INTERACTIVE STORYTELLING FOR PROMOTING CREATIVE EXPRESSION IN MEDIA AND CODING IN YOUTH ONLINE COLLABORATIVES IN SCRATCH

Deborah A. Fields[1], Yasmin B. Kafai*, Anna Strommer[1], Elissa Wolf*, and Bailey Seiner*

Abstract
Constructionist approaches have favored children's learning designing games and software applications because of their social and personal relevance. Designing interactive stories and animations have been seen as equally motivational but less successful in promoting the sophisticated forms of computational thinking and participation that are a prominent part of newly emerging social networking forums for online creative communities. In this paper, we examine two promising aspects of online creative collaboratives for learning in Scratch community: (1) how learning programming can emerge alongside creative expression in collaboratively designed interactive stories, and (2) how feedback can shape the artistic and computational qualities of stories between draft and final project online submissions. The analyses focused on changes in drafts and final projects by examining quantity and quality of media elements and program code in a collaborative story telling activity in Scratch online community. Findings indicate that online collaborative creative storytelling and constructive feedback provided on drafts by community and Scratch team members have the potential to generate both more complex story designs and code development. In the discussion we address the implications for designing creative collaboration that situate programming and media designs in mutually supportive ways.

Keywords online creative collaboration, Scratch programming, digital storytelling

1. Introduction

Constructionism has promoted learning by design approaches for creative expression and knowledge reformulation [27]. These design approaches provide student projects with personal relevance while at the same time promoting technical skills such as programming [26]. Historically however, most research has focused either on the technical skills learned or on the personal and social relevance of the artifacts created by students within constructionist learning environments. More often than not, the technical side of computational learning has been prioritized by focusing on artifacts like designing games [8; 18], simulations [29], or even robots [35]. The creative side of programming has received much less attention, and if so, mostly in relation to creating stories [21], music videos [10], or interactive art [28]. These digital media productions that draw on traditional arts like visual expression, music, theater, and narrative writing have been seen as computationally less challenging. Indeed, in the Scratch online programming community, creating stories around

[1] Utah State University, deborah.fields@usu.edu, amstrommer@gmail.com
* University of Pennsylvania, kafai@upenn.edu, elissa.wolf92@gmail.com, brseiner@gmail.com

young adult fiction has emerged as a popular activity, especially among girls, but one that often involves less sophisticated programming than the equally popular game designs preferred by mostly boys [31]. How to support more sophisticated programming alongside rich storytelling that promotes creative expression is thus a promising area of constructionist approaches.

Constructionist approaches have also focused on social relevance, seeing the construction of digital artifacts as a social enterprise, one that engages learners in sharing their artifacts with others. Papert [27] in particular had been fond of learning communities such as samba schools where the collaborative development, critique and performance of activities is a driving force for the learning of all members. In recent years, many online youth networking communities have begun creating, sharing, and socializing around user-created content [13]. Such social online forums, where others are creating similar types of artifacts, afford many educational opportunities, including designing or writing for a specific audience [23], giving and receiving constructive criticism [1], creating projects collaboratively [2], [20], studying the design of others' projects, remixing or redesigning the designs of others [25], and making mods of games [12], [14]. Youth online programming communities such as Scratch are emerging as sites that may provide the kind of technical, creative, and social support needed for adolescents to develop skills and discipline their creative imaginations, providing a new venue for a "special culture" to support youth development [15]. Yet instrumenting participation and performance in such youth online communities is rare, especially in such ways that specifically support adolescents' programming activities as they take place alongside creative expression.

In this paper, we examine how interactive storytelling can become a context for creative expression that engages youth in more sophisticated design of interactive media and in more complex program code by situating it in the online social networking forums of Scratch. By adopting elements of the studio art model [34] for the design of online collaborative story telling, we used both creative constraints and constructive criticism to support the integration of more intermediate and advanced programming concepts in stories created with Scratch. In alignment with Csikszentmihalyi [5] we view creativity through a socially systemic lens, where individuals build on culturally valued practices, meanings, and designs to produce new variations within a domain. In this context viewing others' work may promote a view of the social side of creativity, learning what a community views as creative, original, good, and thought-provoking [6]. Thus we see creativity as an equally personal and social act that includes sustained discussion with peers and the need for an appreciation of the constraints that one is augmenting or violating while producing a contribution. Such a view of creativity lends itself well to examine creative expression as part of constructionist practices prominent in current-day social networking forums such as the Scratch community.

We developed an online collaborative and creative storytelling activity as an effort to invite Scratch members' participation into collaborations and stimulate their narrative expression and programming skills in a supportive, community context. We chose to target *interactive* stories as a genre to engage youth in intermediate level programming skills (primarily around interactivity) alongside more creative expressions (primarily around media use and narrative) in a meaningfully paired way. While programming productions, alone or in collaboration with others [7], can result in impressive products [22], we were also interested in the role community feedback can play in how programming artifacts can be improved. Such feedback, in the form of comments or critiques or even remixing, is a key dimension of many, even youth online communities [1] but has rarely been examined or even engineered as part of educational interventions around programming. We know from preliminary research that experienced peers and mentors can play an important leadership role in the Scratch community [33].

With this as background, our investigations focused on two key aspects of online creative collaborative programming: (1) how more sophisticated programming can emerge alongside creative expression in collaboratively designed interactive stories in both media and coding used by youth teams, and (2) how an online community can be supportive of youths' development of creative expression in interactive storytelling, in particular how it can shape the artistic and computational qualities of stories between draft and final project submission. During the month of August 2011, we started a Collab Camp in the online Scratch community, as part of a research program on online creative collaboration in youth programming [20], [31]. Our research questions focus on creative expression and programming techniques as well as the role of community support in improving story projects. In this we aim to consider programming as an artistic skill and online community as a supportive social environment for deepening skills and creative expression.

3. Design and Analysis of Scratch Collab Camp Stories

The main context for the online collaborative story writing was the Scratch programming tool and website (http://scratch.mit.edu) which lets members share their work with one another. With over 5 million projects shared since its public launch in 2007, the Scratch website is a vibrant online community with over 1,000 new projects being uploaded every day. Scratch is a media-rich programming language that allows youth to design, share, and remix software programs in the form of games, stories, and animations [30]. The Collab Camp was run for four weeks in August (http://info.scratch.mit.edu/collabchallenge). Open to the entire Scratch community, the Collab Camp had three requirements: (1) teams needed a minimum of two participants; (2) teams had to create an interactive story; and (3) teams had to upload an initial draft midway through the competition to receive constructive feedback from the Scratch team before submitting a final project two weeks later. Both draft and final projects were exhibited in a select gallery where participants could view and comment on each other's projects. Further, a selection of 14 Collab Camp projects were featured in rotating fashion on the front page of Scratch.mit.edu.

At least two members of the Scratch Team or specially trained Collab Counselors left constructive criticism on project drafts. Constructive comments were both positive, listing specific things that were good, and constructive, describing things that could be improved, although they were generally limited to one or two comments of 500 characters each, a conventional limitation of the Scratch site. Commenters focused on elements of story (suspense, drama, conflict), usability (instructions, clarity), interactivity, and media (sound, visuals, music). Where relevant, commenters occasionally left specific programming tips for fixing bugs, but in general comments stuck to more thematic ideas regarding story, usability, interactivity, and media.

A total of 33 collabs, each with 2^+ members from the online Scratch community, completed the competition by submitting both a draft and a final version of their Scratch interactive stories, which were collected for analysis In addition, we recorded the constructive comments left on the submitted drafts by the extended Scratch Team and Collab Counselors. Elsewhere we have examined the forms of collaboration for teams who worked together on the Scratch forums or through comments on projects [20], [31] but in this paper we focus solely on the Scratch projects and comments. To understand changes in Scratch stories, we examined the complexity of (1) media production and (2) program code in draft and final Scratch stories.

To analyze the creative expression of the media arts production, we employed a ground-up approach evaluating several elements in the interactive stories [4]. Two members of the group viewed a sub-sample of the 33 projects and began developing a coding scheme to capture different uses of visual and audio media, and types of interactivity. Examples of such features were mini-games and puzzles, visual graphics, emotional triggers (some projects actually made us laugh), characters, plot, and many others. After several iterations and the inclusion of additional features that changed between the draft and final projects, interactivity, story, and media emerged as key categories in the story production, each with several subcategories. For instance, the theme *interactivity* contained six subcategories (mechanism, clicks to alter looks, plot choices, mini-games, puzzles, and other), *story* included six subcategories (character, dialogue, plot-suspense, plot-choices, plot-resolution, and emotion), and *media* consisted of four subcategories (music, sound effects, voices and images). In addition, we captured different levels of the *quality* of creative expression on a scale of 0, 1, 2, or 3. Using these rubrics one researcher scored 28 drafts and 28 final projects[2], calculating subtotals of scores in each area. This scoring was accompanied by qualitative write-ups of each project to provide a more descriptive account of how the project developed from beginning to end.

To analyze the complexity of programming code, we first used an automated tool, called Scrape, to capture code block frequency and then coded by hand to capture code complexity not included by simple frequency accounts of programming scripts. Scrape's main function is to count how many times each different type of code block is used and also within Scratch block categories in each project [36]. We then developed categories of programming code (user interaction, loops, conditional, communication and synchronization, Booleans, variables, and random numbers) developed by Maloney et al's to see the frequency of programming code in draft projects and whether any groups included new categories of coding in their final projects [24]. We also coded by hand a sample of six interactive story projects for a deeper analysis of the use of loops, synchronizations and events, variables/lists, and Booleans and noted how and when project creators synchronized their sprites and which programming techniques were used. We then used this master list as a checklist while going through the rest of our case study projects, noting common programming concepts and the number of times a concept was used in each sprite in a project. In this way we were able to document common techniques used across interactive stories.

4. Creative Expressions in Collaborative Coding of Interactive Stories

The interactive stories solicited by the Collab Camp demonstrated a range of creative expression as well as strong improvements in the quality of storytelling and the sophistication of programming after constructive criticism. Below we describe the kinds of changes collabs made in their stories and programming and the role of constructive criticism in those changes.

4.1. Media Arts Designs

Each collab took the design challenge of interactive storytelling and created something unique, demonstrating that the design challenge allowed for a range of creative expression distinctive to each Scratch group as we will describe more below. Some stories were more game-like (akin to adventure or role-playing games like Legends of Zelda or Mario Brothers), others were essentially games with a strong underlying narrative background, and others were more basic stories that allowed users to choose alternate endings or affect the looks and choices of select characters. The projects also differed greatly in the size and complexity of the stories, demonstrating a range not only in the interests of the creators but in the experiences they brought to the project, with some projects clearly by younger, less experienced Scratchers and others by more senior, highly experienced Scratchers. Nearly all (26 of 28[2]) of the collabs increased the richness of the creative expression in their project between the draft and final versions, primarily through changes made in interactivity, story, and media. In terms of quantity, changes to story and media were the most predominant, though most online collabs increased the amount and type of interactivity used (from plot choices to mini-games within the stories). Most online collabs developed stronger narratives by adding richer character backgrounds, more dialogue, greater and more focused suspense, and more emotional elements to the stories in their final versions. They also tended to utilize music, sound, and visual imagery in more effective ways that substantially increased the player's emotional investment in the story. Overall, constructive criticism seemed to play a strong role in these changes as out of the 28 collabs, 20 made changes in direct response to constructive feedback given. To illustrate the depth of changes made in the quality and creativity of the interactive stories as well as the role constructive criticism played in the process of revision, below we describe two case studies.

The first case we turn to is *Dragonfly Lagoon*, an elaborately designed quest story where a young boy suffering from amnesia seeks the help of dragonflies in a large lagoon to find out who he is (see Figure 1). The main character, Leo, must succeed in several different quests to regain his memory and he receives help from Fi, a dragonfly. As one might expect, in the draft submission the story was unfinished: it began abruptly and provided not background for the character nor a strong sense of Leo's emotional state at having amnesia. The main character Leo did not speak and was simply thrown into the action of an incomplete story where only one task was complete. Along these lines, Collab camp leader Stareyes made the following comment on the draft version: "I can imagine some fun conversations happening between Fi and Leo as they walk and search. Maybe Leo gets discouraged and Fi encourages him. Maybe Fi is a little persnickety at times and teases Leo. A little conversation could make the searching more fun. :)" The collab jumped on these ideas and in the final version utilized an opening cut scene in which the main character, Leo, has an emotional dialog with his new friend, Fi. The dialog gives context for Leo's location and motivation for his actions, and also creates personality for both characters. These changes appear to have been made in direct response to the constructive feedback of Collab Camp counselors like Stareyes. Many other additions were also made to the project, including more visuals, a start screen, a more extended (and complete) story, multiple kinds of background music (which strengthened the emotional tenor of the story), and improved aesthetics of both images and movement. All of these amplified the quality of the interactive story.

Figure 1: Screen shots of *Dragonfly Lagoon* Scratch story

Another project, *Lumos*, illustrates the diverse range of interactive story submitted to the Collab Camp, while the team also responded in significant ways to constructive criticism offered on draft

[2] Due to technical difficulties 5 of the 33 projects were not analyzed for changes in media, story, or interactivity.

versions. The *Lumos* Project was primarily a game with a strong storyline about a ship with different possible crews—pirates, military, and aliens—all fighting for dominion over the Earth (see Figure 2). One Collab Counselor wrote in the comments, “I like the notes of humor—they add a lot. It took me a long time to kill the spaceship, which got a bit boring. Are there instructions? I couldn’t find any. This will be great—keep working on it!” In response, the *Lumos* team added more touches of humor in the dialog and explanations. They also created a detailed tutorial to explain how to play the gaming parts of the story. The team made many additional changes that went beyond the spectrum of the comments, improving the steering and feel of the spaceship, allowing users to customize the crew and ship, extending the storyline significantly, and adding sound effects. Indeed, the depth and richness of changes made to *Dragonfly Lagoon*, *Lumos* and many other interactive stories suggests that our analysis may have short-changed the role of constructive criticism in supporting revisions and learning between the draft and final projects. While it is refreshing to see that Scratch members responded to suggestions, and deepened the quality of their stories on several levels, in the following analyses we examine how these changes played out on the level of code.

Figure 2: Screen shots of *Lumos* Scratch story

4.2. Code Designs

One of the main goals of Collab Camp was to promote more sophisticated programming within the context of rich storytelling. Comparing the programming of draft and final versions of collab projects with Scrape revealed two trends indicative of positive developments in programming. Programming concepts like loops, conditionals, and synchronization and events were used by nearly all projects in the Collab Camp (see Figure 3). Several Collabs introduced more sophisticated programming concepts into their final projects that were not initially present in their draft submissions: 25% of collabs (8 of 33) introduced Booleans, 21% (7 of 33) introduced variables, and 30% (10 of 33) introduced random numbers. These findings gain particular relevance when compared to analyses conducted on projects where Booleans, variables, and random numbers were utilized in fewer than 12% of Computer Clubhouse projects [24]. Furthermore, Booleans (76%), variables (94%), and random numbers (63%) were used by nearly all projects in the Collab Camp story telling, whereas they were used in far fewer numbers in the Computer Clubhouse (48%, 22%, and 16% respectively). At a very basic level, this suggests that the design of collab activity did indeed increase complexity of programming within a story telling context.

We also sought to understand *how* these programming concepts were being used beyond the mere presence of them. Examining six case studies, we found a greater range of foundational programming skills within all categories. All programs used basic *loops* as well as conditional loops whereas five of the projects even used nested loops—situating one loop within another such that a subset of actions would repeat for a certain number of times before going back to the larger loop. All programs used at least one user-created *variable* with quite a range of ways that variables were used—to control movement and appearance, to vary recursive loops, to keep score in mini-games, and to synchronize an event. More importantly, four programs used multiple variables and three projects used lists in effective ways that supported the storytelling. Three projects utilized Booleans and tended to use them in sophisticated ways, embedding multiple Booleans within Booleans (i.e., if this *and* that is true, *or* this *and* that are true, then do an action). Finally, all six case study programs used a variety of ways to synchronize and orchestrate events, including orchestrated timing, broadcasts, variables, and "wait until" commands. In particular, simple levels of thread synchronization can be learned by novice programmers in Scratch in the form of coordinating sprites' actions via broadcast [9], [24].

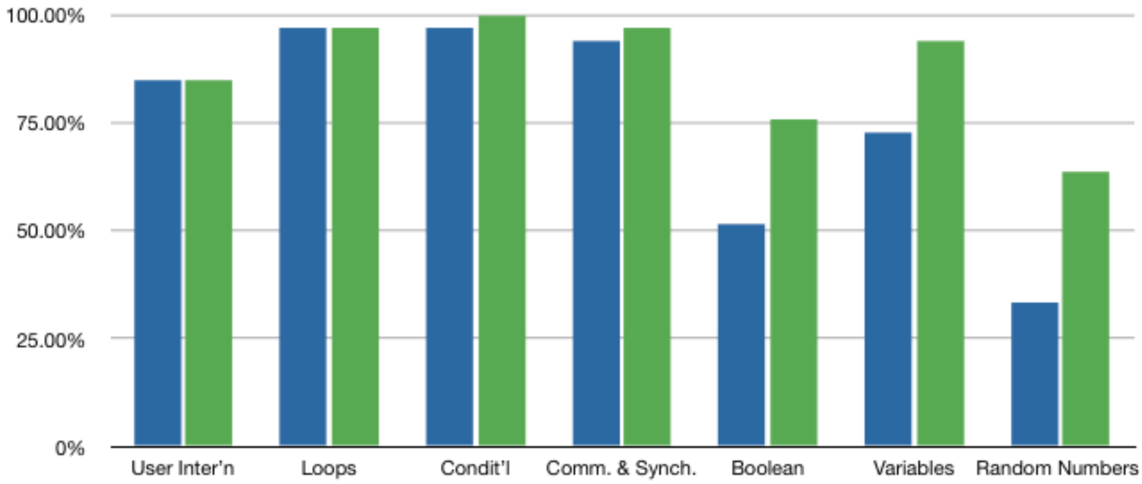

Figure 3: Frequency of programming concepts before and after final story submission

Since the use of 'broadcast' commands lend themselves to event-driven programming, we further investigated their use because novice programmers in Scratch often synchronize events with "wait" (delay) commands. While creating a conversation by having characters wait for a few seconds in between conversational turns is possible, this programming quickly becomes cumbersome. More advanced Scratchers may use the Scratch-based "broadcast" and "when I receive" commands to coordinate actions because broadcast allows one sprite to initiate a signal to one or more sprites, allowing a more abstract and coordinated form of synchronization. Five of the six case study projects utilized "broadcast" and the associated "when I receive" scripts. All five projects increased the number of "when I receive" scripts per broadcast between the first and final submissions. Further, each of the five projects had one sprite that initiated the majority of the "broadcasting," with other sprites doing the majority of the "receiving." Progressing toward using one broadcast for multiple receives and centralizing the broadcasting in a single sprite demonstrates a higher level of abstraction and organization in synchronization and events and exhibits the kind of more sophisticated programming we hoped the interactive story challenge would elicit.

5. Discussion

We investigated the ways in which a collaborative design challenge to create an interactive story could become a context for creative expression through artistic media production and programming. Although interest-driven communities, whether online or locally face-to-face, have received much praise in regard to their potential to help youth learn and engage in producing with digital media in the wild (e.g., [11], [15], [17]), it has been difficult to translate these voluntary engagements to more intentional pursuits. We did so in the two following ways: (1) by situating the story design activity within the larger Scratch online community to engage community members and facilitate social relevance; and (2) by providing constructive criticism at a key mid-point in the challenge to support productive revisions in story projects. The majority of the collabs who submitted both a first and final project made substantive changes in response to the constructive feedback they received, both in the richness of the storytelling through elements of interactivity, narrative, and media, and in the complexity of the programming. The design challenge demonstrates that online communities can be leveraged to support deeper media and programming creation in a socially relevant way.

Our findings further suggest that the Collab Camp provided a meaningful platform for using more sophisticated programming techniques within a story-based, creative context. This in itself is an encouraging finding as it suggests that the interactive story challenge did provide a meaningful context where those more sophisticated techniques were useful. Deeper analysis of a subset of projects revealed not just increased quantity but more sophisticated qualities of using loops, variables/lists, Booleans, and synchronization/events. In particular, the interactive stories appear to encourage sophisticated uses of “broadcasts” in Scratch as a form of event-driven programming. Oftentimes in the literature on engaging youth in technical pursuits, video games, simulations and robotics have taken precedence over more traditional arts like narrative, music, and art [14]. Yet our analysis demonstrates that challenging, sophisticated programming can be facilitated in interactive storytelling. Indeed, those projects that had the highest media/creativity scores were also the ones that had the highest programming scores.

Constructive criticism played an important role [33] in supporting revisions and steering improvements in participants’ stories solicited as part of an online collaborative design challenge. However, analyzing the development of skills and techniques in relation to more artistic media qualities of the interactive stories proved to be a challenge. For one, there is very little literature explicating what counts as more sophisticated programming, especially at novice levels and in informal contexts. More often programming is analyzed in relation to a formal curriculum where everyone completes the same project and it is easy to see who has deviated from the best solution. In contrast, design of narrative elements and graphics were left to Scratch members and thus it was difficult to evaluate how well their interactive stories progressed. Connecting programming techniques and artistic expression in interactive stories was a challenging, yet worthy effort to create contexts for authentic personal expression with digital technology.

The implications from this study suggest that giving kids a design challenge to create interactive stories is an effective way to help them to learn how to synchronize actions and orchestrate events, and in particular how to use “broadcasts.” Stories are in essence a series of cause and effect events told in some kind of narrative. This very characteristic of stories is one reason that they are very effective in learning this particular computational concept. In addition, the added challenge of making the story interactive enhanced the opportunities and need for sophisticated synchronization. Seeing an increase in the amount of centralized coordination is an encouraging finding and suggests that the narrative aspect of the interactive story challenge promoted the use of some challenging programming concepts and techniques. These findings provide design guidelines for how other learning through design activities can be structured for both richer creative expression and programming complexity [20]. In recent work, we have found that collaborative tasks of programming music videos can be designed in such a fashion that computational concepts such as synchronization and initialization integrate collaboration, programming, and creativity in an authentic and meaningful way [10]. Rather than being distinct requirements, the learning of programming and collaboration can become mutually beneficial to each other and realize a core premise of constructionist approaches.

6. Acknowledgments

We wish to thank the MIT Scratch Team, especially Ricarose Roque and Amos Blanton for their role in creating and supporting Collab Camp. Thanks also to Taylor Martin for feedback and support of analysis for this paper. This research has been supported by NSF awards to Yasmin Kafai (#0855868) and to Taylor Martin (#1025243). The views expressed are those of the authors and do not represent the views of the National Science Foundation, Utah State University or the University of Pennsylvania.

References

[1] R. Black, *Adolescents and Online Fan Fiction*. New York: Peter Lang, 2008.

[2] K. Brennan, A. Valverde, J. Prempe., R. Roque, and M. Chung, “More than code: The significance of social interactions in young people's development as interactive media creators,” presented at the World Conference on Educational Multimedia, Hypermedia and Telecommunication, Lisbon, Portugal, 2011.

[3] Q. Burke and Y. B. Kafai, "The writers’ workshop for youth programmers," in *Proceedings of the 43rd SIGCSE Technical Symposium on Computer Science Education*, New York: ACM, 2012, 433-38.

[4] K. Charmaz, “Grounded theory: Objectivist and constructivist methods,” in N. K. Denzin and Y. S. Lincoln (Eds.), *Handbook of Qualitative Researc*h, Thousand Oaks, CA: Sage Publications, 2000, pp. 509-53.

[5] M. Csikszentmihalyi, *Creativity: Flow and the psychology of discovery and invention*. New York, NY: Harper Collins, 1996.

[6] M. Csikszentmihalyi, and R. Wolfe, “New conceptions and research approaches to creativity: Implications of a systems perspective for creativity in education”. In K. A. Heller, F. J. Manks, R. Subotnik, and R. J. Sterberg (Eds.) *International handbook of giftedness and talent*, Oxford: Elsevier, 2001, pp. 81-93.

[7] J. Denner, and L. Werner. "Computer Programming in Middle School: How Pairs Respond to Challenges." *Journal of Education Computing Research* 37, no. 2, 131-50, 2007.

[8] J. Denner, L. Werner, and E. Ortiz, “ Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts? *Computers & Education,* 58(1), 240-249, 2012.

[9] C. L. Fadjo, “Developing computational thinking through grounded embodied cognition. Ph.D. dissertation. Columbia University, New York, New York, 2012.

[10] D. A. Fields, V. Vasudevan, and Y. B. Kafai, “The Programmers’ Collective: Connecting collaboration and computation in a high school Scratch mashup coding workshop,” *Learning and becoming in practice: ICLS 2014 Conference Proceedings*. Boulder, CO: International Society of the Learning Sciences, 2014.

[11] J. P. Gee, *Situated language and learning: A critique of traditional schooling*. New York, NY: Routledge, 2004.

[12] S. M. Grimes, “Persistent and emerging questions about the use of end-user license agreements in children’s online games and virtual worlds,” *UBCL Rev.*, *46*, 681-791, 2013.

[13] S. M. Grimes and D. Fields, *Kids online: A new research agenda for understanding social networking forums.* New York. The Joan Ganz Cooney Center at Sesame Workshop, 2012. [Online]. Available: http://www.joanganzcooneycenter.org/reports-38.html.

[14] E. R. Hayes, and Gee, J. P, “No selling the genie lamp: A game literacy practice in The Sims.” *E-Learning and Digital Media*, *7*(1), 2010. [Online]. Available: http://dx.doi.org/10.2304/elea.2010.7.1.67

[15] M. Honey, and D. Kanter, eds., *Design, Make, Play: Growing the Next Generation of STEM Innovators*. New York: Routledge, 2013.
[16] M. Ito, S. Baumer, M. Bittanti, d. boyd, R. Cody, B. Herr, H. A. Horst, P. G. Lange, D. Mahendran, K. Martinez, C. J. Pascoe, D. Perkel, L. Robinson, C. Sims, and L. Tripp. *Hanging Out, Messing Around, Geeking Out: Living and Learning with New Media*. Cambridge, MA: MIT Press, 2009.
[17] H. Jenkins, K. Clinton, R. Purushotma, A. Robison, and M. Weigel. *Confronting the Challenges of Participatory Culture: Media Education for the 21st Century*. Chicago: MacArthur Foundation, 2006.
[18] Y. B. Kafai, *Minds in Play: Computer Game Design as a Context for Children's Learning*. Mahwah, NJ: Lawrence Erlbaum, 1995.
[19] Y. B. Kafai, and K. A. Peppler. "Youth, Technology and DIY: Developing Participatory Competencies in Creative Media Production." *Review of Research in Education* 35, 89-119, 2011.
[20] Y. B. Kafai, D. A. Fields, R. Roque, W. Q. Burke, and A. Monroy-Hernández, "Collaborative agency in youth online and offline creative production in Scratch." *Research and Practice in Technology Enhanced Learning,* 7, no. 2, 63-87, 2012.
[21] C. Kelleher, and R. Pausch, "Using storytelling to motivate programming," *Communications of the ACM,* 50(7), 58-64, 2007.
[22] K. Luther, and A. Bruckman. "Leadership and Success Factors in Online Creative Collaboration." *IEEE Potentials*, *30*(5), 27–32, 2011.
[23] A. M. Magnifico, "Writing for whom? Cognition, motivation, and a writer's audience." *Educational Psychologist* 45, no. 3, 167-84, 2010.
[24] J. Maloney, K. Peppler, Y. B. Kafai, M. Resnick, and N. Rusk, "Programming by choice. Urban youth learning programming with Scratch," Presented at the SIGCSE 2008 Conference, Portland, Oregon, 2008
[25] A. Monroy-Hernández, B. Mako Hill, J. Gonzalez-Rivero, and D. Boyd, "Computers can't give credit," in *Proceedings of the 2011 Annual Conference on Human Factors in Computing Systems*, 3421-30. Vancouver, BC: n.p., 2011.
[26] S. Papert, "Situating constructionism," in *Constructionism*, edited by I. Harel and S. Papert, 111-30. Norwood, NJ: Ablex Publishing, 1991.
[27] S. Papert, M*indstorm*. 2nd ed. New York: Basic Books, 1980/1993.
[28] K. Peppler, *New Creativity Paradigm: Arts Education in the Digital Age*. New York, NY: Peter Lang Publishers, 2014.
[29] M. Resnick, *Turtles, Termites, and Traffic Jams*. Cambridge, MA: MIT Press, 1994.
[30] M. Resnick, J. Maloney, A. M. Hernández, N. Rusk, E. Eastmond, K. Brennan, A. D. Millner, E. Rosenbaum, J. Silver, B. Silverman, and Y.B. Kafai. "Scratch: Programming for Everyone." *Communications of the ACM* 52, no. 11, 60-67, 2009.
[31] R. Roque, "Creating technologies of their own: Examining young women's participation in an online programming community," Presented at the *2012 Girls and Digital Culture Conference*. London, UK, 2012.
[32] R. Roque, D. A. Fields, J. Siegal, D. Low an Y. B. Kafai, "A clubhouse of their own: A role- playing game society in Scratch programming community," Presented at the annual conference of the *American Education Researchers Association*, Vancouver, Canada, 2012.
[33] R. Roque, N. Rusk, A. Blanton, "Youth roles and development of leadership in an online creative community," in *Computer Supported Collaborative Learning (CSCL)* Madison, WI, 2013.
[34] K. Sawyer, "Learning how to create: Toward a learning sciences of art and design, in *Proceedings of the 10th International Conference of the Learning Sciences,* Sydney, Australia: International Society of the Learning Sciences, 2012, 33-39.
[35] Sullivan, F. "Robotics and Science Literacy: Thinking Skills, Science Process Skills, and Systems Understanding." *Journal of Research in Science Teaching* 45, no. 3, 373-94, 2008.
[36] U. Wolz, B. Taylor, and C. Hallberg. Scrape: A tool for visualizing the code of Scratch programs," Presented at the ACM SIGCSE, Dallas, Texas, 2010.