A Dataset of Scratch Programs: Scraped, Shaped and Scored
Aivaloglou, Efthimia; Hermans, Felienne; Moreno-León, Jesús; Robles, Gregorio

DOI
10.1109/MSR.2017.45

Publication date
2017

Published in
Proceedings of the 14th International Conference on Mining Software Repositories

Citation (APA)

Important note
To cite this publication, please use the final published version (if applicable). Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights. We will remove access to the work immediately and investigate your claim.
A Dataset of Scratch Programs: Scraped, Shaped and Scored

Efthimia Aivaloglou, Felienne Hermans, Jesus Moreno-Leon, and Gregorio Robles
A Dataset of Scratch Programs: Scraped, Shaped and Scored

Efthimia Aivaloglou*, Felienne Hermans*, Jesús Moreno-León†, and Gregorio Robles‡

*Delft University of Technology, The Netherlands
Email: {e.aivaloglou, f.f.j.hermans}@tudelft.nl
†Programamos.es & Universidad Rey Juan Carlos, Spain
Email: jesus.moreno@programamos.es
‡Universidad Rey Juan Carlos, Fuenlabrada (Madrid), Spain
Email: grex@gsyc.urjc.es

Abstract—Scratch is increasingly popular, both as an introductory programming language and as a research target in the computing education research field. In this paper, we present a dataset of 250K recent Scratch projects from 100K different authors scraped from the Scratch project repository. We processed the projects' source code and metadata to encode them into a database that facilitates querying and further analysis. We further evaluated the projects in terms of programming skills and mastery, and included the project scoring results. The dataset enables the analysis of the source code of Scratch projects, of their quality characteristics, and of the programming skills that their authors exhibit. The dataset can be used for empirical research in software engineering and computing education.

I. INTRODUCTION

Scratch [1] is a block-based programming language developed to serve as a stepping stone for children from 8 to 16 years old to the more advanced world of computer programming. It offers a web-based programming environment that enables creating games and interactive animations. The public repository of Scratch programs contains over 19 million projects and 16 million users.

A number of works in the computing education research field attempt to assess the programming skills that novice programmers develop in the Scratch environment. Some utilize program data collected during specific programming courses (e.g., [2], [3], [4]), while others utilize the dataset made available by the Lifelong Kindergarten Group at the MIT Media Lab [5], which contains data for Scratch projects created until 2012 (e.g., [6], [7], [8]). In addition to identifying indications of learning of programming concepts, static analysis of Scratch programs has also been performed for identifying code smells and bad programming practices [9], [10], and automated quality assessment tools have been proposed (e.g., Hairball [11] and Dr. Scratch [12]).

While Scratch is receiving increasing interest as an introductory programming language, there is no recent dataset of Scratch programs available to the research community. The one made available from the MIT Media Lab concerns projects created using the previous, initial version of the Scratch application, before the introduction of the current Scratch version (Scratch 2) and of the web programming interface in 2013. It is since then that the popularity of Scratch started to increase.1 Scratch 2 introduced several features relating to important programming concepts,2 like custom blocks (the equivalent to procedure definitions), and therefore that dataset does not include projects utilizing those features.

The goal of this paper is to present an open and timely dataset of recent Scratch programs, along with their metadata, that can facilitate quantitative research in the fields of source code analysis and computing education. The dataset contains 250,000 Scratch projects, from more than 100,000 different users, that were scraped from the Scratch project repository. It is made available as a database3 which includes, for each Scratch project, its metadata and the program data, along with programming mastery scoring results from the Dr. Scratch quality assessment tool [12].

II. DATASET CONSTRUCTION

A. Data Collection

To collect the data from the web interface of the Scratch project repository, we built a scraping program. The web scraping program starts by reading the Scratch projects page4 and thus obtains the project identifiers of projects that were most recently shared. Subsequently, it retrieves a JSON file for each of the listed projects5.

We ran the scraper on March 2nd 2016 for 24 hours and, during that time, it obtained the JSON files for 250,163 projects. Out of those, we failed to parse and further analyze 2,367 projects due to technical difficulties with the project's JSON representation. We then obtained the project identifiers of those projects that were not successfully parsed by the scraper and further failed to parse 57 projects due to technical difficulties with the project's JSON representation. We thus retrieved the JSON files for 247,219 projects.

Once we obtained the Scratch projects, we parsed the JSON files for each of the listed projects. This resulted in a list of used blocks per project, within the sprites and the stage of the project. We also cross referenced all blocks

1Monthly activity trends can be found at https://scratch.mit.edu/statistics/
2The complete list of features introduced in Scratch 2 can be found at https://wiki.scratch.mit.edu/wiki/Scratch_2.0
3https://github.com/TUDelftScratchLab/ScratchDataset
4https://scratch.mit.edu/explore/projects/all/
5For a given project id x, the program's JSON representation can be obtained via https://cdn.projects.scratch.mit.edu/internalapi/project/x/get
6http://wiki.scratch.mit.edu/wiki/Scratch_File_Format_(2.0)
with the Scratch wiki to determine their shapes and category. For example, When Green Flag Clicked is a Hat block from the Events category. We included blocks from Scratch extensions, such as the LEGO WeDo extensions, that were found in the dataset.

B. Calculation of Programming Mastery Scores

We analyzed the projects with Dr. Scratch, a tool that statically inspects Scratch projects’ source code to assign scores on seven dimensions of computational thinking: abstraction and problem decomposition, logical thinking, synchronization, parallelism, algorithmic notions of flow control, user interactivity and data representation. These dimensions are given a value from 0 to 3. The mastery score is the aggregated punctuation of the seven dimensions, and therefore ranges from 0 to 21.

Dr. Scratch also detects: intra-project software cloning, i.e., repetition of code; the use of custom blocks, which is the way functionality can be reused in Scratch programs; and the use of instances of sprites, a feature labeled in Scratch as clone creation. These values are included in the dataset.

Out of the 250,163 projects, 231,050 were successfully analyzed with Dr. Scratch. This was due to problems with non-official extension blocks and with some non-ASCII characters in sprites names.

C. Importing the Data

All scraped project data and metadata, including the list of used blocks and parameters, were imported in a relational database. We also imported the data on the shapes and the categories of the Scratch blocks. We then used SQL queries for normalizing the data and bringing it in its final schema, outlined in Table I.

III. DATASET DESCRIPTION

A. Data Representation

The projects’ data are stored in a relational database. Each of the projects is identified by its Scratch project ID, stored in field p_id, while its author is identified by the username. If a project is a remix of another one, the original project can be found in the remixes table. Table grades stores the Dr. Scratch results for the programming mastery metrics per project.

The schema of the database, outlined in Table I, reflects the structure of the Scratch programs. Projects contain sprites, which are entities with their own associated code. The code is organized into scripts, i.e., groups of Scratch code blocks, each script belonging to a sprite named sprite-name. In the example in Figure 1 there are 3 scripts, one initiated when the green flag is clicked, one then the sprite is clicked, and the ‘backflip’ custom block definition. Those custom blocks are the equivalent of procedure definitions, with their names and arguments stored in the procedures table.

![Fig. 1. Example of a Scratch program with three scripts in the same sprite](image)

*TABLE I*  
DATABASE SCHEMA: TABLES AND ATTRIBUTES

---

7Each project and author page can be accessed in the Scratch web interface at https://scratch.mit.edu/projects/(p_id) and https://scratch.mit.edu/user/(username) respectively.
The evaluation of block-based languages in general, and Scratch in particular, as tools for programming education is receiving significant research attention. A number of studies have been carried out during the last years on understanding the programming practices of learners in those environments, on the programming skills they develop, and on the quality of their programs.

The first research direction that this dataset can be utilized for concerns the assessment of the programming skills that novice programmers develop in the Scratch environment. The dataset can be used for examining indications of learning of programming concepts and abstractions, as in [10]. Existing works in this research direction include the work by Maloney et al. [4], who analyzed 536 Scratch projects created in an after-school clubhouse for blocks that relate to programming concepts and found that within the least utilized ones are Boolean operators and variables. Another study on the internalization of programming concepts with Scratch with 46 students was presented in [2], where it was found that students had problems with initialization, variables and concurrency.

Towards this research direction, the dataset makes the source code of the Scratch programs available for static analysis and can be used for quantitatively evaluating the application of programming concepts and abstractions through the presence of Scratch blocks of the corresponding types.

Another promising research direction relates to the quality of the programming artifacts developed in the Scratch environment. The dataset has already been used for quality assessment, for the identification of programming smells [10], and for exploring indications of harmful programming habits [13]. Existing works in this direction include a study in a classroom setting [9], which highlighted two bad programming habits in Scratch, namely bottom-up development and extremely fine-grained programming. The authors connected the later to the reduced use of if-blocks and finite loops and the increased use of infinite loops. Related to smell detection are the Scratch automated quality analysis tools Hairball [11], a static analysis tool that can detect initialization problems and unmatched broadcast and receive blocks, and Dr. Scratch [14], which extends Hairball to detect two bad programming habits: not changing the default object names and duplicating scripts.

This dataset will enable examining code quality and code smells on a large set of Scratch projects. Static analysis of the source code can also be performed for the identification of other types of smells [15] that might be common in the artifacts of novice programmers.

Moving from the software engineering to the computing education research field, a research direction with increased data requirements concerns the learning progressions of novice programmers. This includes examining the factors that support learning and improving their programming skills. Existing works in this area have used the dataset of the previous version of Scratch programs, created until 2012, and described in the Introduction. They include the works of Dasgupta et al., who
investigated how project remixing relates to the adoption of
new computational thinking concepts [8], and Yang et al., who
examined the learning patterns of programmers in terms of
block use over their first 50 projects [7].

Research towards this direction requires extending the
dataset with the complete set of projects for the included users,
along with their creation dates. This would enable examining
how their programming skills and mastery evolved through
time, possibly through remixing the projects of others. An even
richer extension of the dataset would be with periodical snap-
shots of the included projects that are still in development. This
would enable reconstructing project versioning information,
not available from the Scratch interface, and thus examining
how the projects are developed and evolve over time.

Apart from quantitative studies, the dataset can support the
design of experiments and field studies, whose material often
includes Scratch programs with specific characteristics. For
example, in [16] examples of Scratch programs exhibiting spe-
cific smells were used in a controlled experiment to determine
how the smells affect the understanding and the modification
of the programs. This queryable dataset can support finding
example programs with characteristics according to the exper-
iment design and the field of the study.

V. LIMITATIONS

The only information about the authors of the Scratch
projects contained in the dataset is their username. This
suffices for creating project portfolios of the authors and for
facilitating research in the directions described in the previous
section. However, other directions in the computing education
research field require richer author data, and especially data
on their gender and age. There are, for example, indications
that specific programming concepts are better understood
after certain ages —Seiter and Foreman [17] analyzed 150
Scratch projects from primary school students and found that
design patterns requiring the understanding of parallelism,
conditionals and, especially, variables were under-represented
by all grades apart from 5 an 6. The effect of gender and
account age were also examined in [6] in relation to the use
of programming concepts. However, this dataset does not include
information on the gender and the age of authors and cannot
be extended to do so using the current Scratch web interface
because this information is not available in the user profile
data2, even though it is provided by users upon registration.

Another limitation of this dataset is that we did not scrape
a random sample of Scratch projects, but the most recent ones
during the time that we run the scraper. It could be the case
that the programming habits of Scratch users are changing over
time. However, we counterbalanced that by collecting a
large dataset which comprises around 1.3% of all 19 million
shared Scratch projects, and is the most recent one to be made
available to the research community. Projects in the dataset are
the ones that their authors shared publicly when the scraper
run. The dataset contains cases of projects that are no longer
publicly shared and are thus no longer accessible through the
Scratch web interface.

VI. CONCLUSION

We presented a dataset of recent Scratch programs scraped
from the Scratch project repository. It is made available as a
database3 which includes the source code of the Scratch
projects, their metadata, and their programming mastery scor-
ing results. The dataset can facilitate research in software
engineering and computing education, for topics like the
assessment of the programming skills that novice programmers
develop, the exploration of their learning progressions, and
issues related to the quality of the programs, such as the
identification of smells and bad programming habits.

REFERENCES

K. Brennan, A. Millner, E. Rosenbaum, J. Silver, B. Silverman, and
Computer Science Concepts with Scratch,” in Proceedings of ICER
games developed by primary school children to gauge understanding
of programming concepts” in European Conference on Games Based
gramming by choice: Urban youth learning programming with Scratch,”
years of public activity in the Scratch online community,” Scientific
in youth computational participation in the online Scratch community,”
C. Beckley, and A. Johri, “Uncovering trajectories of informal learning
in large online communities of creators,” in Proceedings of the Second
ACM Conference on Learning @ Scale. ACM, 2015, pp. 131–140.
as a pathway to Computational Thinking,” in Proceedings of the 19th
ACM Conference on Computer-Supported Cooperative Work & Social
[9] O. Meerbaum-Salant, M. Armoni, and M. Ben-Ari, “Habits of Program-
ming in Scratch,” in Proceedings of the 16th Annual Joint Conference
on Innovation and Technology in Computer Science Education, ser. ITiCSE
exploratory study on the Scratch repository,” in Proceedings of ICER
ball: Lint-inspired Static Analysis of Scratch Projects,” in Proceeding of
habits in Scratch: A preliminary study,” in 2014 IEEE Frontiers in
clones in Scratch projects: On the presence of copy-and-paste in Com-
putational Thinking learning.” in Proceedings of the 11th International
matic Analysis of Scratch Projects to Assess and Foster
Computational Thinking,” RED - Revista de Educación a Distancia,
[16] F. Hermans and E. Aivaloglou, “Do code smells hamper novice program-
ing? A controlled experiment on Scratch programs,” in 2016 IEEE 24th
[17] L. Seiter and B. Foreman, “Modeling the learning progressions of
Computational Thinking of primary grade students,” in Proceedings of