

FACULTAD DE INGENIERÍA

Escuela Académico Profesional de Ingeniería de Sistemas e Informática

Tesis

**Comparison of user experience in online programming
learning platforms: Case study in novice programmers
from Huancayo, Peru**

Gino Joel Taipe Miranda

Para optar el Título Profesional de
Ingeniero de Sistemas e Informática

Huancayo, 2020

Repositorio Institucional Continental
Tesis digital



Esta obra está bajo una Licencia "Creative Commons Atribución 4.0 Internacional" .

Comparison of user experience in online programming learning platforms: Case study in novice programmers from Huancayo, Peru

GINO JOEL TAPE MIRANDA
Continental University

PEDRO YURI MARQUEZ SOLIS
Continental University

ABSTRACT

This research was comparison of user experience between the Scratch and Code.org platforms, for being used by novice programmers. The comparison was taking into seven factors that influence user experience, including usefulness, usability, desirability, findability, accessibility, credibility, and value. The research was carried out with the participation of 40 people with little or no knowledge about programming topics, the people carried out the following tasks: Register on the platform, use a sample project, create a project, program using block-based programming, explore the features and functionalities of the platform. In general, both platforms offer a good user experience for novice programmers. Finally, it is concluded that Scratch offers a better user experience, because it exceeded Code.org in 4 of the 7 factors.

CCS CONCEPTS

• **Human-centered computing**; • **Empirical studies in accessibility**;

KEYWORDS

User experience (UX), programming, block-based programming, scratch, code, learning platform

ACM Reference Format:

GINO JOEL TAPE MIRANDA and PEDRO YURI MARQUEZ SOLIS. 2020. Comparison of user experience in online programming learning platforms: Case study in novice programmers from Huancayo, Peru. In *2020 12th International Conference on Education Technology and Computers (ICETC'20)*, October 23–26, 2020, London, United Kingdom. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3436756.3437013>

1 INTRODUCTION

Nowadays, computing has a great importance, and the skills related to that field are of great value in these times, in [2] the ten skills they should have at work by the year 2020 are described, and for several years, the teaching of programming in children, adolescents and young people has been promoted.[15]

In a study carried out by Dayna Steele [11] it is reported that since 2017 generation Z began to enter the workforce. This indicates

that the most technologically prepared generation is already in the labor market, but according to a Microsoft study [1] shows that 60% of students belonging to generation Z don't know what the term programming means, considering that, according reports by Deloitte [3], among the technical skills necessary for an industry 4.0, is having knowledge of computer programming, coding, etc.

The pandemic caused by SARS-CoV-2 has promoted the use of virtual platforms, causing them to grow exponentially [10]. Among all the existing platforms on the market we have those whose main objective is to learn programming-related topics. Some of them through text-based programming, some through block-based programming, and some by combining both.

Taking the above into consideration, the way you programming visually using blocks allows novice programmers to better understand programming concepts, in addition to allowing them to perform the tasks that may be assigned to them faster.[9] In [12] it is reported that what contributes to the ease of use of block-based programming are the natural language description of the, the drag-and-drop composition interaction, and the ease of navigating the language. In addition, in [13] block-based programming and text-based programming are compared, and it showed that students who programmed using blocks have a greater interest in programming and in knowing more about future computer courses.

Some platforms that offer block programming are Blockly games, MIT App Inventor, Code.org, Microsoft MakeCode, CodeBug, Scratch, mBlock, etc. [14] Among all this variety of platforms, Scratch has more than 56 million registered accounts [16] and Code.org has more than 37 million registered accounts [17], therefore both platforms were compared in the present research.

If the programming learning platforms were designed and developed without considering the user experience (UX), they could cause novice programmers lose interest in programming topics because they don't understand how the virtual platform works. The assessment the UX that Scratch and Code.org platforms have in novice programmers is unknown, as can be seen in section 3. In the research there will be comparison user experience of the Scratch and Code.org platforms, in novice programmers from the city of Huancayo. The comparison was taking into seven factors that influence user experience [5], considering the participation of 40 people, divided into two groups of 20 people [6]. The research re-sult showed that the Scratch platform offers a better user experience compared to Code.org.

This study didn't consider the learning contents of the platforms under study, it focused on assessing the user experience on the Scratch and Code.org platforms. Considering this, some limitations that the study had was access to people, because the pandemic caused by SARS-CoV-2 made it difficult to recruit people to support the research. Consequently, the sample size could not be larger,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
ICETC'20, October 23–26, 2020, London, United Kingdom
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-8827-6/20/10... \$15.00
<https://doi.org/10.1145/3436756.3437013>

although it was possible to recruit the minimum number of participants per platform [6], which allowed obtaining statistically significant results in the present research work.

2 FACTORS THAT INFLUENCE USER EXPERIENCE (UX)

User experience (UX) encapsulates all aspects of the end-user interaction with a company, its services and products.[7]

Peter Morville [5] describes seven factors that influence UX, what he called User Experience Honeycomb. The seven factors are: utility, usability, desirability, findability, accessibility, credibility, and value. In [5] each is defined. Utility refers to the fact that the product must have a purpose that meets the needs of the user. Usability, the user must achieve their goal effectively and efficiently. Findability, as the user must easily find the content of the product. Credibility, for the user's confidence in the product they are using. Desirability, to aesthetics and emotional design. Accessibility, that a variety of users with different skills; this includes people with a disability, they can use the product. Value, because the product must deliver some value to the company that created it and to the user who uses it.

3 RELATED WORKS

A search for similar research to the present research was carried out, resulting in [8] a first comparison between Scratch and App Inventor is made, the research focuses on comparing general features of both platforms, such as usability, frequency of problems occurred during use, supported operating system, etc. As a result, it was obtained, that it is recommended that students use Scratch first and then App Inventor. The study of [8] did not consider the user experience offered by both platforms.

In [4] the usability of Scratch was evaluated in novice programmers. As a result, three usability problems were found. The first was that the icons and texts were very small, and this caused frustrations in the users. The second was the lack of user feedback, which led the research participants to spend a significant amount of time searching for features and functionality. The third problem, was the need to switch between different sprites to visualize program execution, debugging and synchronization, which obstructed users' ability to resolve program bugs.

It can be seen that, in the research found, they evaluate the usability of the platforms, but did not carried out a more in-depth research of the user experience that these platforms offer.

4 METHOD

For the research, people in the age range belonging to the so-called generation Z were recruited, people who were born between 1996 and 2012, with little or no knowledge of programming topics. The research was carried out over three weeks in July 2020, during which virtual sessions were held with each of the people participating in the research.

To know the user experience on each of the platforms, the participants were randomly divided into two groups, each group made up of 20 people. One group interacted with the Scratch platform and the other group with the Code.org platform.

During the individual virtual sessions carried out with the members of each group, the tasks they had to carry out were explained to them, the participants performed the same tasks on both platforms. These tasks were: Register on the platform, select an example project that was on the platform, create your own project, program using block-based programming, explore the features and functionalities of the platform.

At the end of the tasks, the research participants answered a survey about their experience in the respective platform they used, said survey was prepared taking the seven factors that influence user experience. The following table (see Table 1) shows the evaluation criteria and level of compliance considered for each factor that influences user experience.

5 RESULTS

Based on the surveys, observations and comments by the participants about their experience on the platforms. A qualitative and quantitative analysis of the data obtained was carried out.

5.1 Quantitative Analysis Based on Achieve of Tasks

A comparative was graph between both platforms (see Figure 1), in which the fulfilment of each of the tasks performed by the research participants is appreciated. Table 2 shows the order and description of the tasks that were performed by the participants: 1) Register on the platform. 2) Use a sample project. 3) Create their own project. 4) Program using block-based programming. 5) Explore the features and functionalities of the platforms.

Figure 1 shows that the most difficult task was block-based programming, which consisted in that each time the Sprite was clicked / tapped, it would move down. On the Code.org platform, none of the participants managed to complete the task, and on the Scratch platform only three participants managed to complete the requested program. In the tasks of registering on the platform and exploring features and functionalities, the same number of participants managed to perform these tasks. Obviating these two tasks, and making a comparison between the other tasks, it is seen that Scratch surpasses Code.org.

5.2 Qualitative Analysis

The following table (see Table 3) shows a qualitative comparison of the user experience between both platforms. The average (avg) of the data obtained in the survey was obtained, as well as a total average of each of the platforms

It is worth mentioning that in [5] which factor is more important depends on its unique balance of context, content and users. Therefore, for the present research it was considered that all the factors have the same weighting.

The following points briefly describe the results in each of the factors that influence the user experience, taking into account the level of compliance in table 1

5.2.1 Useful. The useful of the platform to learn and understand basic programming concepts was considered. Resulting that both platforms are equally (avg = 2.60) useful, considering 2.60 closer to very useful for users. Some of the participants commented that

Table 1: Evaluation criteria of the factors that influence UX

Factor	Evaluation criteria	1	Level of compliance 2	3
Useful	The usefulness of the platform to learn and understand basic programming concepts	Little useful	Useful	Very useful
Usable	The number of tasks achieved	Less than two tasks achieved	Most tasks achieved	All tasks achieved
Desirable	The probability of re-entering the platform	Unlikely to re-enter	Likely to re-enter	Very likely to re-enter
Findable	The ease to navigate and find functions on the platform	Not finding the functions of the platform	Difficulties finding the functions of the platform	Easily find functions of the platform
Accessible	The level of accessibility of the platform for people with different capacities	Not very accessible	Accessible	Very accessible
Credible	The level of trust of the content offered by the platform	Unreliable	Trustworthy	Very reliable
Value	The contribution of the platform to improve programming knowledge	Disagreement	Neither agree or disagree	Agree

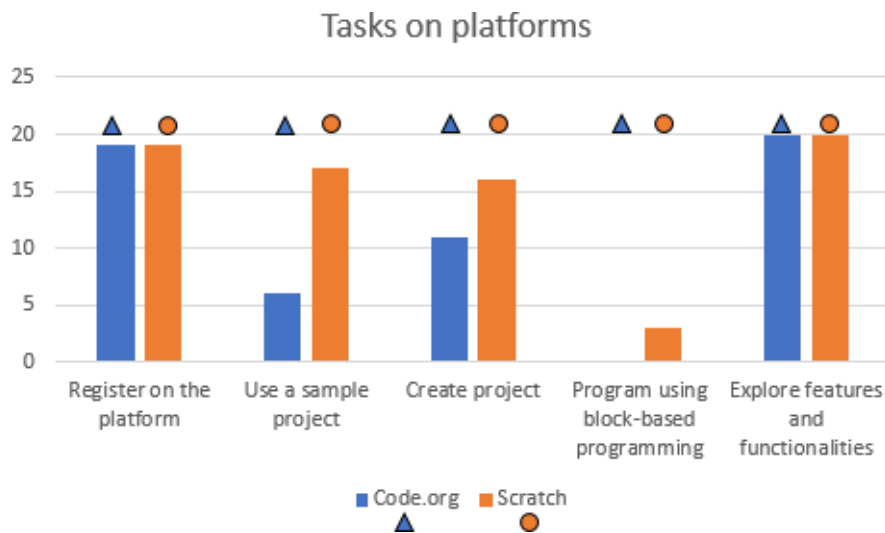


Figure 1: Comparison of tasks by platform.

Table 2: Legend of task

Numerical value	Task	Task description
1	Register on the platform	The participants register and log into the platform
2	Use a sample project	The participants choose one of the example projects that the platform offers
3	Create project	The participants create their own project
4	Program using block-based programming	The participants perform a basic sequential program, which consists in that each time he clicks / taps the Sprite, it moves down
5	Explore features and functionalities	The participants freely navigate through all the functions and features offered by the platform

Table 3: UX at Code.org and Scratch

Factor	Code.org (avg)	Scratch (avg)
Useful	2.60	2.60
Usable	1.60	2.00
Desirable	2.30	2.50
Findable	2.25	2.35
Accessible	2.00	2.30
Credible	2.50	2.20
Value	2.95	2.90
Total average	2.31	2.41

they found the platform they tested useful, but that they would need someone to guide them in using these platforms.

5.2.2 Usable. The number of tasks achieved was considered an evaluation criterion of the factor. Resulting, Scratch participants (avg = 2.00) performed most of the requested tasks efficiently, compared to Code.org (avg = 1.60) that participants were unable to achieve most of the tasks. In addition, since on the Code.org platform the participants had complications with the English language, and it was difficult for them to find the option to switch languages, so they could not efficiently perform the requested tasks. On the Scratch platform, most of the participants had no major difficulties to change the language of the platform, being able to complete the requested tasks efficiently.

5.2.3 Desirable. The probability of re-entering the platform was considered as evaluation criterion of the factor. Resulting, users are highly likely to re-enter Scratch (avg = 2.50) while Code.org (avg = 2.30) is only likely. The aesthetics of the user interface and the language of the platform were the most mentioned factors for not re-entering the platform.

5.2.4 Findable. The ease of navigating and finding functions such as registering on the platform, selecting a sample project, creating a sample project, programming using block-based programming, was considered as evaluation criteria of the factor. Resulting that the participants placed the Scratch functions (avg = 2.35) with some drawbacks, but in a simpler way, compared to Code.org (avg = 2.25) where the participants also had drawbacks. It was observed that the few options that Scratch offers to navigate allowed a better performance of the research participants.

5.2.5 Accessible. The level of accessibility of the platform for people with different abilities was considered as an evaluation criterion of the factor. Resulting in that Scratch (avg = 2.30) is perceived to be closer to very accessible compared to Code.org (avg = 2.00). Participants commented that platforms should continue to improve accessibility so that people with different abilities (auditory, motor, visual, cognitive, etc.) use the platforms appropriately.

5.2.6 Credible. The level of trust of the content offered by the platform was considered as an evaluation criterion of the factor. Resulting, Scratch (avg = 2.20) and Code.org (avg = 2.50) offer highly reliable content, giving users the confidence that they can learn programming concepts correctly. Some participants commented

that they found Code.org trustworthy because it is supported by various technology organizations.

5.2.7 Valuable. The contribution of the platform to improve programming knowledge was considered as a factor evaluation criterion. Resulting, participants agree that Scratch (avg = 2.90) offers them great value just like Code.org (avg=2.95). It is worth mentioning that the participants perceived that Code.org would allow them to improve their knowledge and programming skills through the learning path offered by this platform.

6 CONCLUSIONS

This research showed that the platform that offers a better user experience to novice programmers in the city of Huancayo is Scratch, because it exceeded Code.org in 4 out of 7 factors, and the latter only surpassed Scratch in 2 factors. In addition to that in the total average Scratch also surpasses Code.org. It is worth mentioning that although Scratch surpassed Code.org both in the analysis of qualitative and quantitative results, Scratch is not a perfect platform and that it could improve the user experience that it currently offers to its users.

Online learning platforms, should comply with the 7 factors that influence the user experience so that students remain motivated when using said platforms, and not only them, but also the public in general. In addition, it also allows the companies in charge of these platforms to differentiate themselves from the rest of the existing platforms.

The little knowledge of the participants in programming topics shows that the educational institutions of the city of Huancayo still do not implement the teaching of programming topics in their educational curricula or that the teachers are not sufficiently trained to teach that course, a research work more depth about it could show the true causes. This research is expected that of this research can help the educational institutions of the city of Huancayo take more into consideration the teaching of programming to their students.

ACKNOWLEDGMENTS

We thank the students of the different educational institutions of primary, secondary and higher levels of the city of Huancayo, for the support provided in this research.

REFERENCES

- [1] Marcelo Belluci. 2016. Generación Z: Se creen expertos en computación pero saben muy poco - Clarín. Retrieved July 21, 2020 from https://www.clarin.com/sociedad/generacion-creen-expertos-computacion-saben_0_E1RvFD11W.html
- [2] Ana Davies, Devin Fidler, and Marina Gorbis. 2011. IFTF: Future Work Skills 2020. Retrieved July 22, 2020 from <https://www.iftf.org/futureworkskills>
- [3] Deloitte. 2018. Preparing tomorrow's workforce for the Fourth Industrial Revolution | Deloitte | About. Retrieved July 19, 2020 from <https://www2.deloitte.com/global/en/pages/about-deloitte/articles/gx-preparing-tomorrow-workforce-for-the-fourth-industrial-revolution.html>
- [4] Yerika Jimenez, Amanpreet Kapoor, and Christina Gardner-McCune. 2018. Usability challenges that novice programmers experience when using scratch for the first time. In Proceedings of IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC, IEEE Computer Society, 327–328. DOI:<https://doi.org/10.1109/VLHCC.2018.8506560>
- [5] Peter Morville. 2004. User Experience Design. Retrieved July 15, 2020 from http://semanticstudios.com/user_experience_design/
- [6] Jakob Nielsen. 2012. How Many Test Users in a Usability Study? Retrieved July 13, 2020 from <https://www.nngroup.com/articles/how-many-test-users/>

- [7] Don Norman and Jakob Nielsen. The Definition of User Experience (UX). Retrieved July 12, 2020 from <https://www.nngroup.com/articles/definition-user-experience/>
- [8] Stamatios Papadakis, Michail Kalogiannakis, Vasileios Orfanakis, and Nicholas Zaranis. 2014. Novice programming environments. Scratch & app inventor: A first comparison. In ACM International Conference Proceeding Series, Association for Computing Machinery, New York, New York, USA, 1–7. DOI:<https://doi.org/10.1145/2643604.2643613>
- [9] Thomas W. Price and Tiffany Barnes. 2015. Comparing textual and block interfaces in a novice programming environment. In ICER 2015 - Proceedings of the 2015 ACM Conference on International Computing Education Research, Association for Computing Machinery, Inc, New York, New York, USA, 91–100. DOI:<https://doi.org/10.1145/2787622.2787712>
- [10] Milagros Quispe. 2020. Educación virtual: la gran oportunidad que ha dado el COVID-19 | Noticias | Universidad Nacional Mayor de San Marcos | UNMSM. Retrieved July 18, 2020 from <http://www.unmsm.edu.pe/noticias/ver/Educacion-virtual-la-gran-oportunidad-que-ha-dado-el-COVID-19>
- [11] Dayna Steele. The successful generation on your team. Retrieved July 16, 2020 from <https://www.microsoft.com/en-us/microsoft-365/business-insights-ideas/resources/the-successful-generation-on-your-team>
- [12] David Weintrop and Uri Wilensky. 2015. To block or not to block, that is the question: Students' perceptions of blocks-based programming. In Proceedings of IDC 2015: The 14th International Conference on Interaction Design and Children, Association for Computing Machinery, Inc, New York, New York, USA, 199–208. DOI:<https://doi.org/10.1145/2771839.2771860>
- [13] David Weintrop and Uri Wilensky. 2017. Comparing block-based and text-based programming in high school computer science classrooms. ACM Trans. Comput. Educ. 18, 1 (October 2017), 1–25. DOI:<https://doi.org/10.1145/3089799>
- [14] 2019. Blockly | Google Developers. Retrieved July 14, 2020 from <https://developers.google.com/blockly>
- [15] 2020. El lenguaje de programación llega a las aulas | Fundación Telefónica España. Retrieved July 22, 2020 from <https://www.fundaciontelefonica.com/noticias/importancia-aprender-lenguaje-programacion/>
- [16] 2020. Scratch Statistics - Imagine, Program, Share. Retrieved July 10, 2020 from <https://scratch.mit.edu/statistics/>
- [17] 2020. Statistics | Code.org. Retrieved July 10, 2020 from <https://code.org/statistics>