Fostering Engagement and Creativity through Programming: The Beauty and Joy of Computing in a First-year Engineering class

Federico Meza*, Nicolás Torres†, Andrea Vásquez‡
Departamento de Informática, Universidad Técnica Federico Santa María
Valparaíso, Chile
*Email: *fmeza@inf.utfsm.cl, †nicolas.torresr@usm.cl, ‡andrea.vasquez@inf.utfsm.cl*

***Abstract*—There is a growing interest in learning computer programming even among students from majors other than computer science (CS). Many universities offer a common Introduction to Programming (CS1) course, but this approach is usually detrimental to those who are not pursuing a CS major. This article is an experience report that summarizes the results of adapting and implementing the course "The Beauty and Joy of Computing" (BJC) in an Engineering in Product Design first-year class, a group that showed below-average performance in the past. BJC is a course for undergraduate non-CS majors at UC Berkeley, designed to broaden student participation in Computer Science. After a fully online semester of BJC at UTFSM, students showed greater responsibility and commitment when compared to previous cohorts. They also reported greater satisfaction with programming while having a lower attrition rate and comparable final grades to students in the regular CS1 class.**

***Keywords*—Computer science education, CS0, introduction to programming, block-based programming, BJC, Snap!.**

1. Introduction

The use of computational methods and tools is transforming every discipline; it is thought that by the middle of this century these set of skills will be deemed as fundamental as reading and writing [1]. In consequence, there is a growing interest in learning to program, even among students who do not plan to (or did not) pursue a career in computer science [2], [3]. This interest is in turn driving a sustained increase in enrollment in higher education programming courses [4], [5].

Many universities offer a single introduction to programming (CS1) course, using the same syllabus and methodology for both CS and non-CS majors. For undergraduate students, this is usually their first experience in the discipline, as opposed to school level mathematics or physics [6], [7]. As such, it can have a profound impact on their intention to pursue a STEM career [8]. But one size does not fit all, and offering the same programming class for majors and non-majors can have a detrimental effect on attrition [9]. Furthermore, even though there has been an improvement on passing rates, we are still short of strategies to respond to the current increase in enrollment while being inclusive in programming classes [10].

The Beauty and Joy of Computing (BJC) is a course at UC Berkeley for undergraduate non-CS majors, designed to broaden participation in computer science of underrepresented groups [11]. This is achieved by emphasizing programming and presenting "big ideas" that are usually avoided in introductory classes, such as recursion and higher order functions, in a way that challenges and grips the interest of students. BJC puts "experience before formality", and challenges students to own and enjoy their logic and creativity to make things, promoting a sense of programming agency in an audience that is usually excluded [12].

Universidad Técnica Federico Santa María, a polytechnical university in Chile, offers a wide array of majors in engineering (civil, electrical, electronics, metallurgical, mechanical, chemical and industrial), science (physics, astronomy, maths, chemistry), informatics, architecture and business administration. Most of our first-year students take the same mandatory Introduction to Programming course (IWI-131, equivalent to CS1), with an enrollment of 2400+ students every year. This is also the first course for Informatics majors.

In the past, we have observed that the major that a student chooses has an impact on performance and motivation levels when learning programming. We believe that this is due to misconceptions regarding the importance of the course in their professional development. One case in particular came to our attention: grade reports from the last two years showed that students from the Engineering in Product Design (EPD) major often exhibit a below-average performance in IWI-131. They also reported feeling less satisfied with their experience of the course overall.

Fig. 1 shows the distribution of final grades in the regular IWI-131 course for the years 2018 and 2019, comparing EPD students and students pursuing other majors. EPD students exhibit lower performance overall with a median of 55.5/100 points for their grades, while other majors grades have a median of 68.0/100 points. Maximum and minimum grades are also significantly lower for EPD students.

Authors appear in alphabetical order.

978-1-7281-8328-2/20/$31.00 ©2020 IEEE

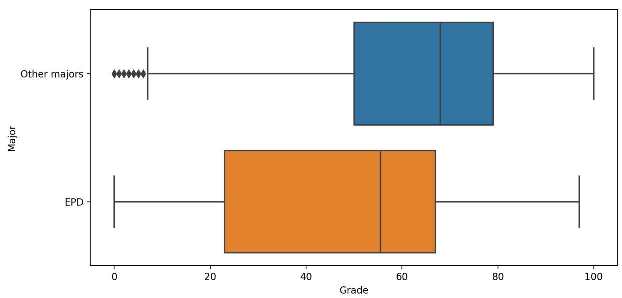

Figure 1: Final-Grade Comparison for years 2018-2019

The EPD major focuses on the design, planning and management of projects with the aim of developing products and services. Thus, EPD students are more likely to enjoy a visual and project-oriented course.

For these reasons –a below-average performance, poor satisfaction levels, and the affinity of the students for visual content– we decided to adapt and implement BJC for first-year students of EPD.

2. BJC Course Adaptation

In order to bridge the achievement gap for EPD students, we decided to adapt BJC by incorporating their realities and concerns, as well as the expected outcomes of the original programme. It is important to note that the changes were made amid the COVID-19 pandemic, and the course was taught fully online.

The course was divided into five units, each one based on a specific learning objective from the original IWI-131 course:

1) Introduction, where students describe the solution process for a problematic situation using algorithms, as a transformation from an initial state to a final state.
2) Serial algorithms, conditionals (`if`) and loops (`while`) for solving engineering problems.
3) Pattern-finding in shapes, for drawing figures. This learning outcome is not part of the original course, but an addition from the BJC original syllabus
4) Functions to solve particular subproblems within a broader problem, using a well-defined interface (parameters and return value).
5) Solving problems by using collections of data such as strings and lists. Also, because of the BJC syllabus, we included a section where students design interfaces that allow users to interact through graphical icons.

Core to the BJC curriculum is using a block-based language for instruction, called Snap!. Snap! is a modified version of Scratch, with advanced capabilities (such as recursion, higher order functions and object-oriented programming) and visual metaphors to aid students to focus on logic and coding instead of syntactic errors [12].

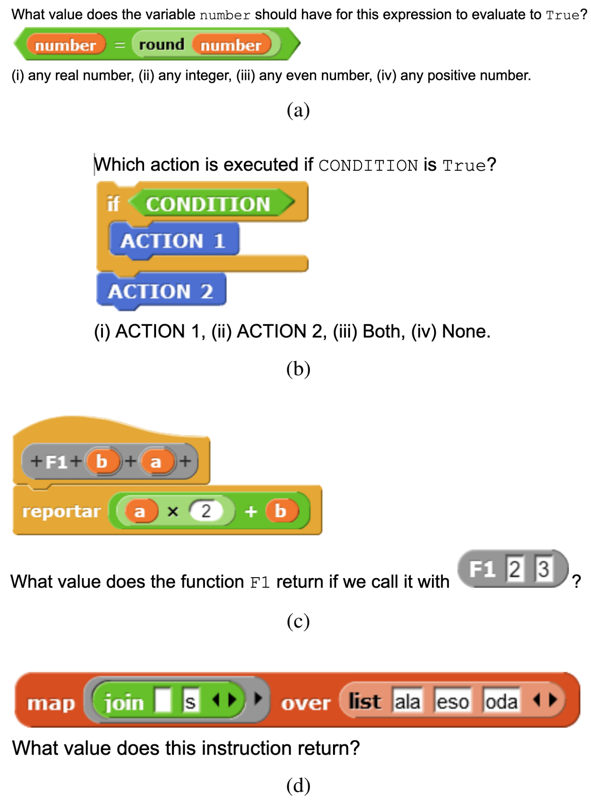

Figure 2: Examples of activities.

We delivered each unit considering practical, experiential activities to stimulate students' creativity in problem-solving and to increase their abilities in computing programming, while encouraging them to express their feelings. Following BJC core principles, evaluations were meant to be fun, and thus activities included design of user interfaces. Directions for each activity were carefully crafted to make them clear and intuitive, and had graphic hints and test cases, as shown in Fig. 2.

This careful design of the evaluations sought to avoid unnecessary cognitive load and confusion, encouraging students to persist and thus lowering attrition [13]. This is especially important in the case of female students, as they usually express higher levels of anxiety when facing tests than men [14].

As the closing activity of the course, students presented their final projects during a "demo day". This semester, many of these projects were games (arcade, car racing, quizzes), where all the sprites and stages were designed by the students (Fig. 3a, 3b, 3c). Other projects were related to music and art, for instance, composing music using a computer keyboard (Fig. 3d) or a simulated scenario for learning guitar chords (Fig. 3e). Also, some students decided to create their own world by telling a story (Fig. 3f, 3g).

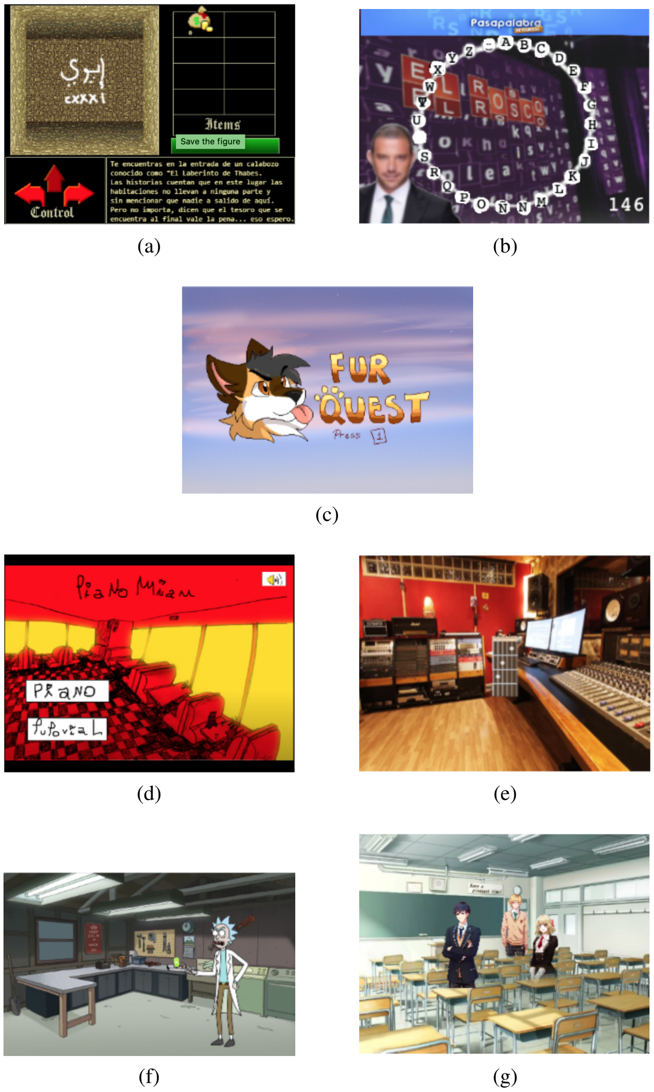

Figure 3: Screenshots of students projects: Games in (a), (b), (c); Music and art in (d), (e); Storytelling in (f), (g)

3. Results

We had 38 students enrolled in the course (19 women and 19 men), of which 36 students passed and 2 failed. On the other hand, 388 students took the regular CS1 course. Table 1 shows descriptive statistics for the final grade of the BJC class as well as attrition and pass rates, compared to other majors. Fig. 4 shows a graphical comparison of these groups during the 2020-1 semester.

The significant differences observed in the past years have been reduced, both in the median and in max/min grades. In terms of the median (see coloured boxes in the boxplot, representing 50% of the students of each group), there is a slight difference (5/100 points) in favour of the students from other majors. The difference in the median was 13 points in previous years.

TABLE 1: Term 2020-1 Descriptive statistics

	EPD	***Other majors***
No. of students	39	388
Mean	76.5	76.44
Median	81.5	86.0
STDev	21.64	25.89
Attrition	2.63%	7.99%
Pass rate	95%	85%

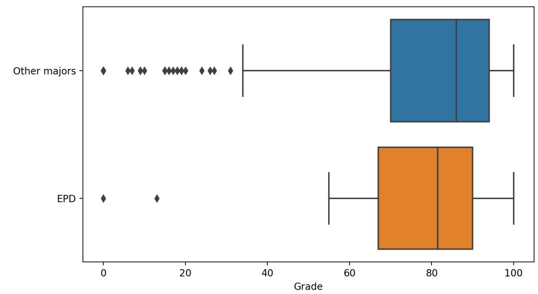

Figure 4: Final-Grade Comparison for term 2020-1

On the other hand, the mean was higher for EPD students, with a lower standard deviation. Also, the attrition rate was significantly lower for them.

Nine students shared their thoughts in an end-of-term survey. They were satisfied with the experience and enjoyed the class and problem-solving process, as shown in Fig. 5. Words like *amusing*, *learning*, *solving*, *problems*, *interesting*, and *think*, spontaneously appeared. Some of the students declared that the block-based language was useful at the beginning, but suggested that it should be replaced by Python later in the course. Finally, most students reported that having their programming course fully online was not detrimental to their learning.

Figure 5: Tag cloud from end-of-term survey responses

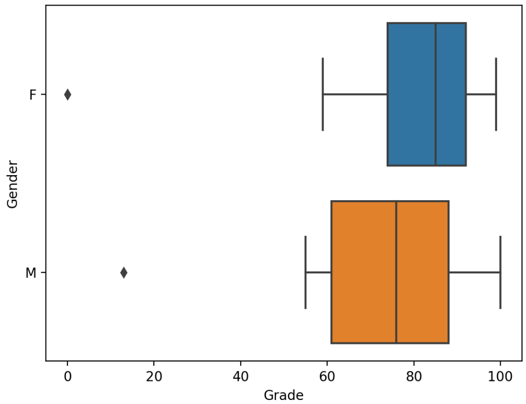

Figure 6: EDP Grades by Gender, Term 2020-1

4. Discussion

In the previous section we presented descriptive statistics related to student performance. Compared to other majors, EPD students following BJC exhibited a higher mean in the final grade and improved significantly in several measures, such as the median, minimum and maximum grades. Also, attrition rate was lower and pass rate was higher for BJC students. It is important to note that, when compared to 2018 and 2019, the grades of EPD students exhibited lower variability and significantly improved.

Regarding gender, female students in BJC obtained higher grades than male students, as we can see in Fig. 6.

From a qualitative perspective, students at first were reluctant to participate, but later there was a change of attitude and responded positively to the course. The lessons and contributions of the students created a vivid and imaginative experience and, in the end, programming became a new technique to tell more about themselves. Students' motivation certainly increased when doing their personal project. The beauty of block-based programming is that everyone can use it without getting distracted by syntactic errors, leaving room to focus on the actual logic behind the algorithms.

This novel approach to programming was particularly relevant for women. Female students were much more talkative than male, and were significantly more active in all the course activities. This could be an explanation for women achieving higher scores than their male counterparts (Fig. 6).

The role of the teacher was also affected by this intervention. Knowledge was not given, it was shared and co-constructed with the students' contributions. BJC is a creative experience that encourages CS teachers to leave their comfort zone, as it comprises methods, exercises, and activities which might not be familiar. This new teacher mindset could be transferred to any other class.

5. Conclusion and Future Work

This experience report summarises the process of adapting and implementing the course "The Beauty and Joy of Computing" in an Engineering in Product Design first-year class. Previous generations of students showed a lack of interest in programming and had a lower-than-average performance in the mandatory CS1 class. After BJC, students were much more responsible and committed than previous cohorts, reported greater satisfaction with the course and got a similar final average than students in the regular CS1. This goes in line with previous findings about the positive impact of programming courses designed for students from majors other than computer science [9], [15].

Contextual factors of BJC affect the teaching and learning process, and should be replicated in other courses. For instance, providing different opportunities for students to demonstrate achievement of course goals, rather than relying upon a single examination, carefully crafting and sizing each evaluation, and developing group norms that support academic honesty, could strengthen the learning process. For these reasons, the role of the teacher is crucial when it comes to students' motivation, confidence and awareness of their own learning process.

After this intervention, some questions arise for future work. To better understand the impact that this intervention has on learning, in a next article the performance of this group is going to be compared to similar students in the regular CS1.

As female students scored a higher final average than male students, it might be interesting to explore if it is possible to replicate this result in a traditional introduction to programming class. Are there any qualitative characteristics of BJC that promote female participation and involvement?

References

[1] J. M. Wing and D. Stanzione, "Progress in Computational Thinking, and Expanding the HPC Community," *Communications of the ACM*, vol. 59, no. 7, pp. 10–11, Jul. 2016.

[2] V. Mulas, C. M. Paradi-Guilford, E. Allende Letona, and Z. V. Dalphond, "Coding Bootcamps : Building Future-Proof Skills through Rapid Skills Training," World Bank Group, Washington, D.C, Tech. Rep., 2017.

[3] W. Aspray, "Recent Efforts to Broaden Informal Computer Science Education," in *Participation in Computing. History of Computing.* Springer, Cham, 2016, pp. 147–163.

[4] L. J. Sax, K. J. Lehman, and C. Zavala, "Examining the enrollment growth: Non-CS majors in CS1 courses," in *Proceedings of the Conference on Integrating Technology into Computer Science Education, ITiCSE.* Seattle, Washington, USA: Association for Computing Machinery, Mar. 2017, pp. 513–518.

[5] K. J. Lehman, A. M. Wofford, M. Sendowski, K. N. Newhouse, and L. J. Sax, "Better Late Than Never: Exploring Students' Pathways to Computing in Later Stages of College," in *Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE*, no. 7. Portland, OR, USA: Association for Computing Machinery, Feb. 2020, pp. 1075–1081.

[6] M. Guzdial and R. K. Hill, "Getting High School, College Students Interested in CS," *Communications of the ACM*, vol. 62, no. 12, pp. 10–11, Dec. 2019.

[7] F. González, C. López, and C. Castro, "Development of Computational Thinking in High School Students: A Case Study in Chile," in *37th International Conference of the Chilean Computer Science Society, SCCC 2018, Santiago, Chile, November 5-9, 2018*. IEEE Computer Society, 2018.

[8] D. F. Shell, L. K. Soh, A. E. Flanigan, and M. S. Peteranetz, "Students' Initial Course Motivation and Their Achievement and Retention in College CS1 Courses," in *Proceedings of the 47th ACM Technical Symposium on Computing Science Education, SIGCSE*. Memphis, Tennessee, USA: Association for Computing Machinery, Feb. 2016, pp. 639–644.

[9] J. Q. Dawson, A. Campbell, M. Allen, and A. Valair, "Designing an Introductory Programming Course to Improve Non-Majors' Experiences," in *SIGCSE 2018 - Proceedings of the 49th ACM Technical Symposium on Computer Science Education*. Baltimore, Maryland, USA: Association for Computing Machinery, Feb. 2018, pp. 26–31.

[10] J. Bennedsen and M. E. Caspersen, "Failure Rates in Introductory Programming: 12 Years Later," *ACM Inroads*, vol. 10, no. 2, pp. 30–35, Apr. 2019.

[11] B. Harvey, "The Beauty and Joy of Computing: Computer Science for Everyone," in *Constructionism 2012: Theory, Practice and Impact*, Athens, Greece, 2012, pp. 33–39.

[12] P. Goldenberg, J. Mark, B. Harvey, A. Cuoco, and M. Fries, "Design Principles behind Beauty and Joy of Computing," in *SIGCSE 2020 - Proceedings of the 51th ACM Technical Symposium on Computer Science Education*. Portland, Oregon, USA: Association for Computing Machinery, Feb. 2020, p. 220–226.

[13] M. Guzdial, B. J. Ericson, T. McKlin, and S. Engelman, "A Statewide Survey on Computing Education Pathways and Influences: Factors in Broadening Participation in Computing," in *ICER'12 - Proceedings of the 9th Annual International Conference on International Computing Education Research*. Auckland, New Zealand: Association for Computing Machinery, 2012, pp. 143–150.

[14] K. Quille, N. Culligan, and S. Bergin, "Insights on Gender Differences in CS1: A Multi-Institutional, Multi-Variate Study," in *Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE*. Bologna, Italy: Association for Computing Machinery, Jun. 2017, pp. 263–268.

[15] M. S. Peteranetz, L. K. Soh, and E. Ingraham, "Building Computational Creativity in an Online Course for Non-Majors," in *SIGCSE 2019 - Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. Minneapolis, MN, USA: Association for Computing Machinery, Feb. 2019, pp. 442–448.