Be Constructive: Learning Computational Thinking Using Scratch™ Online Community

Bushra Chowdhury[1], Aditya Johri[2(✉)], Dennis Kafura[1], and Vinod Lohani[1]

[1] Virginia Tech, Blacksburg, VA 24061, USA
bushrac@gmail.com, kafura@cs.vt.edu, vlohani@vt.edu
[2] George Mason University, Fairfax, VA 22030, USA
johri@gmu.edu

Abstract. Online learning communities are predicated on the assumption that social interaction among participants will lead to learning. Yet, research has shown that not all interactions result in learning and that there is a need to develop a more nuanced understanding of the nature of activities in online communities and their relationship with learning. We analyzed data from the Scratch™ online learning community, a platform designed to teach Computational Thinking (CT) through block-based activities, using the Differentiated Overt Learning Activities (DOLA) framework to assess learning. We found that users who engaged in constructive activities demonstrated higher learning, as illustrated by the complexity of their contributions, compared to users who were merely active on the platform. We compared users across two sub-communities within Scratch and found that participation and contributions across the two domains resulted in different learning outcomes, showcasing the effect of context on learning within online communities.

Keywords: Online community · Computational thinking · Informal learning · Collaborative learning · Scratch

1 Introduction

The use of online communities to foster learning is well-documented in the literature especially for computational concepts and programming [28, 31]. The context for this research is Scratch™, specifically its community-based platform where one can code games, animations, and stories using media-based programming language. In Scratch, one uses ‘blocks’, which are puzzle shaped pieces, to create code. The blocks connect to each other like a jigsaw puzzle, where each block represents a particular programming concepts (e.g., if, do-if, repeat, end). Within Scratch, users have the opportunity to see projects completed by others, use pre-existing code, comment on others’ projects and seek assistance from others. Community based interaction in Scratch manifests as commenting, remixing, and sharing of projects [11]. Scratch has been used in AP courses and introductory programming courses at the college level [9, 10]. Scratch has been designed primarily to support the development of Computational thinking (CT) in primarily in young people (ages 8 to 16) [4–7] but has also found some adoption

M. A. Herzog et al. (Eds.): ICWL 2019, LNCS 11841, pp. 49–60, 2019.
https://doi.org/10.1007/978-3-030-35758-0_5

among adults [8]. CT as a concept articulates a set of problem-solving thought processes derived from computer science but applicable in any domain [29]. In its earlier incarnations CT was largely seen as "algorithmic thinking" and referred to using an ordered and precise sequence of steps to solve problems. Wing (2006) defined it as "solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science" [30]. Overall, CT is a broad umbrella term that encompasses core computational science concepts as well as programming skills.

2 Prior Work and Research Objective

Several studies of the Scratch community have examined the relational aspect of user's social behavior. Researchers have used comments, number of projects created, remixing, favorites, love-its and friend requests as a way to characterize collaboration [11–13]. Sylvan [14] investigated social interactions in terms of project influence and social influence. Project influence was measured as the number of times a project has been downloaded and social influence in terms of betweenness centrality of friendship networks. Studies have also looked at number of times a user's project had been featured and gallery participation [14]. Similarly based on social factors and differences in projects created, Scaffidi and Chambers [21] categorized users as project leader, active user, peripheral user and remixer/passive user and Monroy-Hernández and Resnick [22] categorized users in terms as active consumer, passive producer active producer. These studies have focused on the overall community or the user themself but have not examined CT. On the other hand, researchers that have examined the use of computational concepts within Scratch [15–20] have not focused on the interaction among users. The exception is Dasgupta et al. who in their study focused on the association between remixing and learning CT [5]. They studied number of remixes, downloads made by the user, experience of the user, comments received on projects created by the user, and number of different blocks used by the user as predictors.

Overall, although studies have looked at social and collaborative aspects of Scratch and also at CT, there is limited understanding of how specific kinds of activities that users engage in can lead to CT learning. This study hypothesizes that difference in the nature of social interactions exhibited by a user can lead to difference in learning of CT. To examine directly the relationship between the nature of interactions and learning outcomes, this study leverages a framework advances within the cognitive sciences and uses data from the online community for Scratch™.

3 Analytical Framework

3.1 Differentiated Overt Learning Activities (DOLA) Framework

Few frameworks allow for differentiation of learner activities so that we might be able to tease out what activities actually lead to learning. Chi proposed a framework – Differential Overt Learning Activities or DOLA – which categorizes learner activities as being active, constructive or interactive [3]. An interactive activity involves higher cognitive process than constructive and constructive is higher cognitively than active learning. In contrast to any form of active learning, passive learning involves teacher-centered methods and is a form of learning applicable largely to formal educational contexts, and therefore, not applicable to our study [23].

An active activity can be noticed when a person takes an action, does something physically or verbally. A constructive activity is demonstrated when the output of the interaction goes beyond the information initially provided. An interactive activity is exhibited between partners when both parties are involved in collaboration contribute equally. Table 1 below provides brief definitions of active, constructive and interactive activities and the supposed cognitive processes associated with them.

Table 1. Chi's Differentiated Overt Learning Activities (DOLA) framework

	Active	Constructive	Interactive
Feature	Doing something physically	Producing outputs that contain ideas that go beyond the presented information	Dialoging substantively on the same topic, and not ignoring partner's contributions
Cognitive processes	*Attending Processes* Activate existing knowledge Assimilate, encode, or store new information Search existing knowledge	*Creating Processes* Infer new knowledge Integrate new information with existing knowledge Organize own knowledge for coherence Repair own faulty knowledge Restructure own knowledge	*Jointly Creating Processes* Creating processes that incorporate partner's contributions

Classroom studies [23, 24] using Chi's DOLA framework have characterized selecting, repeating, paraphrasing as active behavior. In terms of virtual learning environments, simply manipulating an existing scenario in simulation software was considered to be active. On the other hand, when a learner elaborately explains a problem, makes a connection to previous problems, generates a hypothesis, compare and contrasts, draws analogies the learner is considered to be constructive. An interactive activity in collaboration is revealed when both partner's debate each other's ideas, when an instructor provides feedback which leads to a more extended dialogue discussing the issue etc. Although Chi's framework looks at all activities, alone or in collaboration, it can be easily extended to study collaborative learning and studies using Chi's DOLA framework have been used to evaluated student's collaborative behavior in learning concepts in physics, mathematics, bridge design, evolutionary biology, human circulatory system, introductory materials science and engineering.

3.2 Operationalizing DOLA for Scratch

For the purposes of this study, we operationalize Scratch users' activities in terms of being active and constructive and investigate their relationship to CT learning. Within the context of Scratch online community most activities are collaborative in the sense that they involve either interacting with or using elements of others' project. In particular, we were interested in understanding these socially driven activities since they are a better indicator of collaboration.

We did not examine interactive activity since the inherent affordances of the platform restricted interaction to asynchronous and even if users interacted synchronously, we did not have the data for the analysis (there was no clickstream data, for instance). We also did not operationalize passive activity since the online platform data did not contain data of passive activity such as number of times a user played or watched a project.

This study defines collaborative activities of a user by counting the number of times the user initiates different types of interactions. For example: the total number of users a particular user follows, a number of times user favorites other projects, the number of times a user goes out and makes comment on other's projects, and the number of times a user remixes another user's project. In case of adapting Chi's DOLA framework the study further classifies these interactions into active and constructive. Table 2 provides definitions of active interactions. Interactions that do not modify or elaborate on the topic were considered active (e.g. follow, favorite).

Table 2. DOLA operationalized for scratch – active category

Active activities	Definition
Favorited	Total number of times a user favorites other projects
Follow	Total number of users a particular user follows
Active_Comment	Total number of times a user makes active comments on other projects word count less than 18 usually emojis, encouraging phrases/verbs/adjectives [1]
Active_Remix	Total number of projects created by a user that was a remix of another user's project where the number of different types of blocks uses is the same as the original project

To categorize comments as active and constructive, Velasquez et al. [11]'s findings have been applied. According to Velasquez et al. [11], comments with word count less than 18 (usually emojis, encouraging phrases, verbs and adjectives) were categorized as active and comments with word count more 18 as constructive. However, in order to better understand the constructive nature of the comments, comments with greater than 18 words were manually coded for our study. An interaction was characterized as constructive interaction when a user is assumed to modify and elaborate on a particular topic (e.g. modify a projects code). Table 3 provides definitions of constructive interactions.

Table 3. DOLA operationalized for scratch – constructive category

Constructive activities	Definition
Original_Projects	Total number of original projects created by a user
Constructive_Comment	Total number of times a user makes constructive comments on other projects Word count >18.320 and containing constructive praise or criticism [1]
Constructive_Remix	Total number of projects created by a user that was a remix of another user's project where the number of different types of blocks uses is the more than the original project

4 Data Description and Selection

The scratch community datasets from 2007 through 2012 are publicly available to researchers for analysis through MIT Media Lab. Scratch Research Data (available at: https://llk.media.mit.edu/scratch-dataset/). This study used the following data tables for analysis: comments, downloaders, favorites, friends, projects, project_blocks, and users.

To find a relevant dataset within the larger data corpus, we used pre-classified data from Gelman et al. [2]. Gelman et al. [2] in their study identified clusters of Scratch 8184 users who had more than 25 followers. Gelman et al. used OpenOrd layout in Gephi to identify different models of community growth over time to understand how scratch user's behavior and dynamics impact community participation. OpenOrd is a multi-level, force-directed layout and uses average-link clustering based on both edge weights and distance, where distance is determined using a force-directed algorithm [25]. Clusters of nodes were replaced by single nodes, and the clustering was repeated until a certain distance threshold between the nodes was reached. After the clustering was complete, the graph was expanded by replacing the individual nodes with the original graphs in each cluster. Using a text mining approach and by concatenating project titles, descriptions, and tags for all projects within each cluster, each cluster was represented as a document with a bag-of-words approach. For each term, the term frequency inverse document frequency (TFIDF) value was calculated resulting in five clusters: a cluster that heavily featured old Scratch users (number of users 278), a young cluster of game makers (number of users 1798), an comparatively mature game making user cluster (number of users 2710), a cluster of users focusing on art projects and another cluster which had a Variety of projects (number of users 2260). The reason for choosing TFIDF over Term Frequency (TF) was to determine the relevance of a particular term within a particular cluster versus its relevance across all clusters.

We sub-selected two of the pre-identified clusters by Gelman et al. [2], the cluster with comparatively mature game makers and the cluster with variety of projects. The Gaming cluster initially had 2710 users. However, only 2173 users only had at least 5 projects. Similarly, in the Variety cluster initially 1934 out of 2260 users were used for analysis. From now onwards in this paper we will identify these two clusters as game cluster and Variety cluster. The reason for selecting two completely diverse clusters was to illustrate difference in cluster behavior. The Gaming cluster would exemplify comparatively keen CT learners and the Variety cluster would provide more of a general user behavior in Scratch.

5 Data Analysis

5.1 Learner Initiated Computational Thinking

In this study, we aim to focus only on actions initiated by the user. It is important to investigate such self-initiated social behaviors because these interactions are self-motivated and self-regulated by the learner himself/herself. Thus, in order to assess a learner's overall experience in learning in an open online line platform it is necessary to focus on what s/he does as well as what s/he learns from the community.

Figure 1 illustrates different social interactions initiated by a user in the context of the Scratch community. For example, User A in Fig. 1 can leave a comment on a project, favorite projects created by other users, and follow other users in the Scratch community. User A can also create a project from scratch (original project) or remixing code from pre-existing projects (remixed project). All these interactions: comment, favorite, follow and creating/remixing are user initiated.

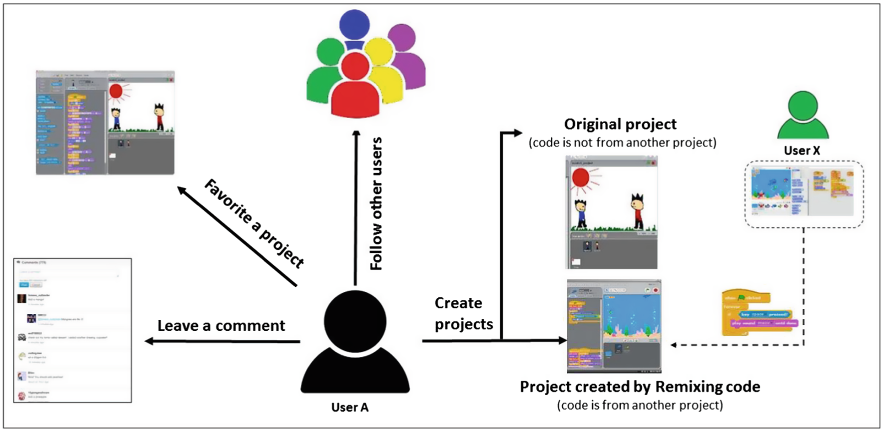

Fig. 1. Social interaction initiated by a user in scratch

5.2 Learning Analysis Using IDF

In this study, CT as a process has been evaluated as a combination of user's self-initiated social interactions and by evaluating the amount of CT learning skills the user is exhibiting with- in projects. One of the studies to directly examine learning within

Scratch is Yang et al. [15]. They used Inverse Document Frequency (IDF) to assess learning. IDF is a widely used statistical measure for assessing the importance of a word is in a document [26, 27]. Whereas, Term Frequency (TF) is simply the number of times a word appears in a document, IDF assigns more weight to key words that appear rarely than the ones that are more commonly used. Thus, IDF helps to better determine the breadth of use of different CT operators used with-in projects created by Scratch users. Yang et al.'s model assigned higher weight to computational blocks that were rarely used and lower weight to frequently used blocks. Based on the different types blocks used in original projects created by a user, Yang's model calculates a cumulative value of learning. We apply the same method to assess learning for all users in a cluster with at least 5 projects. The cumulative value was calculated based on all original projects created by a user. We further subcategorize learning across three parameters: loop learning, conditional learning and operator learning. Table 4 indicates the different blocks used to operationalize each sub learning categories. Studies evaluating CT in Scratch have similarly used blocks to assess the different categories CT learning [4, 5, 7].

Table 4. Categories of CT and corresponding blocks used for evaluation

Type of learning	Blocks used
Loop	forever, foreverIf, repeat, repeatUntil
Conditional	waitUntil, foreverIf, if, ifElse, repeatUntil, bounceOffEdge, turnAwayFromEdge, touching, touchingColor, colorSees, mousePressed, keyPressed, isLoud, sensor, sensorPressed, lessThan, equalTo, greaterThan, and, or, not, listContains
Operator	lessThan, equalTo, greaterThan, and, or, not, add, subtract, multiply, divide, pickRandomFromTo, concatenateWith, letterOf, stringLength, mod, round, abs, sqrt

6 Results and Discussion

In order to test the hypothesis proposed in this study we conduct three steps of analysis. For both clusters we first measure interactions of each user; second, we measure CT learning of each user; and third, we correlate interactions with CT learned. These three steps allow us to examine the one-to-one relationship between different types of social interactions and CT learning.

The summary statistics for measures of social interactions and learning of users of the Gaming and the Variety cluster are provided below in Table 5 (G: Gaming; V: Variety).

Table 5. Summary statistics for gaming and variety clusters

		M Median	$\bar{x}$ Mean	Σ S.D	Range
Favorited	G	36	105	238.66	[0–4232]
	V	28	79.98	182.057	[0–3466]
Follow	G	43	90.92	152.9	[1–2061]
	V	44	95.94	177.350	[0–3070]
Active_Comment	G	276.95	414.95	491.908	[0–7400]
	V	206	352.39	456.456	[0–5882]
Active_Remix	G	8	16.67	27.27	[0–501]
	V	8	17.25	28.751	[0–429]
Original _Projects	G	44	71.89	93.04	[5–1841]
	V	38	70.22	117.350	[5–1903]
Constr_Comment		Very limited, disregarded this factor			
Constructive_Remix	G	7	14.31	25.93	[0–617]
	V	4	10.19	22.86	[0–565]
Total CT learning	G	101	99.15	36.65	[.8–318.54]
	V	72.07	73.60	43.03	[0–312.37]
Loop learning	G	4.19	3.66	.96	[0–4.19]
	V	3.13	2.97	1.39	[0–4.19]
Conditional learning	G	18.9	18.49	6.19	[0–31.48]
	V	15.25	14.34	7.47	[0–28.77]
Operator learning	G	12.47	12.43	6.11	[0–41.39]
	V	8.55	8.86	6.70	[0–41.39]

In terms of differences between clusters, values for favorites (Gaming cluster user's median favorited: 36, Variety cluster user's median favorited: 28) were higher in Gaming cluster than the Variety cluster. All other social interactions values were less in the same range for both the Gaming and Variety cluster.

In terms of learning, the Gaming cluster had higher values (user's median use of loop in a project: 4.19, conditional: 18.9 operator: 12.47) than the Variety cluster (loop: 3.13, conditional: 15.25, operator: 8.55). Before correlating social interactions with learning some factors (e.g. remix_active and constructive, follow, favorited) were log transformed to achieve normality. This was done because those particular data sets were highly skewed. Previous studies [5, 17] on Scratch data have also used log-transformed data sets for analysis. According to our model, the more challenging or rare and constructive is the nature of the task, the higher is the learning.

The findings of the study confirm the proposed hypothesis that in the context of Scratch users, difference in the type (active versus constructive) of activity exhibited by a user can lead to difference in learning of CT. For example, a user who had created more original projects and more remix projects with added features to the existing code learned more than a user who did less of either of these two constructive activities. Creating remixed projects and commenting on projects were also found to be active behavior that correlates to learning of CT.

In terms of separate CT concepts, including Loop, Conditions and Operators; Conditional learning correlated higher with constructive interactions than the Loop or Operator learning. For both clusters, learning from constructive interactions was found to be stronger than the relationship between CT learning and active interactions. In terms of difference between clusters and relationship between social interactions and learning CT, users of the Variety cluster learned more by creating original projects and extensively remixing (remix_constructive) projects than users of the Gaming clusters. However, when it came to active interactions, users of the Gaming cluster learned more by just following, commenting and simply remixing (remix_active) code than users of the Variety cluster (Table 6).

Table 6. Correlational analysis of activity and CT learning

		Total CT learn	Loop learn	Condition learn	Operator learn
Active Interactions					
log_follow	G	.077**	.067**	.072**	.070**
	V	.024	−.026	.002	.019
log_comment_active	G	.293**	.252**	.259**	.221**
	V	.351**	.278**	.297**	.282**
log_favorited	G	.228**	.199**	.228**	.191**
	V	.209**	.176**	.182**	.178**
log_remix_active	G	.378**	.268**	.347**	.363**
		.466**	.390**	.438**	.419**
Total_log_active	G	.321**	.261**	.293**	.261**
	V	.466**	.390**	.438**	.419**
Constructive Interactions					
log_original	G	.466**	.312**	.404**	.430**
	V	.545**	.450**	.507**	.483**
log_remix_constructive	G	.509**	.370**	.478**	.456**
	V	.636**	.533**	.603**	.588**
Total_log_constructive	G	.497**	.337**	.459**	.433**
	V	.636**	.533**	.602**	.588**

A stronger association of active interactions with the Gaming clusters and that of constructive interactions with the Variety clusters could be explained by the difference in the learning scores of the users in both clusters. Since users of the Gaming cluster scored higher compared to users of the Variety cluster, possibly users of the Gaming cluster learned more by casually investigating (favoriting/active_remix) projects, whereas the users of the Variety cluster needed to explore the project in depth (constructive_remix) to understand and use a computational concept. Users with higher prior knowledge found it easier to learn compared to a novice CT learner and a novice CT learner needs to do more constructive tasks to learn more.

7 Discussion and Conclusion

In this study, we evaluated social interactions initiated by users in the Scratch community using Chi's DOLA framework to characterize online behavior that suggests learning. The analysis revealed a relationship between active, constructive social interactions and learning of CT. Different clusters exhibited difference in learning based on the type of social interactions. Users of the Gaming cluster were able to learn at a higher rate while being socially active (active social interactions) whereas for users of the Variety cluster to be learning at higher rate they needed to be socially constructively. To put in simply, a seasoned CT learner (e.g. Gaming Cluster user) can gain knowledge by glancing at another project's code, whereas for a novice CT (e.g. Variety cluster users) learner needs to be hands-on constructively engaged to learn CT.

Overall, the findings from this study support prior work that shows a clear connection between the ability of online communities to support different forms collaborative activities and the affordances that provides for learning [28, 32–34]. This work also showcases the potential upside of using data mining and machine learning to analyze learning [35, 36]. In terms of practical application of this work, educators should design problems that foster active/constructive behavior in novice CT learners. Novice learners can start off by solving active problems such as examination of pre-written codes by experts, or making minor changes to pre-existing codes. Gradually, the learners should be encouraged to add new features to pre-written code (the constructive idea of remixing with added features) and keep on creating new computational projects.

8 Limitations

The primary limitation of this work is the lack of self-reported information (e.g., age, sex) of participants. We were unable to collect any data reported directly by the learner to assess learning. The dataset is relatively older but given the comprehensive nature of the data, it is still relevant for answering the research questions we have raised.

Acknowledgements. This work is supported in part by U.S. National Science Foundation Awards #142444, 1408674, & 1712129 (PI: Johri) and Award#1624320 (PI: Kafura). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies. We thank Ben Gelman for his support with the data collection and analysis.

References

1. Kreijns, K., Kirschner, P.A., Jochems, W.: Identifying the pitfalls for social interaction in computer-supported collaborative learning environments: a review of the research. Comput. Hum. Behav. **19**(3), 335–353 (2003)
2. Gelman, B., Beckley, C., Johri, A., Yang, S., Domeniconi, C.: Online urbanism: interest-based subcultures as drivers of informal learning in an online community. In: Proceedings of ACM Learning at Scale Conference 2016 (2016)

3. Chi, M.: Active-constructive-interactive: a conceptual framework for differentiating learning activities. Top. Cogn. Sci. **1**(1), 73–105 (2009)
4. Brennan, K., Resnick, M.: New frameworks for studying and assessing the development of computational thinking. Presented at annual meeting of the American Educational Research Association, Vancouver, Canada (2012)
5. Dasgupta, S., et al.: Remixing as a pathway to computational thinking. In: Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing (2016)
6. Robles, G., et al.: Software clones in scratch projects: on the presence of copy-and-paste in computational thinking learning. In: IEEE 11th International Workshop on Software Clones (IWSC) (2017)
7. Moreno-León, J., Robles, G., Román-González, M.: Scratch: automatic analysis of scratch projects to assess and foster computational thinking. RED Revista de Educación a Distancia **15**(46), 1–23 (2015). https://www.um.es/ead/red/46/moreno_robles.pdf
8. Resnick, M., et al.: Scratch: programming for all. Commun. ACM **52**(11), 60–67 (2009)
9. Ericson, B., Guzdial, M., Biggers, M.: Improving secondary CS education: progress and problems. In: ACM SIGCSE Bulletin. ACM (2007)
10. Bryant, R., et al.: Computational thinking: what is it, how is it relevant, who's doing what with it? J. Comput. Sci. Colleges **25**(1), 170–171 (2009)
11. Velasquez, N.F., et al.: Novice Programmers Talking about Projects: What Automated Text Analysis Reveals about Online Scratch Users' Comments (2014)
12. Scaffidi, C., Dahotre, A., Zhang, Y.: How well do online forums facilitate discussion and collaboration among novice animation programmers? In: Proceedings of the 43rd ACM technical symposium on Computer Science Education. ACM (2012)
13. Fields, D.A., Giang, M., Kafai, Y.: Understanding collaborative practices in the Scratch online community: Patterns of participation among youth designers. In: CSCL 2013 Conference Proceedings. International Society of the Learning Sciences (2013)
14. Sylvan, E.: Predicting influence in an online community of creators. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM (2010)
15. Yang, S., Domeniconi, C., Revelle, M., Sweeney, M., Gelman, B., Beckley, C., Johri, C.: Uncovering trajectories of informal learning in large online communities of creators. In: Proceedings of ACM Learning at Scale (2015)
16. Manovich, L., Remixing and remixability. Retrieved on Jan, 2005. 10: p. (2008)
17. Hill, B.M., Monroy-Hernández, A.: The remixing dilemma the trade-off between generativity and originality. Am. Behav. Sci. **57**(5), 643–663 (2013)
18. Hermans, F., Aivaloglou, E.: Do code smells hamper novice programming? Delft University of Technology, Software Engineering Research Group (2016)
19. Matias, J.N., Dasgupta, S., Hill, B.M.: Skill progression in scratch revisited. In: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems. ACM (2016)
20. Ota, G., Morimoto, Y., Kato, H.: Ninja code village for scratch: Function samples/function analyser and automatic assessment of computational thinking concepts. In: Proceedings of 2016 IEEE Visual Languages and Human-Centric Computing (VL/HCC)
21. Scaffidi, C., Chambers, C.: Skill progression demonstrated by users in the Scratch animation environment. Int. J. Hum. Comput. Interact. **28**(6), 383–398 (2012)
22. Monroy-Hernández, A., Resnick, M.: Empowering kids to create and share programmable media. Interactions **15**, 50–53 (2008). ACM ID, 1340974
23. Menekse, M., et al.: Differentiated overt learning activities for effective instruction in engineering classrooms. J. Eng. Educ. **102**(3), 346–374 (2013)
24. Chi, M., Wylie, R.: The ICAP framework: linking cognitive engagement to active learning outcomes. Educ. Psychol. **49**(4), 219–243 (2014)

25. Martin, S., et al.: OpenOrd: an open-source toolbox for large graph layout. In: IS&T/SPIE Electronic Imaging. International Society for Optics and Photonics (2011)
26. Sparck Jones, K.: A statistical interpretation of term specificity and its application in retrieval. J. Documentation **28**(1), 11–21 (1972)
27. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval, vol. 1. Cambridge University Press, Cambridge (2008)
28. Bruckman, A.: Learning in online communities. In: Sawyer, R.K. (ed.) The Cambridge Handbook of the Learning Sciences, Cambridge University Press, New York, pp. 461–472 (2006)
29. Wing, J.M.: Computational thinking. Commun. ACM **49**(3), 33–35 (2006)
30. Yadav, A., Stephenson, C., Hong, H.: Computational thinking for teacher education. Commun. ACM **60**(4), 55–62 (2017)
31. Brandt, J., Guo, P., Lewenstein, J., Dontcheva, M., Klemmer, S.: Two studies of opportunistic programming: interleaving web foraging, learning, and writing code. In: Proceedings of CHI, Boston, MA, USA, pp. 1589–1598 (2009)
32. Johri, A., Yang, S.: Scaffolded help for informal learning: how experts support newcomers' productive participation in an online community. In: Proceedings of Communities and Technologies (C&T) (2017)
33. Almatrafi, O., Johri, A.: Showing and telling: response dynamics in an online community of makers. In: Proceedings of CSCL 2017 (2017)
34. Teo, H., Johri, A., Lohani, V.: Analytics and patterns of knowledge creation: experts at work in an online engineering community. Comput. Educ. **112**, 18–36 (2017)
35. Almatrafi, O., Johri, A., Rangwala, H.: Needle in a haystack: identifying learner posts that require urgent response in MOOC discussion forums. Comput. Educ. **118**, 1–9 (2018)
36. Lester, J., Klein, C., Rangwala, H., Johri, A.: Learning Analytics in Higher Education. ASHE Monograph Series, vol. 3, Issue 5 (2017). https://twitter.com/_CHINOSAUR/status/461864317415989248