



King's Research Portal

DOI:

[10.1145/3641555.3705129](https://doi.org/10.1145/3641555.3705129).

Document Version

Peer reviewed version

[Link to publication record in King's Research Portal](#)

Citation for published version (APA):

Messer, M., Brown, N., Kölling, M., & Shi, M. (in press). Menagerie: A Dataset of Graded Programming Assignments. In *In Proceedings of the 56th ACM Technical Symposium on Computer Science Education* (Vol. 2) <https://doi.org/10.1145/3641555.3705129>.

Citing this paper

Please note that where the full-text provided on King's Research Portal is the Author Accepted Manuscript or Post-Print version this may differ from the final Published version. If citing, it is advised that you check and use the publisher's definitive version for pagination, volume/issue, and date of publication details. And where the final published version is provided on the Research Portal, if citing you are again advised to check the publisher's website for any subsequent corrections.

General rights

Copyright and moral rights for the publications made accessible in the Research Portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognize and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the Research Portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the Research Portal

Take down policy

If you believe that this document breaches copyright please contact librarypure@kcl.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.

Menagerie: A Dataset of Graded Programming Assignments

Marcus Messer
Department of Informatics
King's College London
London, UK
marcus.messer@kcl.ac.uk

Michael Kölling
Department of Informatics
King's College London
London, UK
michael.kolling@kcl.ac.uk

Neil C. C. Brown
Department of Informatics
King's College London
London, UK
neil.c.c.brown@kcl.ac.uk

Miaoqing Shi
College of Electronic and Information Engineering
Tongji University
Shanghai, China
mshi@tongji.edu.cn

ABSTRACT

We present Menagerie, an open-ended project-scale second-semester CS1 Java assignment dataset that ran over four academic years (18/19 - 21/22). It comprises 667 submissions, with 273 being subsequently graded *post hoc*. The assignment was open-ended, with the students being asked to implement a 'predator/prey' simulator and to meet specific criteria, which included adding five species, competing for the same food source and keeping track of the time of day. The submissions were assessed as part of a separate study on the correctness of the solution, how well the code was designed, how readable the code is, and the quality of the documentation.

ACM Reference Format:

Marcus Messer, Neil C. C. Brown, Michael Kölling, and Miaoqing Shi. 2018. Menagerie: A Dataset of Graded Programming Assignments. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 2 pages. <https://doi.org/XXXXXXXX.XXXXXXX>

1 INTRODUCTION

Education datasets are essential to research education and developing interventions and software to aid student learning [7]. Data from tools such as intelligent tutoring systems, education games and simulations allow education researchers to address questions on many aspects of students' learning, including cognition, metacognition, and motivation [8]. While several datasets of students' programming assignments exist, many are not publicly accessible [9], consist of small-scale programs [2, 5] or do not capture contextual details [4].

Most publicly available programming education datasets are small-scale and focus on a single task. In these smaller-scale datasets, including FalconCode [5], Dublin City University's Programming

Submission Dataset [2], Hour of Code¹, and online judge tools, such as HackerRank², the task is known. However, all of these datasets are closed-ended and focus on small-scale tasks, including highly structured template code, specific names of classes and functions defined in the requirements, or simple one-function tasks, such as FizzBuzz or the Rainfall Problem [12].

Some project-scale datasets exist, such as the Scratch website scenarios³ or the Blackbox dataset [4] that are public or available on request. However, they have the disadvantage that the context of the users is unknown (they may be students at any stage of education or not even students at all), and the aim of their project is unknown, preventing judgment against any criteria for progress or success.

We present the Menagerie dataset – a publicly available dataset containing 667 real student submissions of an open-ended, project-scale CS1 Java programming assignment, 273 of which were subsequently graded as part of an independent study.

2 THE DATASET

The Menagerie dataset comprises 667 real student submissions for King's College London's second semester CS1 Introduction to Programming Java course and is available publicly. [1]. The assignment was given to students between 2017 and 2021 inclusive, with minor changes each year. The changes each year included updating the assignment instructions to clarify points of misunderstanding from the previous years' assignments, with the tasks and the provided template code remaining the same year on year.

The dataset consists of the following:

- The assignment requirements and grading rubric.
- The project's starter code.
- The students' Java files.
- Grades and associated feedback for 273 submissions (produced *post-hoc* as part of an independent study).

The dataset does not include the students' original awarded grades, as we did not receive ethical approval to release them publicly. However, as part of an independent study, we produced grades and feedback for 273 submissions.

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.
Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06...\$15.00
<https://doi.org/XXXXXXXX.XXXXXXX>

¹Hour Of Code: <https://code.org/research>

²HackerRank: <https://www.hackerrank.com/>

³Scratch Scenarios: <https://scratch.mit.edu/ideas>

We have received ethical approval from our institution⁴ to publicly release the dataset of student work and the *post-hoc* grades.

2.1 The Assignment

The assignment was a small-group, open-ended paired programming assignment to utilise object-oriented programming concepts to develop a predator/prey simulator with groups of two or three. The students were provided with a template project based on the “foxes-and-rabbit” project from the sixth edition of Barnes and Kölling [3]’s book on “Objects First with Java”.

Students were asked to extend the template code with the four base tasks, including adding five species, two predators and two prey, and the simulation should keep track of the time of day. After completing the base tasks, the students were asked to implement one or more challenge tasks. They could choose to invent their own tasks or one of the provided suggestions, which included simulating plants, weather or disease. Full details on the assignment are given in detail on the Open Science Foundation [1].

During the assignment, students could choose which species they implemented, how those interacted with other species, and how their implementation completed the provided tasks. The ability of the students to design their solution to a problem and the open-ended nature of the assignment allows instructors to assess a wider variety of skills than they typically could in a close-ended small-scale assignment like the Rainfall problem [6, 12].

2.1.1 Dataset Limitations. The students were asked to conduct pair programming with explicit instructions on how to pair program. However, anecdotally, some groups started pair programming, but as the submission deadline approached, groups tended to separate the work, with students writing code in parallel.

These are historical assignments that we received *post hoc* permission to release, the students’ demographics were not captured. However, all students were in their first year at King’s College London and were a mixture of domestic and international students.

2.2 Grading and Feedback

We asked 28 graders to produce the grades and feedback included within this dataset. Each assignment was graded four times, each time by a different grader. Most of our graders identified as male, with only five identifying as female, which follows a similar distribution of students enrolled in the computer science courses at our institution. They were either PhD, Masters or final-year undergraduate students, with most participants being Masters students. Across many institutions, PhD, Master’s, and undergraduate students participate in the assessment [10, 11].

Graders were asked to grade and give feedback on the programming skills; grades were between A and F, with ‘+’ and ‘-’ grades, with exception submissions being awarded A++:

- **Correctness:** Has the submission met the requirements of the assignment?
- **Code Elegance:** Has the code followed the design principles in the course, including polymorphism and inheritance?
- **Readability:** Is the code well organised, appropriately indented, and uses meaningful identifier names?

- **Documentation:** Are the docstrings well-written, organised, and clearly explain what the code accomplishes?

3 POTENTIAL USES

This dataset can be used for many computer science education research purposes, such as:

- Applying large-language models to novices’ programs for hint generation or code summarisation, or
- Detecting plagiarism by comparing different student submissions to the same problem, or
- Automated assessment of novices’ programs, or
- Analysing student programming habits, such as correct use of object-oriented design principles.

Reusing this dataset will enable more research without individual researchers needing to gather their own data before investigating these questions.

4 DATA AVAILABILITY

The dataset can be found on the Open Science Foundation [1].

ACKNOWLEDGMENTS

We want to thank the King’s College Teaching Fund for funding part of the data capture for this dataset.

REFERENCES

- [1] Messer, Marcus and Brown, Neil and Kölling, Michael and Shi, Miaoqing. 2024. Menagerie: A Dataset of Graded Programming Assignments.
- [2] David Azcona and Alan Smeaton. 2020. +5 Million Python & Bash Programming Submissions for 5 Courses & Grades for Computer-Based Exams over 3 academic years.
- [3] David John Barnes and Michael Kölling. 2006. *Objects First with Java: A practical introduction using BlueJ*. Pearson/Prentice Hall, London, United Kingdom.
- [4] Neil C. C. Brown, Michael Kölling, Davin McCall, and Ian Utting. 2014. Blackbox: A Large Scale Repository of Novice Programmers’ Activity. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education* (Atlanta, Georgia, USA). Association for Computing Machinery, New York, NY, USA, 223–228.
- [5] Adrian de Freitas, Joel Coffman, Michelle de Freitas, Justin Wilson, and Troy Weingart. 2023. FalconCode: A Multiyear Dataset of Python Code Samples from an Introductory Computer Science Course. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*. Association for Computing Machinery, New York, NY, USA, 938–944.
- [6] Kathi Fisler. 2014. The Recurring Rainfall Problem. In *Proceedings of the Tenth Annual Conference on International Computing Education Research* (Glasgow, Scotland, United Kingdom). Association for Computing Machinery, New York, NY, USA, 35–42.
- [7] Natalie Kiesler and Daniel Schiffrer. 2023. Why we need open data in computer science education research. In *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1*. 348–353.
- [8] Kenneth R. Koedinger, Sidney D’Mello, Elizabeth A. McLaughlin, Zachary A. Pardos, and Carolyn P. Rosé. 2015. Data mining and education. *WIREs Cognitive Science* 6, 4 (2015), 333–353. arXiv:https://wires.onlinelibrary.wiley.com/
- [9] Marcus Messer, Neil C. C. Brown, Michael Kölling, and Miaoqing Shi. 2024. Automated Grading and Feedback Tools for Programming Education: A Systematic Review. *ACM Trans. Comput. Educ.* 24, 1, Article 10 (feb 2024), 43 pages.
- [10] Diba Mirza, Phillip T. Conrad, Christian Lloyd, Ziad Matni, and Arthur Gatin. 2019. Undergraduate Teaching Assistants in Computer Science: A Systematic Literature Review. In *Proceedings of the 2019 ACM Conference on International Computing Education Research* (Toronto ON, Canada) (ICER ’19). Association for Computing Machinery, New York, NY, USA, 31–40.
- [11] Emma Riese, Madeleine Lorås, Martin Ukrop, and Tomáš Effenberger. 2021. Challenges Faced by Teaching Assistants in Computer Science Education Across Europe. In *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1* (Virtual Event, Germany) (ITICSE ’21). Association for Computing Machinery, New York, NY, USA, 547–553.
- [12] E. Soloway. 1986. Learning to Program = Learning to Construct Mechanisms and Explanations. *Commun. ACM* 29, 9 (1986), 850–858.

⁴Ethics review references: LRS/DP-22/23-34272, LRS/DP-22/23-35578