scale. In *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play (CHI PLAY '16)*. ACM, New York, NY, USA, 229–243.

21. B. Trilling and C. Fadel. 2012. *21st century skills: Learning for life in our times*. John Wiley Sons.

22. P. Tuomi, J. Multisilta, P. Saarikoski, and J. Suominen. 2018. Coding skills as a success factor for a society. *Education and Information Technologies* 23, 1 (2018), 419–434.

23. L.E. Winslow. 1996. Programming Pedagogy&Mdash;a Psychological Overview. *SIGCSE Bull.* 28, 3 (1996), 17–22.

24. J.R. Young. 2012. 'Badges' Earned Online Pose Challenge to Traditional College Diplomas. *The Education Digest* 78, 2 (2012), 48–52.

APPENDIX I

Table 3: SPL2C apps: Gameful elements summary

APPS / FEATURES	DaCa	Code	Enki	Mimo	Gras	PrHe	PrHu	SoLe	CoAc	Py	LePr	MaAn	JaPr	**TOTAL**
GEN - On-boarding/Tutorials	N	N	N	N	Y	N	N	N	Y	Y	N	N	N	3
GEN - Signposting	N	N	Y	N	N	N	N	Y	Y	N	N	N	N	3
GEN - Flow	N	N	Y	Y	Y	N	N	Y	N	Y	N	N	N	5
GEN - Theme	N	N	N	N	Y	Y	N	N	N	N	N	N	N	2
GEN - Progress/Feedback	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	11
GEN - Consequences	N	Y	N	N	N	Y	N	Y	N	N	N	N	N	3
GEN - Strategy	N	N	N	N	N	N	N	N	N	N	Y	N	N	1
SCH - Fixed Reward Schedule	Y	Y	Y	Y	Y	Y	N	Y	Y	N	N	N	N	8
SCH -Time Dependent Rewards	Y	Y	N	Y	N	N	N	Y	N	N	N	N	N	4
SCH -Random Rewards	N	N	N	N	N	Y	N	Y	N	N	N	N	N	2
PLA - Badges/Achievements	N	Y	N	Y	Y	Y	N	Y	Y	N	Y	Y	Y	9
PLA - Leaderboards/Ladders	N	N	N	N	N	Y	N	Y	N	N	N	N	N	2
PLA - Virtual Economy	N	Y	N	N	N	N	N	N	N	N	N	N	N	1
PLA - Points/Experience (XP)	Y	Y	N	N	Y	Y	N	Y	Y	Y	Y	Y	Y	10
ACH - Levels/Progression	Y	Y	N	N	N	Y	N	Y	N	Y	Y	N	N	6
ACH - Certificates	N	N	N	Y	N	Y	Y	Y	Y	N	N	N	N	5
ACH - Challenges	Y	N	N	Y	Y	Y	N	Y	Y	Y	N	N	N	7
SOC - Guilds / Teams	N	N	Y	N	N	Y	N	N	N	N	N	N	N	2
SOC - Competition (PvP)	N	N	N	N	N	N	N	Y	N	N	N	N	N	1
SOC - Social Status	N	N	N	N	N	Y	N	Y	Y	N	Y	N	N	4
SOC - Social Network	N	N	N	N	N	N	N	Y	N	N	N	N	N	1
FRS - Creativity Tools	N	N	N	N	N	N	N	Y	N	N	N	N	N	1
FRS - Customization	N	N	N	N	N	N	N	N	N	N	Y	N	N	1
PHI - Sharing Knowledge	Y	Y	N	N	Y	Y	N	Y	Y	N	N	N	N	6
DIS - Voting/Voice	N	N	N	N	N	Y	N	N	N	N	N	N	N	1
Total elements included	7	9	5	7	9	15	2	18	10	6	7	2	2	