Suggesting a Log-Based Creativity Measurement for Online Programming Learning Environment

Lilach Gal
Tel Aviv University
Tel Aviv Israel
lilachgal90@gmail.com

Arnon Hershkovitz
Tel Aviv University
Tel Aviv Israel
arnonhe@tauex.tau.ac.il

Andoni Eguíluz Morán
University of Deusto
Bilbao, Spain
andoni.eguiluz@deusto.es

Mariluz Guenaga
University of Deusto
Bilbao, Spain
mlguenaga@deusto.es

Pablo Garaizar
University of Deusto
Bilbao, Spain
garaizar@deusto.es

ABSTRACT
Creativity has long been suggested as an important factor in learning. In this paper, we present a preliminary study of creativity in an online programming learning environment. We operationalize creativity using an existing scheme for scoring it, and then measure it automatically based on the system log files. We analyze the data in order to explore the associations between creativity and personal/contextual variables. Creativity is associated with contextual variables and is not associated with personal variables. Directions for continuing this research are discussed.

Author Keywords
Creativity; programming learning; log-based measurement; learning analytics.

ACM Classification Keywords
H.1.2. User/Machine Systems: Software psychology; K.3.2. Computer and Information Science Education: Computer science education.

INTRODUCTION
Creativity refers to the creation of something (a product, a solution, etc.) that has some value. The importance of creativity in learning and for learners has been suggested since the mid-20th century. Still, there is no single definition of this construct; actually, more than a hundred definitions of creativity have been suggested [14], a direct result of its multifaceted nature.

L@S 2017, April 20–21, 2017, Cambridge, MA, USA.
© 2017 ACM ISBN 978-1-4503-4450-0/17/04...$15.00.
DOI: http://dx.doi.org/10.1145/3051457.3054003

Overall, it is agreed that creativity has four dimensions: fluency, flexibility, originality (also referred to as novelty), and elaboration [13]. The latter is relevant to non-verbal settings, hence will be neglected here. To put it simply, these dimensions measure the number (fluency), nature (flexibility), and uniqueness (originality) of the learner's products. As was previously shown, creativity in computer science (CS) education might improve CS knowledge and skills [12], and teaching CS might improve learners' creativity at large [3,5,10,11].

Contrary to previous attempts to automatically measure creativity of computer programs [6,8], our operationalization is directly based on the common multi-dimensional definition of creativity and on analyzing the whole process of using a programming learning system.

RESEARCH GOAL
Our main goal in the current study is to explore characteristics of creativity among children (6-18y/o) in an online environment for computational thinking learning. The research questions are:

1. What are the associations between creativity and personal characteristics (gender, age, programming knowledge, tech affinity)?
2. What are the associations between creativity and task difficulty?
3. Is creativity a state or a trait?

METHODOLOGY

The Learning Environment: Kodetu
Kodetu is a Web app, built using Google's Blockly, for teaching basic programming skills. Each of Kodetu's 15 levels presents the user with a maze in which an astronaut should get to a marked destination; guiding the astronaut is done via a block-based code the user is editing. The game levels introduce simple forward/turn commands, loops and conditions. While using the app, the system logs any action taken by its users. An example is brought in Figure 1.

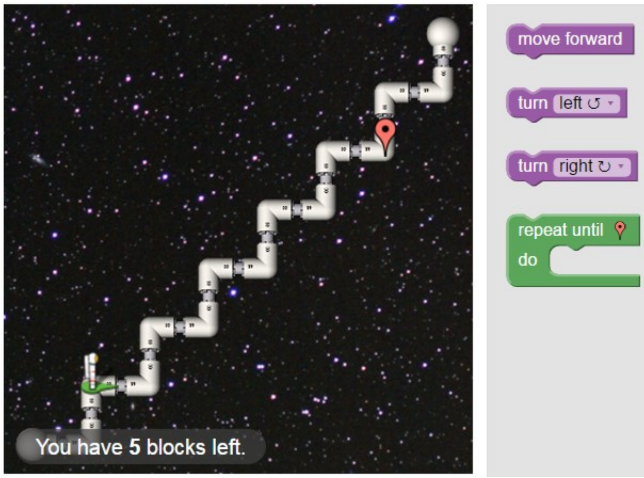

Figure 1. Level 9 of Kodetu. The user should edit a code, using the available blocks, to get the astronaut to the red mark (the coding-area is not shown here)

Population and Dataset

The dataset we analyzed is derived mostly from a set of technology workshops given to primary and secondary school students from the north-central part of Spain.

The original file included 307,608 rows, documenting 1,044 sessions taken between October 2014 and February 2015. The system logs the data anonymously, hence a session holds all the actions that occurred from start using the system until leaving it; therefore, a single user may be logged with multiple sessions. Logs document all users' suggested solutions and include timestamp, level [1-15], action result [Success, Failure, Timeout, Error, Unset], and the code. Triangulating with another file that held user's self-reported information, we also had for each session some personal characteristics of the user: age, gender, interface language [English, Spanish, Basque]; school-level [Primary School, Secondary School, High School, University, Professional Training, Not a Student]; programming background [Yes, No]; and technology affinity [1-10].

First, we removed rows that did not represent running a code (mainly, dragging a block to the editing windows – the resulting code was later documented when the user hit the "Run Program" button), as well as erroneously-logged rows. The final dataset includes 34,894 actions in 1,024 session. Overall, 45% of the sessions were reported to be taken by females and 55% by males, with an average reported age of 12.9 (SD=8.3), with about three-quarters of the sessions were taken by users at the ages 10-11y/o (756, 73.8%). About two-thirds (64.6%) of the sessions were taken by users who had no background in programming, and the average score for technology affinity was 8.7 (SD=1.6).

Operationalization of Creativity

We adapted Leikin's scoring scheme for creativity [7], developed in the context of multiple solution tasks in mathematics. As Kodetu lets the learners move to the next stage once a correct solution was given—just like many other CS learning environments, e.g., Code Monkey[1], Code.org, as well as other tutors in general, e.g., the mathematics tutor, ASSISTments[2]—we score for creativity in *solution attempts* rather than in *solutions*.

Scoring Fluency. Originally in [7], each student's solution got 1 point for each correct solution. We score with 1 point every solution attempt.

Scoring Flexibility. We establish groups of solutions based on the solution strategies. Our strategies are based on the use of control blocks (*While*, *If-Then*, *If-Then-Else*), presented from Level 7 and on. For levels 8-15, we defined 4 strategies:

- The solution attempt uses *While* and not using *If*;
- The solution attempt uses *If* and not using *While*;
- The solution attempt uses both *While* and *If*;
- The solution attempt uses neither *While* nor *If*.

Each solution attempt is scored with 10 points for a first time using a given strategy, and 0.1 points for any other solution attempt in a strategy previously used. (For levels 1-6, all solutions are referred to as belonging to a single strategy).

Scoring Originality

Instead of using a pre-defined rarity thresholds (15% and 40% in [7]), we decided to re-define these thresholds for each level separately, based on the distribution of unique solutions in it (by manually examining the solutions' distribution histogram and the knee of that curve).

Total Creativity Score for each Solution Attempt

Total creativity score for each solution was calculated by multiplying its flexibility and its originality scores.

Total Creativity Score for a Session

Total creativity score for a session is calculated as the average creativity score across all the solution attempts in it.

RESULTS

Taking the average of originality, flexibility, and creativity within each session (N=1,024), we get an average *Session Creativity* of 7.9 (SD=9.84), an average *Session Originality* of 9.8 (SD=13.5), and an average *Session Flexibility* of 12.8 (SD=4.2). Overall, the average of solutions attempts across all sessions was 3.9 (SD=3.4, skewness=4.7). All these variables are not normally distributed, hence we will use nonparametric statistical tests.

[1] http://playcodemonkey.com.
[2] https://www.assistments.org.

Next, we calculate averages of creativity, originality and flexibility for each level, across all sessions (see Table 1). Minimum and maximum values of originality in levels 8 and 13, respectively, reflect the number of unique solutions, which was minimal for level 8 and maximal for level 13.

Level	N	Mean Solution Attempts (SD)	Mean Originality (SD)	Mean Flexibility (SD)	Mean Creativity (SD)
1	857	1.4 (0.9)	1.1 (3.4)	10.0 (0.1)	5.9 (19.2)
2	833	2.7 (1.8)	6.5 (12.0)	10.2 (0.2)	9.0 (15.8)
3	787	2.5 (1.7)	2.8 (7.0)	10.1 (0.2)	4.7 (10.1)
4	756	2.8 (1.8)	3.8 (7.6)	10.2 (0.2)	7.2 (15.8)
5	721	2.3 (2.0)	3.8 (10.8)	10.1 (0.2)	4.6 (10.3)
6	704	3.5 (2.8)	8.9 (17.0)	10.3 (0.3)	10.9 (16.2)
7	665	4.1 (3.0)	9.6 (15.1)	10.3 (0.3)	10.3 (14.4)
8	639	2.1 (2.0)	0.8 (1.7)	12.3 (4.3)	1.6 (2.6)
9	638	3.9 (3.6)	9.6 (21.1)	12.5 (4.3)	5.4 (9.8)
10	619	3.4 (3.4)	10.1 (20.3)	12.3 (4.2)	8.2 (14.7)
11	589	6.5 (5.5)	22.0 (30.4)	22.8 (10.7)	10.8 (15.2)
12	530	5.0 (5.4)	17.1 (32.1)	16.5 (9.0)	9.9 (16.9)
13	456	8.3 (7.9)	48.1 (60.4)	18.9 (9.7)	14.8 (23.5)
14	344	6.1 (7.1)	5.1 (6.5)	17.2 (9.9)	3.9 (3.4)
15	283	8.0 (7.7)	6.9 (7.1)	16.2 (8.8)	3.0 (2.6)

Table 1. Descriptive statistics of creativity for each level

Creativity and User Characteristics

Two of the four user characteristics variables are binary: **gender**, and **background in programming**. We compare creativity between the two groups defined by each of these variables using Mann-Whitney U-test. No difference was found by **gender** in creativity, with Z=1.4 (p=0.17), respectively; same was found regarding fluency and flexibility. A marginally significant difference in creativity was found between those users with no **background in programming** and those with a programming background, with Z=1.89, at p=0.06, however with a negligible effect size[3] of r=0.06. Fluency and flexibility were shown with significantly different, again with very small effect sizes.

Technology affinity was found to be correlated with flexibility only, however with a low coefficient, ρ=0.1, at p<0.001; the other five comparisons of creativity dimensions and technology affinity/gender were not significant.

Creativity and Level

Correlating creativity measures with level, we get no significant results, with Spearman's ρ=-0.01, at p=0.96. It might be that Level 8, 14, and 15 are outliers in that context: Level 8 forces a two-block solution, which reduces significantly the possible solution space, hence affecting the *originality* score; Level 14 is rather an easy level; and Level 15 sets a very challenging task (solving a general maze), which is difficult even for experienced programmers. Omitting these two levels, we get a marginally significant correlation of ρ=0.6, at p=0.055, that is, the more difficult the task is, the higher the creativity is.

In order to better understand if there are differences in creativity based among the game levels, we ran Kruskal-Wallis one-way analysis of variance, comparing between the 15 groups defined by the game levels. This was run on a dataset of all pairs of session-level, N=9,421. The result is statistically significant, with $\chi^2(14)=882.6$, at p<0.001. An eta-squared index (a measure of effect size for ANOVA) can be derived from this value[4], resulting in $\eta^2=0.1$, a medium effect size.

We ran a post-hoc test, comparing creativity values between each pair of levels (for users who played in both levels). As we run an overall of 105 tests, we correct for multiple comparisons using the post-hoc False Discovery Rate (FDR) method, which produces a q-value that can be interpreted the same way as a p-value. The results suggest that in most cases (80 of 105 pairs), creativity average was statistically significantly different between levels. Effect sizes[5] range from very small values (0.05) to medium values (0.5).

More on the State-or-Trait Question

For shedding more light on the state-or-trait question, that is, is creativity more associated with user characteristics (trait) or with contextual variables (state), we set up two linear regression models to predict creativity on the full dataset, which includes 9,421 user-level pairs.

The first model tries to predict creativity by level. It uses 15 variables that denote the game levels. We will refer to this model as the *State Model*. Similarly, we set up a *Trait Model* to predict creativity by user; this model uses 1024 variables that denote the users. This approach is similar to the one applied in [2].

The linear regression models were built with M5' feature selection[6] and their goodness of fit was measured using Pearson's correlation and Root Mean Squared Error (RMSE). To validate the generalizability of the models, we calculated their fitness using 2-fold cross-validation. The models were built using RapidMiner Studio Version 7.2.003.

[3] Computed as $r = \frac{Z}{\sqrt{N}}$

[4] Computed as $\eta^2 = \frac{\chi^2}{N-1}$

[5] Here, $r = \frac{Z}{\sqrt{2N}}$, as we run within-subject comparisons, hence number of observations is doubled.

[6] In each iteration, the attribute with the smallest standardized coefficient is removed and another regression is performed; if the result is improved (based on Akaike Information Criterion, AIC), this attribute is dropped. This process is repeated until no attributes can be removed.

The *State Model* resulted with a correlation of 0.21 and RMSE=14.4. The *Trait Model* resulted with a correlation of 0.06 and RMSE=16.1. Therefore, in accordance with the above results, state explanations are better predictors of user creativity than trait explanations.

DISCUSSION

Our operationalization of creativity as a multi-dimensional contsruct, based on an established scheme (though from a different domain [7]), is an important step forward to analyze creativity at scale, as it allows automatization of evaluating creativity measures. Such an automatization will benefit both CS students and instructors by enabling the development of creativity assessment mechanisms.

Originally considered as a fixed personality trait, it is now suggested that a few forces influence creativity, some of which are personality traits (e.g., openness to experience, or persistent work style), and some of which are contextual (e.g., intrinsic task motivation) [1]. In the context of CS, an analysis of creativity in programming resulted with different creativity scores for different teaching methods of the same teacher to the same class, which demonstrate the influence of state-related variables on creativity [6]. Our findings support the "contextual approach"; specifically, creativity was higher as level progressed. In Kodetu, it is not only difficulty that is getting higher as levels progress, but only constraints are being presented to users in terms of code-length. As previously shown, constraints might be a fertile ground to creativity [9].

Also note, in the context of personal-related variables, that age, gender, programming background and technology affinity – were all unrelated to creativity in our data. Previous studies were also not able to find direct associations between creativity and gender, age, or cognitive abilities. Future work will be dedicated to shed more light on the associations between personal characteristics and creativity.

One barrier that we will need to tackle with further research lies in the fact that our operationalization of creativity, which originates in *multiple solution* tasks (in mathematics [7]) is being used in *multiple solution-attempts* tasks, which is popular in many tutoring systems. Transforming the idea of multiple solutions to multiple solution-attempts might hinder the validity of the construct, although previous studies associated multiple attempts to give a correct solution to creativity [4]. In any case, the validity question is not as simple as it may look, as it is not clear which validation might be tested. This is an issue that we plan to seriously tackle in the near future.

REFERENCES

[1] Amabile, T.M. & Pillemer, J. (2012). Perspectives on the social psychology of creativity. *Journal of Creative Behavior, 46*(1), 3-15.

[2] Baker, R. (2007). Is gaming the system state-or-trait? Educational data mining through the multi-contextual application of a validated behavioral model. *Presented in the Workshop on Data Mining for User Modeling at the 11th International Conference on User Modeling (July 25-29, Corfu, Greece)*.

[3] Bustillo, J. & Garaizar, P. (2016). Using Scratch to foster creativity behind bars: Two positive experiences in jail. *Thinking Skills and Creativity, 19*, 60-72.

[4] Karakok, G., Savic, M., Tang, G., & El Turkey, H. (2015). Mathematician's view on undersgraduate students' creativity. *Proceedings of the Ninth Congress of the European Society for Research in Mathematics Education (February 4-8, Prague, Czech Republic)*, 1003-1009.

[5] Kobsiripat, W. (2015). Effects of the media to promote the scratch programming capabilities creativity of elementary school students. *Procedia – Social and Behavioral Sciences, 174*, 227-232.

[6] Koh, K.H., Bennett, V., & Repenning, A. (2011). Computing indicators of creativity. *Presented at the ACM Creativity & Cognition Conference (November 3-6, Atlanta, GA)*.

[7] Leikin, R. (2009). Exploring mathematical creativity using multiple solution tasks. In R. Leikin, A. Berman, & B. Koichu (Eds.), *Creativity in Mathematics and the Education of Gifted Students (pp. 129–145)*. Rotterdam, the Netherlands: Sense Publisher.

[8] Manske, S., & Hoppe, H.U. (2014). Automated indicators to assess the creativity of solutions to programming exercises. In *Proceedings of the 2014 IEEE 14th International Conference on Advanced Learning Technologies*, 497-501.

[9] Moreau, C.P. & Dahl, D.W. (2005). Designing the solution: The impact of constraints on consumers' creativity. *Journal of Consumer Research, 32*(1), 13-22.

[10] Pardamean, B. (2014). Enhancement of creativity through Logo programming. *American Journal of Applied Science, 11*(4), 528-533.

[11] Scherer, R. (2016). Learning from the past – The need for empirical evidence on the transfer effects of computer programming skills. *Frontiers in Psychology,* doi: 10.3389/fpsyg.2016.01390.

[12] Shell, D. F., Hazley, M.P., Soh, L.K., Dee Miller, L., Chiriacescu, V., & Ingraham, E. (2015). Improving learning of computational thinking using computational creativity exercises in a college CS1 computer science course for engineers. *Presented at the 44th Annual Frontiers in Education (FIE) Conference (October 22-25, Madrid, Spain)*.

[13] Torrance, E.P. (1974). Torrance tests of creative thinking. Bensenville, IL: Scholastic Testing Service.

[14] Treffinger, D.J., Renzulli, J.S., & Feldhusen, J.F. (1971). Problems in the assessment of creative thinking. *The Journal of Creative Thinking, 5*, 104-112.