Coding is Fun: Engaging Adult On-Line Learners Using Programming Games

Joan Arnedo-Moreno(✉) and David García-Solórzano

Universitat Oberta de Catalunya, Barcelona, Spain
{jarnedo,dgarciaso}@uoc.edu

Abstract. Programming and computational thinking have become very important in the context of general learning for the near future, and for that reason, a wide breadth of initiatives to promote their acquisition exist. However, two of the main entry barriers are student engagement, and the fact that solving coding problems requires a more abstract approach, instead of a purely technical one, relying on problem-solving skills. To address these two issues, this paper presents a study on the use of two very popular and highly rated computer games of the "programming" genre (named "Human Resource Machine" and "7 Billion Humans", applied in different distance learning courses aimed at adult students (median age, 31). The results show that both were perceived as very useful during the initial weeks of the semester to complement their knowledge acquisition through practice, as well as having a positive impact in their sense of self-efficacy. Mostly, thanks to their engaging interface, immediate feedback and presenting failure as just a challenge to the player to try again and do better.

Keywords: e-learning · video games · coding · programming game · computational thinking · Tomorrow Corporation · edutainment

1 Introduction

Coding skills have gained special relevance even beyond the areas associated with technical programs of higher education (HE). This is mainly due to the resurgence of the concept of computational thinking [1, 2], now considered the fifth "C" in the 21st century skills [3], along with critical thinking, creativity, collaboration, and communication. However, computer programming students seem to suffer from a wide range of difficulties when facing this subject [4, 5]. Students tend to approach computer programming from a low-level technical standpoint, instead of a more abstract one, relying on their problem-solving skills [6]. This has led to a longstanding exchange of ideas in academia on what challenges novice students encounter and how to best introduce this topic to beginners [7].

In recent years there is a resurgence of what are colloquially known as "programming games", although this genre has been formally identified since 2001, at least [8]. In such games, challenges are purposely designed and presented to the player in such a manner that computational thinking and abstract problem-solving have a fundamental role in

F. J. García-Peñalvo and A. García-Holgado (Eds.): TEEM 2022, LNET, pp. 606–614, 2023.
https://doi.org/10.1007/978-981-99-0942-1_63

successfully overcoming them. Even though many of them could be considered close to serious games (SG) [9], the most successful and well-known ones are designed as pure entertainment products, proliferating in digital distribution platforms such as Steam[1] or GOG[2], as commercial off-the-shelf (COTS) games sold next to the latest installments of other famous AAA video game franchises.

It may seem counter-intuitive to learn from playing digital games that were designed purely just for fun, and not education. However, it could be hypothesized that at least some of them could be useful to introduce basic computational thinking or coding skills, despite their entertainment first approach. This would allow educators to capitalize on their game design principles, based on a proven engagement track record, and their ability to reach a mainstream population, being COTS products. But first, it would be necessary to subject them to a degree of scrutiny as any other educational tool [10].

This paper presents a study on the use of two very popular and highly lauded computer games of the "programming" genre (named "Human Resource Machine" and "7 Billion Humans" [11, 12]), as the means to instruct and engage adult learners across different technical bachelor degrees in their first semester learning computer programming. The main contribution of this study is to provide educators interested in "edutainment" [13] approaches with a better understanding of the impact of such games on an older audience, in a distance learning context. This is of special significance, since studies on programming games usually focus on younger audiences, from primary school to the first steps of higher education.

This paper is structured as follows. Section 2 provides the theoretical background for this work, presenting the main characteristics of the programming genre of video games, and how it can be reconciled with the Computer Science (CS) programming curriculum in higher education. Next, Sect. 3 formalizes the research questions and describes the methodological design of the study. The results and their interpretation are presented in Sect. 4. Finally, Sect. 5 concludes this paper and provides some insights on present and future work.

2 Background

Programming games are described by Wolf [8] as ones "*in which the player writes short programs that control agents within a game. These agents then compete and react to situations based on the player's programming*". Nevertheless, depending on what are the goals of the game and the possible actions of the agents, a programming game may also share other genres (e.g. adventure, platform, etc.). Even though still a niche genre, a brief analysis of Steam, the most popular video game digital distribution platform among PC, can provide a general idea about the growing interest in this kind of games in recent years.

From a purely academic standpoint, the study and development of programming games to promote the acquisition of algorithmic and coding skills is certainly not new. There is a vast literature on the topic, summarized in successive reviews, that also provide

[1] https://store.steampowered.com.

[2] https://www.gog.com.

an estimation of the number of existing games in this genre that have been considered for use at different stages of education [9, 14, 15]. Each of these studies includes 40, 49, and 70 games, respectively, and even though there is some overlap (13 repetitions out of 159 games), there are many games that only appear in one of them. The first two studies mostly consider games that were created with a clear educational context in mind, and in fact, they simply refer to them as "serious games for learning programming". However, many of the games included have a distribution model very limited to academic contexts or were made just as experimental projects (some have been abandoned, now). In contrast, the latest study focuses on widely available, and some of them very popular, COTS games.

The most important aspect of these studies is their effort to discern which are the computing skills and learning outcomes covered by each game, to guide educators in their application in class. The preferred method is to evaluate each game according to a list of topics extracted from the CS2013's SDF knowledge area (assessing 10 and 18 programming topics for the first two papers, respectively), although the latter survey prefers to use Becker and Fitzpatrick's list of 15 items. Nevertheless, the final categories are similar in all cases, encompassing most of the topics to be expected in introductory programming, with just minor differences in the nomenclature.

Even though, under further inspection, not all games labelled as "programming" in digital distribution platforms are actually about programming at all, there is ample evidence in the scientific literature that some of them are able to fulfill an educational purpose, confirming Aldrich's proposition that well-designed video games can also contain sound learning principles [16]. In fact, many are aligned with some highly regarded academic proposals for a CS curriculum in introductory programming, such as the CS2013, thus carrying out the basic tenets of the educational process [17, 18].

3 Methodology

The main purpose of this study was to test whether the appropriate COTS programming games could actually be used as fruitful educational tools for adult learners, despite their inherent nature (i.e., designed for entertainment, first and foremost). Thus, the following research questions are proposed:

RQ1: What is the adult students' perception of them as educational resources?

RQ2: Did it have any impact in the student's sense of self-efficacy?

3.1 Context of the Study

The study was conducted at a fully online university offering graduate and postgraduate programs in different languages, currently amounting to about 75.000 students. The typical students are fully autonomous adult learners, 64% of them are more than 30 years old, and 27% for the 40+ age bracket. Most work full-time and are financially independent (95%), married (73%), or have children (58%), according to internal polls performed during the enrollment process. In general, they chose to study online because they cannot attend classes and need to be fully autonomous in their study time.

Specifically, the different offerings of the Programming Fundamentals (ProgFun) course were chosen. This course is held during the first year of the Computer Science, Telecommunication Engineering, and Data Science bachelor's degrees, as well as being mandatory prior to accessing any technical M.Sc. Degree program for students who lack programming skills because of their academic background (e.g. artists who want to start the M.Sc. in Video game Development). It is assumed that students enrolled in the ProgFun course have absolutely no previous knowledge in computer coding.

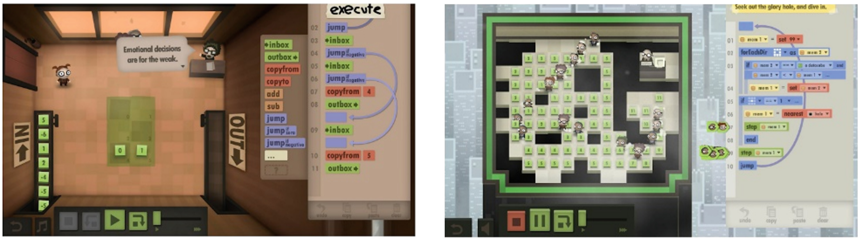
The course can be divided in three main thematic blocks. The first block encompasses the first 6 weeks and presents the basic concepts of programming: stages of the development process, problem-solving strategies, variables and basic data types, functions, expressions, and flow control (conditional and iterative). During this block, pseudocode is used at the beginning, where the approach is more conceptual, and later moving on to the C programming language towards the end. The second block is 3 weeks long and presents the concepts of modularity, structs, pointers and I/O parameters. Finally, the third block is 6 weeks long and is concerned with all the different Abstract Data Types (ADT), their implementation, and route and search strategies. Assessment relies on small weekly assignments, two projects, and a final exam (100% virtual).

3.2 Game Selection Process

Since this study was meant to focus on the properties of widely available COTS games, the initial list of basic desirable properties was defined as follows: a) The chosen game should cover part of the learning outcomes in the ProgFun course, especially the first block; b) It should focus on the more abstract parts of the learning process; c) A COTS finished game with a simple installation process and no extra requirements (i.e. no mandatory online platform or services, or DRM); d) Easily available for different platforms, that the user can choose from, e) Good production values (graphics, interface, etc.) but low hardware requirements (e.g. usable in non-gaming laptops) and high user review ratings.

After a thorough review of all available programming games, following the conclusions on learning outcomes from the literature described in Sect. 2, and checking their alignment with the introductory programming curriculum, two games were chosen: "Human Resource Machine" and "7 Billion Humans" (or HRM and 7BH) [11, 12]. Both games are highly regarded in the programming game community. They respectively have a 90.08% and 88.13% rating in Steam, or a 4.4 rating in the Android Play Store. To provide a better idea of these games, Fig. 1 shows a sample screenshot from each one.

Both games use a "Code block" interface style, using highly visual color blocks based on a drag & drop interface to create the programs. Some block types may also accept some parameters in text form (e.g., expressions). This approach was considered desirable above a game with a direct coding text interface, so the experiment would be more language agnostic.



```
jump
inbox
jump if negative
inbox
jump if negative
copyfrom 4
outbox
inbox
copyfrom 5
outbox
```

(a) Human Resource Machine

```
mem1 = set 99
forEachDir as mem2
    if mem2 == a datacube and
       mem2 < mem1 ...
        mem1 = set mem2
if == 1 ...
    mem1 = nearest hole
    step mem1
end
step mem1
jump
```

(b) 7 Billion Humans

Fig. 1. Sample screenshots from "Human Resource Machine" and "7 Billion Humans".

3.3 Research Method

Participation in the study was voluntary, and the enrollment started one week before the start of the semester through an open call to all students who had enrolled in ProgFun for any academic program in the university where the course was offered. The call was disseminated via personal tutors, and the only specific requirement to be met by any student who wanted to volunteer was that it should be their first time at the university coursing this subject.

The study was based on a single group research design, using a qualitative approach based on inductive coding through the use of reflective journals to analyze the students' experiences. Journals are an instrument considered helpful to glean useful information on a person's behavior, perceptions and feeling during an educational process [19–21], allowing the capture of the students' change in perspectives as they progress in their learning experiences with a degree of detail and personalization that can be difficult to measure with other methods [22]. Furthermore, journals are especially effective at creating and reinforcing a bond between the teacher and the student, providing an additional layer of engagement, and enhancing the learning process itself by fostering reflection and critical thinking [23, 24].

The journaling process was guided though the definition of a set of questions for each expected entry, that would help students organize their thoughts and write down their experiences in a cohesive manner. The design process of these questions followed the general recommendations by Dunlap [25], as well as some parts of the literature on self-efficacy, such as Schunk's test [26], which added a small measure of quantitative data. Beyond these guiding questions to spur their reflection process, students were absolutely free in how to approach their writings. Nevertheless, the first entry was a practice one where they presented themselves and expressed their expectations about the course. Apart from providing some useful information, feedback could be provided in a timely manner on their journaling style, to guarantee a quality baseline in further entries.

The experiment lasted for six weeks, which encompassed the first block of the ProgFun course, regardless of the academic program of origin. Participants had to write a journal entry on a weekly basis. At the start of each week, researchers would send a document introducing the broad guidelines for that week's journal entry, which students had to deliver by the end of the same week. No specific assignments were given for the week, and the experiment never changed the usual tasks expected from a student in the ProgFun course. To truly test the games' effectiveness in engaging the students, participants could interact with the games on their own volition.

4 Findings from the Journaling Process

When the course started, 51 students had committed to participate, coming from different five academic programs: B.Sc. in Computer Engineering (CE), B.Sc. in Telecommunications Engineering (TE), B.Sc. in Data Science (DS), M.Sc. in Video game Development (VD), and M.Sc. in Data Science. Figure 2 summarizes the main demographic variables of the participants in the study. Their ages were comprised between 18 and 63 years old (mean: 33.352; std: 9.688; median: 31), just a bit younger than the above-noted typical student, and with a gender distribution of 25.5% female and 74.5% male (0% others).

Journal entries were examined to determine changes in the students' perception in regard to gameplay during the experiment. Each respective RQ established the main coding scheme to analyze the qualitative data, looking for keywords or expressions related to them, until saturation was reached. Four themes were considered as the most prominent.

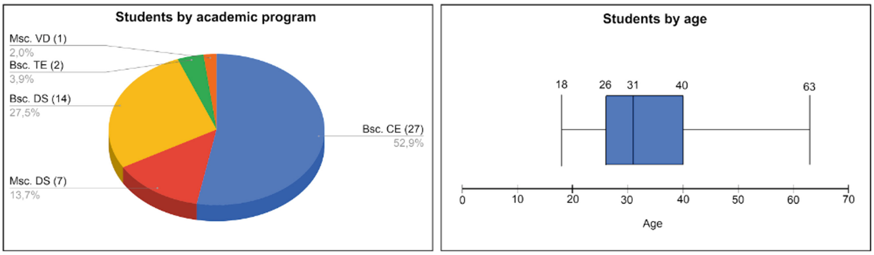


Fig. 2. Demographics of the subjects who agreed to participate in the study.

Foremost, a high degree of *stealth learning* was found. In the initial weeks, students found it very hard to correlate the games with the rest of the course contents. At the same time, they would often report that they had not learned anything specially outstanding during the course that week. Then, in later journal entries, they would realize that this perception was actually caused because they had already interiorized some concepts just from playing the games beforehand.

"This week also makes me feel like I learned very little in the subject. But unlike the previous week, this time I see clearly that it's because I had already learned it through games."

"Having to learn directly through the challenges of each level and the optimizations have made the learning process move forward on its own, and although it does not always have an obvious impact"

"The truth is that I have been "hooked" and that little by little I have discovered the relationship with the subject. In any case, I guess it should be like a voluntary complementary activity."

Second, another very important recurring topic found in the journals was the reported degree of engagement, which had mixed results. Some students were highly engaged, whereas a few hardly interacted at all, unless explicitly instructed to do so. Because of the adult learner characteristics, free time was a very relevant factor. However, the most important aspect was frustration management when they got stuck in a level. Some students considered this situation an opportunity to improve, whereas others just gave up. This frustration had a negative impact in the learning process.

Surprisingly, in many instances, the maximum levels of reported frustration were not caused by the inability to get to the next level, but when facing optimization challenges (i.e. looking for perfect scores in levels, minimizing the number of instructions in memory, or steps). The inclusion of optimization challenges was one of the most commented aspects about how students interacted with the games.

"Some levels in both cases have been frustrating to me since I have not been able to overcome them with the minimum of possible steps. In spite of trying to improve them, I have not seen the possibility."

"The game has not helped me, rather it has generated stress and slowed me down, and I have not seen that it helped me to go faster in the tasks of the subjects."

"More than understanding the subject, it helped me (and a lot) to manage my frustration when i was stuck at some level."

"I've been stuck for weeks, and I don't have the time to dedicate to the game to be able to overcome the challenges it poses."

Somehow related to the previous one, the third theme was the engagement mechanisms provided by the games. Students fancied the feedback mechanisms, that would allow them to slowly advance towards the solution (or an optimized score), and interact with the system in a very intuitive manner. Nevertheless, some students considered the games' look&feel too child-like, or even the art direction a bit disgusting. But, at the same time, sometimes this was considered an advantage, making them easy to pick up and understand.

"When you finish a level, it tells you the minimum commands and the steps you used and if you can still reduce the number of orders or time, this makes me want to improve my results and stay at the same level until I get the two green lights."

"I have shown it to my wife, who has never worked with computers, and she has reached level 3 and enjoyed it."

Finally, positive impacts on the student's belief in his or her capacity to execute behaviors necessary to produce the required performance attainments were also found. Again, the optimization challenges were key, even when taking the possibility of frustration into account. Not just passing a level, but getting a good score, made students confident in their abilities to pass the course.

"I was optimizing levels from the first week. I think it's the part I like to do the most and where I often learn the most, because it's no longer just about knowing how to solve it, it's about knowing that it's a solution that is considered optimized and therefore makes me feel more confident."

"I am very happy because I have managed to improve and optimize them."

Indeed, in addition to the qualitative analysis of journal entries, a simple single group pre-post test approach using Schunk's self-efficacy test [26] was performed. The test was presented during weeks two (Mean = 4.02083; SD = 0.62348; SEM = 0.17998) and six (Mean = 4.28125; SD = 0.56940; SEM = 0.16437), at the very end of the journaling process. The two-tailed value o P was 0.0626, which could be considered statistically significant. These results further back some of the previous topics and key statements detected in the journals.

5 Conclusions and Future Work

Given the findings from the analysis of the student journals, it is possible to provide some answers to the proposed research questions. Nevertheless. The scope of the study is limited to its application in a 100% on-line institution with mostly adult learners (30 years or older), who have their own expectations, idiosyncrasies, and learning styles.

Regarding RQ1, it can be concluded that the studied programming games were considered somehow useful during the learning process, even though never to be used as standalone resources or as an alternative to more focused programming exercises. Despite the different degrees of engagement detected during the study, especially regarding frustration levels, most students eagerly recommended them as an educational resource and enjoyed the optimization challenges on top.

As far as RQ2 is concerned, some evidence was found that the use of this kind of educational resources had a positive impact on self-efficacy, from a qualitative and quantitative standpoint. Even when the taking into account that the starting values were quite high already (Mean = 4.02083, in 1–5 a range), including people who did not know how to program at all when the courses began.

Further research includes using the student feedback to improve on the methods to integrate the use of these games as useful educational resources, as well as testing the effectiveness of other, more complex, programming games.

References

1. García-Peñalvo, F.J.: Editorial computational thinking. IEEE Revista Iberoamericana de Tecnologias del Aprendizaje **13**, 17–19 (2018)
2. Papert, S.: An exploration in the space of mathematics educations. IJ Comput. Math. Learn. **1**, 95–123 (1996)
3. Trilling, B., Fadel, C.: 21st Century Skills: Learning for Life in Our Times. Wiley (2009)
4. Lahtinen, E., Ala-Mutka, K., Järvinen, H.M.: A study of the difficulties of novice programmers. ACM SIGCSE Bull. **37**, 14–18 (2005)
5. Coull, N.J., Duncan, I.M.: Emergent requirements for supporting introductory programming. Innov. Teach. Learn. Inf. Comput. Sci. **10**, 78–85 (2011)
6. Bennedsen, J., Caspersen, M.E., Kölling, M. (eds.): LNCS, vol. 4821. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-77934-6
7. Medeiros, R.P., Ramalho, G.L., Falcão, T.P.: A systematic literature review on teaching and learning introductory programming in higher education. IEEE Trans. Educ. **62**, 77–90 (2018)
8. Wolf, M.J.: The Medium of the Video Game. University of Texas Press (2001)
9. Miljanovic, M. A., Bradbury, J. S.: A review of serious games for programming. In: Göbel, S., et al. (eds.) JCSG 2018. LNCS, vol. 11243, pp. 204–216. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-02762-9_21

10. Tomorrow Corporation: Human Resource Machine. Game [Steam] (2015)
11. Tomorrow Corporation: 7 Billion Humans. Game [Steam], 2018 (2022)
12. Anikina, O.V., Yakimenko, E.V.: Edutainment as a modern technology of education. Procedia Soc. Behav. Sci. **166**, 475–479 (2015)
13. Vahldick, A., Mendes, A.J., Marcelino, M.J.: A review of games designed to improve introductory computer programming competencies. In: 2014 IEEE Frontiers in Education Conference (FIE) Proceedings, pp. 1–7. IEEE (2014)
14. Arnedo-Moreno, J., García-Solórzano, D.: Programming is fun! A survey of the Steam digital distribution platform. In: 2020 IEEE 32nd Conference on Software Engineering Education and Training (CSEE&T), pp. 1–4. IEEE (2020)
15. Aldrich, C.: Learning by Doing: A Comprehensive Guide to Simulations, Computer Games, and Pedagogy in e-learning and Other Educational Experiences. Wiley (2005)
16. Malone, T.W.: What makes things fun to learn? A study of intrinsically motivating computer games. PhD thesis, Stanford University (1980)
17. Prensky, M.: Digital game-based learning. Computers in Entertainment (CIE), vol. 1, pp. 21–21 (2003)
18. Janesick, V.J.: A journal about journal writing as a qualitative research technique: history, issues, and reflections. Qual. Inq. **5**, 505–524 (1999)
19. Morrison, K., Manion, L., Cohen, L.: Research Methods in Education, vol. 5 (2007)
20. Goetz, J., LeCompte, M.: Ethnological and qualitative design in educational research (1984)
21. Boud, D.: Using journal writing to enhance reflective practice. In: New Directions for Adult and Continuing Education 2001, pp. 9–18 (2001)
22. Connor-Greene, P.A.: Making connections: evaluating the effectiveness of journal writing in enhancing student learning. Teach. Psychol. **27**, 44–46 (2000)
23. Hettich, P.: Journal writing: old fare or nouvelle cuisine? Teach. Psychol. **17**, 36–39 (1990)
24. Dunlap, J.C.: Using guided reflective journaling activities to capture students' changing perceptions. TechTrends **50**, 20–26 (2006)
25. Bandura, A., Freeman, W.H., Lightsey, R.: Self-efficacy: the exercise of control (1999)
26. Schunk, D.H.: Self-efficacy and cognitive skill learning. In: Research on Motivation in Education, Goals and Cognitions, 3 (1989)