The Effects of Providing Starter Projects in Open-Ended Scratch Activities

Merijke Coenraad[†], Jen Palmer[*], David Weintrop[†], Donna Eatinger[*], Zachary Crenshaw[*], Hoang Pham[*], Diana Franklin[*],
[*]University of Chicago, Chicago, IL, USA
[†] University of Maryland, College Park, College Park, MD, USA
{mcoenraa,weintrop}@umd.edu;{jenpalmer,dmeatinger,zcrenshaw,hoangsp,dmfranklin}@uchicago.edu

ABSTRACT

Given the importance of broadening participation in the field of computing, goals of supporting personal expression and developing a sense of belonging must live alongside the goals of conceptual knowledge and developing disciplinary expertise. Integrating opportunities for students to be creative in how they enact computing ideas plays an important role when designing curricula. We examine how student creativity, as expressed through theme and the use of costumes, backdrops, and narrative in Scratch projects, is affected by using a themed starter project. Starter projects are Scratch projects that include a set of sprites and backdrops aligned to a theme (e.g. baseball), but no code. Using within-group and between-group comparisons, we establish a baseline of what students do when they are given a starter project and explore how their projects differ in the absence of a starter project. This work contributes to our understanding of the impacts of structured elements within open-ended learning tasks and how we can design computer science learning experiences for students that promote opportunities for self-expression while engaging them in computing.

CCS CONCEPTS

• **Social and professional topics → K-12 education**; **Computer science education**;

KEYWORDS

computer science education; K-12; Scratch; scaffolding; creativity

ACM Reference Format:
Merijke Coenraad, Jen Palmer, David Weintrop, Donna Eatinger, Zachary Crenshaw, Hoang Pham, and Diana Franklin. 2021. The Effects of Providing Starter Projects in Open-Ended Scratch Activities. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education (SIGCSE '21), March 13–20, 2021, Virtual Event, USA* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3408877.3432390

SIGCSE '21, March 13–20, 2021, Virtual Event, USA

ACM ISBN 978-1-4503-8062-1/21/03…$15.00
https://doi.org/10.1145/3408877.3432390

1 INTRODUCTION

As part of an effort to equitably and sustainably introduce computer science (CS) into K-12 schools, a growing number of CS curricula are being designed to introduce students to computing. Given the emphasis on broadening participation in computing and ensuring large-scale CS education initiatives succeed with all students, many curricula offer opportunities for students to draw on their existing knowledge and situate computing content in contexts that resonate with them. Despite a shared approach of aligning computing content with students' existing interests and providing opportunities for self-expression, there remain questions as to how different curricular structures support this goal. How much freedom should a curriculum provide to students? What are the implications of this flexible structure for students and educators? Given these questions, current CS curricula exist along a continuum ranging from highly-structured, prescriptive activities to open-ended, creative activities.

In our work, we designed Scratch Encore, a curriculum that uses Scratch [26, 38] and follows the Use-Modify-Create pedagogical strategy [23] to introduce upper-elementary students to CS concepts in a culturally responsive way [14]. Units introduce content through themed Modify and Create activities (e.g., a texting conversation). To support students during the Create task, we provided starter projects containing a themed set of assets (e.g. sprites, sprite costumes, and backdrops) and no code. A list of required tasks was provided to ensure projects demonstrated specific CS knowledge.

In this paper, we explore how the use of themed starter projects impacted how students engaged with open-ended Create projects. We investigate if the inclusion of themed sprites, costumes, and backdrops spurred or hindered student creativity. We analyze student-created projects to answer the research question: *How does the use of a themed starter project affect student creativity when authoring a computer program?*

We continue with a review of background literature and a discussion of the theoretical lens we bring to this work (section 2). In section 3 we present our methodology. Then, we present our findings in section 4. First, we set a baseline by examining projects from all participating students when using a starter project. Second, we compare projects from the same unit for students who used themed starter projects and from those who did not use the starter project. Finally, we investigate within-group differences by following the same set of students as they move from using a starter project to creating without one. We conclude by discussing the project's implications and the importance of creativity and self-expression in introductory computing coursework in section 5.

2 THEORY AND PRIOR WORK

Influences including Constructionism, available K-12 CS curricula, and research on creativity in computing inform this work.

2.1 Theoretical Orientation

Scratch Encore is grounded in the Constructionist philosophy that the construction of personally meaningful artifacts is a central learning activity [34]. It posits that students should be in control of their learning and empowered to express themselves. The resulting constructions are more meaningful to the student, resulting in a more personal connection with the content [35]. Creating space in educational activities for students to draw on their interests and prior knowledge also draws from work on resource pedagogies [21, 30, 36], which show these practices are especially impactful for students from non-dominant populations [31].

This approach introduces a tension, called the play paradox, between granting students freedom to explore while ensuring they engage with focal concepts [32]. We resolve this in two ways in Scratch Encore. First, it uses Use-Modify-Create pedagogical approach [23] provides both structured activities (Use/Modify) to focus on content and open-ended activities (Create) that invite personal expression and creativity[13]. Another scaffolded approach is creating halfbaked [20] or Scratch microworlds [44]. These themed, incomplete projects provide an initial structure but leave opportunity for students to extend and personalize projects. Scratch Encore provides themed starter projects where sprites, costumes, and backdrops provide a framework within which the student can explore and build Create projects.

2.2 Background Literature

2.2.1 K-12 CS Curricula. CS curricula range from highly structured, regimented activities to open-ended, creativity-focused activities. Structured curricula, such as from code.org [2], prioritize students working through puzzle-like activities to ensure they demonstrate knowledge of specific concepts. Each puzzle has a correct answer, and students progress as they demonstrate the ability to use the CS concept. More open-ended curricula, like Creative Computing [5], use exploratory activities where students develop understanding of CS concepts as they design and implement projects of their own creation. These curricula rely on platforms where students introduce their own images and audio to personalize the project and incorporate their own ideas. In the middle of the spectrum are curricula like the grade 3-5 Green Curriculum [40], Scratch Act 1 [1], and Scratch Encore[14]. These curricula provide a blend of structured and open-ended activities to enable students to learn the CS content and express themselves creatively. But, even within such an approach, there remains a spectrum of the amount of scaffolding and structure provided. We explore that spectrum to understand how scaffolds, specifically themed starter projects, support students.

2.2.2 Creativity in Coding and Computing Education. Creativity is identified as a CS Principles Big Idea [43] and often motivates students [6, 17, 27, 37]. Given initiatives to broaden participation in computing [42, 46] and the potential of partnering digital technologies and creativity [24], creativity has become a key design concept for CS learning tools [15, 18, 25, 38, 41, 44] as researchers stress the importance of valuing both CS conceptual learning and and the creative vision enacted through their projects [4, 28]. Researchers have found that within learning environments, creativity leads to increased persistence and engagement in computing [3, 18, 29] while allowing students to maintain agency over their work [28].

However, the subjective nature of creativity makes it difficult to assess in computing projects [12]. Teachers assess creativity by focusing on a student's design process and planning, using rubrics, providing opportunities for self-assessment, and making space within projects for both technical and creative aspects, like characters and the project "world" [4]. Based on the validated Creative Product Semantic Scale [33], creativity in Scratch projects has been assessed by examining originality when compared to others within the same Scratch studio; code complexity (e.g., the variety and number of scripts or blocks used); and the diversity of narrative elements (e.g., messages, questions) and visual elements (e.g., the variety and sources of images used in costumes and backdrops) [19]. Here, we build off of this model of assessing creativity.

3 METHODS

We use a quasi-experimental design with qualitative and quantitative analysis to analyze within-group and between-group trends in projects across the use of themed starter projects in two CS units.

3.1 Curriculum

Scratch Encore is a 5th-8th grade (10-14 y.o.) intermediate, Scratch-based [26, 38] CS curriculum [14] developed through a researcher practitioner partnership [7] to bridge the gap between coding experiences in elementary and high school. It is designed to be culturally relevant to students, accessible to teachers, and flexible for varied school contexts. The curriculum incorporates themes and ideas gathered from participatory design sessions [8–10] with students, teachers, parents, and administrators from our partner district and selected in partnership with district practitioners.

Scratch Encore follows Use-Modify-Create [23] to scaffold student learning through a process of gradual release [13]. First, students Use example code (which students are familiarized to through the TIPP&SEE strategy [39]) focused on a single topic. Students then Modify the same Scratch project. The module ends with a Create task with no starting code: building a project that meets a set of requirements in a context of interest to students. Students are provided idea prompts or a themed starter project with assets (e.g., sprites, sprite costumes, backdrops). All activities are mediated through worksheets that provide student prompts and task checklists. To support teachers, the curriculum provides lesson plans with discussion prompts, potential student answers, and lesson flow.

We examine students' projects when they do or do not use themed starter projects in the basic loops and conditional loops units. The starter project for the basic loops unit had a dance party theme. The starter project included seven visible "dancer" sprites and five backdrop options. Each sprite had between three and 13 available costumes. The starter project for the conditional loops unit had a transportation theme. The project included two visible sprites, a vehicle sprite, and a sign sprite with five backdrops of urban transportation locations. The vehicle sprite included nine costumes and the sign sprite included three.

3.2 Participants and Context

We examined the Scratch projects of 160 5th-8th grade (10-14 y.o.) students who attended schools in a large metropolitan school district in the Midwestern United States and completed both the basic loops and conditional loops units of Scratch Encore. Students were taught by pilot teachers who participated in a three-day professional development workshop the prior summer about Scratch Encore, its CS content, TIPP&SEE, and Use-Modify-Create.

Group A consisted of 78 students from three class sections taught by the same teacher; one class in each grade level, 6th, 7th, 8th. This group used the themed starter project for the create tasks in the basic loops unit but did not use the themed starter project in the conditional loops unit.

Group B consisted of a similarly-sized (82 students) comparison group of similarly-achieving students who used themed starter projects for both units. It was constructed of students in five 5th-7th grade classes taught by four teachers. Classes were selected based on the portion of project requirements completed by students in the class. The average completion of project requirements within both units combined for Group B were within 5% of of Group A.

3.3 Data Collection and Analysis

We collected and viewed student-created Scratch projects from all classrooms through classroom studios on Scratch. Teachers provided links to a studio of their specific class and the Scratch projects within that link were catalogued.

To analyze the student Scratch projects, we inductively generated a codebook of relevant project attributes (e.g., number of sprites, sprite origin, theme, use of narration, backdrop, inclusion and functioning enactment of the CS concepts). For each of the three types of projects (basic loops, conditional loops with a starter project, conditional loops without a starter project), the researchers first discussed the codebook and coded one project from each condition together. Then, the researchers individually coded 20% of the projects in each condition. Following initial coding, the researchers met to discuss discrepancies. Interrater reliability (IRR), calculated by project type using Fleiss' Kappa, was between 0.876 and 0.958, within the almost perfect agreement range [22]. The remaining projects were split between the coders for analysis.

We performed exploratory data analysis to determine overall trends in all projects created by students. In a few cases, this included students submitting two different projects, which were both examined. Indicators of interest were analyzed to determine statistical significance using the Mann-Whitney U test for quantitative dependent variables and Fisher's Exact Test for categorical dependent variables.

4 RESULTS

In this study, we performed three analyses to examine the effects of using themed starter projects on students' Scratch projects.

(1) A between-group analysis of the use of starter projects in the basic loops unit across participants in Group A and Group B.
(2) A between-group analysis of projects in the conditional loops unit from Group A without starter projects and Group B with starter projects.

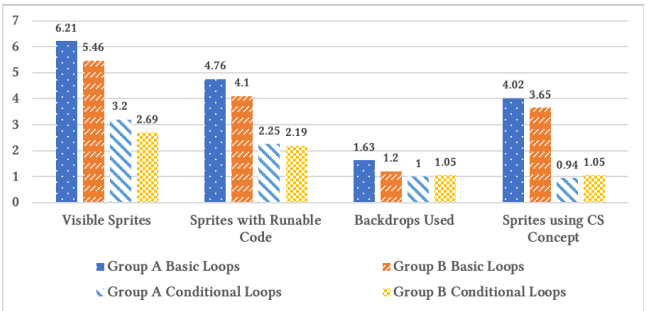

Figure 1: Mean Average Sprites and Backdrops by Project

(3) A within-group analysis of projects from Group A with starter projects in the basic loops unit and without starter projects in the conditional loops unit.

We present findings from each and discuss their implications on the use of themed starter projects.

4.1 Between-Group: Basic Loops

A between-group analysis of the basic loops unit in which both groups used themed starter projects revealed similar trends in student projects across groups, although it also revealed some differences. Typical projects across both groups situated a collection of sprites against one of the provided backdrops.

Finding 1: Students in both groups used similar numbers of backdrops and animated a similar number of sprites, but showed and programmed different numbers of sprites.

Students tended to use one backdrop, and there was no significant difference between the two groups (median A = 1, B = 1; U = 3429.5, $P = 0.21$; Figure 1). Group A projects used a median seven visible sprites and a median 4.5 sprites with functioning code. This is a significant difference compared to Group B projects that used a median five sprites and a median of three sprites with functioning code (U = 3852.5, $P = 0.03$; Figure 1). Despite this difference, students in both groups animated a similar number of sprites (median A = 3.5, B = 3; U = 3540.5, $P = 0.29$; Figure 1). Sprites were considered animated if a loop was used to perform a repeated action, either in place or with movement.

Finding 2: Projects across groups maintained the dance theme of the starter project, but in both groups students incorporated additional youth culture and nonsensical elements.

Projects from both groups maintained the starter project dance theme (A = 97.44% B = 97.59%, $P = 1.00$; Figure 2), but some incorporated additional youth culture references. Students included references to characters or celebrities from popular media and references to internet memes. 18.18% of projects from students in Group A and 7.23% of students from Group B included additional youth culture within their projects ($P = 0.054$). Some students also incorporated nonsensical elements in their projects (i.e., the projects were intentionally silly, contained no cohesive narrative or theme, or incorporated a hodgepodge of characters). There was no significant difference in the inclusion of these nonsensical elements across groups (A = 24.39%, B = 14.46%, $P = 0.16$).

Finding 3: Students in Group A used narrative significantly more than students in Group B.

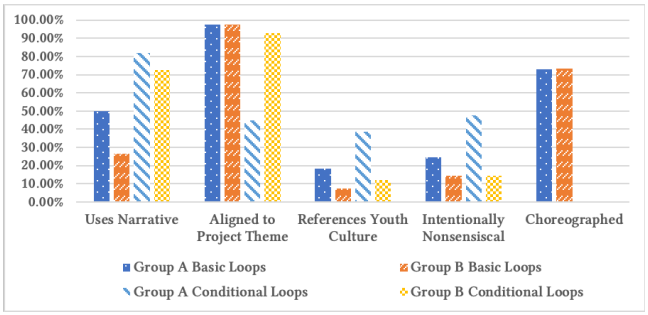

Figure 2: Use of Narration, Theme, and Animation by Project

One significant difference between groups was the use of narrative (Figure 2; P = 0.003). Projects in both groups used say blocks to convey narrative, but Group A did so more frequently (50% vs. 26.51%). Students created narrative through one- or two-way dialogue between sprites and directed narrative to the user, either as a listener (e.g, "DANCE BATTLE!") or to provide the user with instructions (e.g., "Click on me!").

Finding 4: Students in both groups found ways to individualize projects within the starter project theme.

Students found ways to differentiate their projects and express creativity within the starter project theme. Students in both groups used choreography in their projects (A=73.08%, B=73.49%, P = 1.00). Choreography included manipulated costume order or selection, synchronized timing or dance moves across sprites, and sprite organization on stage. Project differentiation was also accomplished through modifying, uploading, and drawing costumes (Figure 3 Top) and backdrops (Figure 3 Bottom). Almost all projects retained at least one provided sprite (A = 97.4%, B = 97.6%). Group B projects (16.87%) modified provided sprite costumes significantly more than Group A projects (0.00%; P<0.001). Group B projects were more likely to add new costumes from the Scratch library (B = 32.53%, A = 16.88%, P = 0.03). Group A projects were significantly more likely to use a provided backdrop (70.51%, P < 0.001) while Group B projects elected to use Scratch Library backdrops (59.04%, P= 0.004). For example, one project that included uploaded sprites and those from the starter project showed the provided sprites arranged alongside a meme character and a Fortnite character.

4.2 Between-Group: Conditional Loops

In the conditional loops unit, students in Group B used the themed starter project, but students in Group A did not. The Group A teacher chose not to use the transportation themed starter project because many of her students couldn't relate to driving or being in cars so she "wanted them to choose...something they knew more." Next, we examine the similarities and differences between Group A and Group B projects, highlighting differences in student expression, individualization, and creativity.

Finding 5: Student projects created without starter projects differed aesthetically and thematically from those with starter projects, but used CS concepts in similar ways.

Projects included different numbers of visible sprites between the two groups (median A = 2, B = 3; U = 2392, P = 0.002), but similar numbers of sprites with functioning code (median A = 2, B = 2; U = 3432, P = 0.58) and backdrops (median A = 1, B = 1,

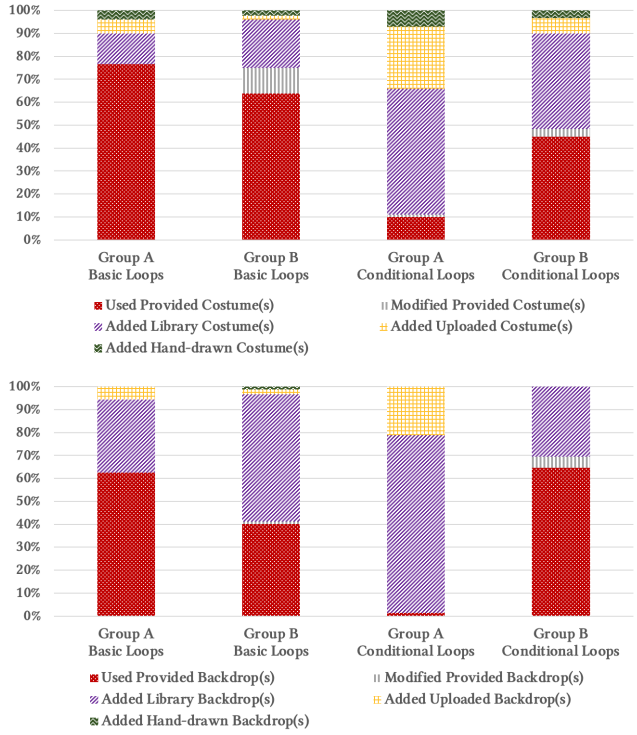

Figure 3: Costume and Backdrop Origin by Project

Figure 4: Transportation projects in Groups A and B

U = 3402, P = 0.07). The number of sprites with functioning code enacting CS focus, conditional loops, was approximately equal between groups (median A = 1, B = 1; U = 3584.5, P = 0.24; Figure 1). Yet, the manner in which students used the assets and overall theme of their projects differed greatly. Significantly more students in Group B used the transportation theme of the starter project (92.86% vs.44.87%, P< 0.001). Group A projects that used a transportation theme did so in different ways. While transportation-themed Group B projects typically involved vehicles arriving at a station (Figure 4b), transportation-themed Group A projects were less uniform (e.g., a Lamborghini with a llama riding on its roof; Figure 4a).

Group A students were also more likely to reference youth culture in their projects (38.46% vs. 11.90%, P < 0.001; Figures 2 5a). In both groups, projects referenced a LeBron James "Sprite Cranberry" meme. A Group A student recreated another meme featuring a K-pop star. Many projects incorporated popular culture elements through uploaded sprite costumes or references to student activities. Examples from Group A included a project that depicts a scene in which Mickey Mouse rides in a Lamborghini to get bubble tea and

Figure 5: Examples of projects using Conditional Loops

JoJo Siwa at Target. Several projects referenced video game characters like Kirby and Pokémon. Multiple projects referenced popular music, including Group A projects that mention a boy band.

Group A students expressed themselves through nonsensical elements more often than Group B (A = 47.44%, B = 14.29%, $P<0.001$), exemplified by a project with a giraffe and a pair of Apple AirPod headphones floating in the air above mountains; the giraffe says "Come here airpods" and moves across the screen until it reaches them; at the same time, a toy train moves across the bottom of the screen until it reaches a stop sign and announces "Welcome to Chinatown the 40th state of Antarctica" (Figure 5b).

Students expressed their unique themes through the sprites and backdrops they used (Figure 3). Rather than starting with an asset-filled starter project, Group A students began with a new Scratch project, a white screen with "Scratch Cat". While 80.95% of Group B projects used a provided costume (vehicle or stop sprite), only 12.82% of Group A students kept the provided sprite in their projects, a significant difference ($P<0.001$). Over 70% of students from each group added library sprites to their projects. In Group B, this often meant adding sprites to those already in the project. A difference between the two groups was the rate at which they added uploaded assets. Few Group B students uploaded costumes (12.1%) or backdrops (0.00%), whereas significantly more Group A students added each (costumes = 34.62%, $P= 0.002$; backdrops = 19.23%, $P<0.001$). These differences are exemplified by a Group A project using a library backdrop and two uploaded sprites, Thomas the Tank engine and a stick figure (Figure 5c), and a Group B project that did not use the starter assets, instead adding a backdrop and two car sprites from the library (Figure 5d).

Students across groups expressed more narrative in their conditional loops projects than in the basic loops projects (Figure 2). But, the themes, content, and length of narrative varied between groups. Group B students worked within the transportation theme and tended to create narratives about a train or bus arriving late. Group A project narrative was more varied (e.g., a scene depicts a hare and skeleton in a kung fu battle narrated by an apple).

4.3 Within-Group: Group A

Differences exist in how Group A students express themselves and enact creativity across projects from the two units. Three types of choices stand out: election of themes (including references to youth culture and nonsensical elements), selection of costumes and backdrops, and use of narrative.

Finding 6: Starting without a themed starter project gave students the flexibility to choose a theme and incorporate youth culture and nonsense within their projects.

When students in Group A did not have a themed starter project, they created projects with a broader range of themes. Nearly all basic loops projects built from a starter project retained the original dance theme (97.44%), but themes were more varied when no starter was used, although 44.87% of students maintained the transportation theme of the modify project"(Figure 2). With the freedom to choose a theme, students incorporated a wide range of assets. Some themes were cohesive across a project, creating a story or scene (e.g., a scene in which a sprite talks in the style of a nature documentary). Other projects were a collage of objects from the Scratch libraries or popular media in a setting that may or may not be conventional for that set of sprites (Figure 5a & b).

When they did not use a starter project and could choose a theme, Group A students had greater opportunity to reference youth culture. Twice as many projects (30 as compared to 14 with the starter) referenced youth culture when a starter project was not used (Figure 2). These youth culture references demonstrate the unique interests of students. The projects that did not begin with a starter also demonstrate students' proclivity to create nonsensical projects (44.87% did so). This nonsense was paramount to students expressing themselves and their creativity (Figure 5a & b).

Finding 7: Students created variation within their projects by selecting and uploading specific sprites and backdrops.

Students' themes were accomplished by selecting and uploading specific sprites, costumes, and backdrops (Figure 3). While sensible that more students used a Scratch library costume (70.51%) or backdrop (70.51%) when they started with a blank project, there was also an increase in the number of projects with uploaded sprites (34.62%) and backdrops (19.23%) and students were exact about the assets they used. For example, rather than any car, students brought in specific models, as seen in Figure 4a. The ability to upload assets made the inclusion of youth culture possible, with many uploaded assets being characters from memes, shows, and video games. This transformed student projects from generic (e.g., those created by students who began with the same set of sprites) to distinct (Figure 4a-c). While some students may have inspired others in their class by sharing ideas as they worked, no two projects created without the starter projects were the same, meaning students incorporated their own interests and creativity.

Finding 8: Students used narrative, which grew more varied when they did not use themed starter projects, to express themselves within projects.

Group A expressed themselves through narrative whether a themed starter project was used or not. But narrative use in the two units was notably different. In basic loops, Group A students aligned narrative to the dance theme, in contrast to the varied narrative themes in conditional loops. Basic loops projects from Group

A centered narrative around the dance theme to give instructions or created a scene with dialogue between sprites or more often outward to the user (e.g., "OOOOOOOO Them Moves!"). In conditional loops, when Group A students did not use a themed starter project, narrative themes were far more varied (e.g., a witch visits a food truck outside a Taco Bell: "Oh i want a taco!", "hmm i would like a beef taco with hot sauce.", and "TYSM!!!!!!".

5 DISCUSSION

Given the importance of creativity in broadening participation and allowing students to connect to the curriculum [21, 30, 31, 34–36], it is vital to consider the balance of structure and creativity within any CS curriculum, especially an introductory one. Alongside the need to include creative opportunities, researchers cite the need to balance structure and creativity [4, 23, 28]. Previous work discusses giving students starter projects [16, 45], but few conclusions are drawn about the effect of those starter projects on students' final projects experiences. Other work using starter projects found students desired to expand upon themes and left the Microworld to gain access to more assets [44]. This work builds on the field's knowledge of how the balance between structure and creativity can take place. While there is no correct answer for how to introduce students to CS, this analysis demonstrates themed starter projects do not affect student's incorporation of focal CS concepts, but they do change the types of assets and themes used within student projects. The use of themed starter projects thereby affects students' opportunity to incorporate creativity and showcase their interests through their selection of sprites, backdrops, and themes and the use of animation and narrative.

We used between-group and within-group analyses across two computing units. In the first unit, basic loops, where both groups used the themed starter projects, we observed minor differences between the two groups, but overall alignment between the groups and a tendency for all students to maintain the dance party theme, sprites, and backdrops provided to them. In the conditional loops unit, where Group A did not use a starter project and Group B did, we saw similarities in the number of sprites and backdrops used, but differences in how those sprites and backdrops were used and where they came from (provided vs. library or uploaded). The students not given a starter project included a greater variety of themes, demonstrated in their asset choices and narration. When comparing within Group A across the two units when first they used a starter project and after they did not, we saw more creative elements and self-representation in the second project.

Despite differences in assets, themes, and narration of students' projects, inclusion of the CS topics, basic loops and conditional loops, remain about the same across all comparisons. Students animated or used conditional loops at the same frequency whether or not they received a themed starter project. While alignment between classes is predictable since classes were selected to have similar achievement levels, it demonstrates that students' scaffolded enactment of CS concepts within their projects does not appear to be affected by the use of a themed starter project. With attention toward continuing questions about the extent of scaffolding necessary when enacting the Use-Modify-Create pedagogical approach [23] and the balance between teaching CS concepts and allowing for creativity in the play paradox [32], these findings point toward themed starter projects as being unnecessary in supporting students to finish open-ended projects. While potentially beneficial for students who need support getting started and helpful in streamlining the amount of time students take to create their projects, these themed starter projects appear to have limited or no effect on integration of computing concepts into students' work. Conversely, themed starter projects affect student creativity and the rate at which students incorporate uploaded assets and self-generated themes in their projects. Thus, CS educators, researchers, and curriculum developers should carefully consider whether to include themed starter projects based on the potential benefits to and needs of their specific student populations.

6 LIMITATIONS

The main limitation of our work is the limited sample of student projects we were able to examine. Since all Group A students came from the same school and teacher, though across three grades, these findings may be unique to that context. Additionally, since each teacher presented the materials in a unique context, it is possible that, despite having the same curricular materials, implementation may have varied by teacher. Future research should consider how the use of themed starter projects affects student creativity across multiple settings.

7 CONCLUSIONS

As CS expands into K-12 classrooms, it is important to balance structure with opportunities for students to express themselves for engagement and to create a curriculum that is responsive to students. Projects created without a themed starter project gave students the opportunity to customize their creations and reflect mastery of CS content without having a "standard" appearance. Students represented themselves in their work and expressed their individuality alongside computing ideas. Given the importance of students being able to integrate their interests and ways of knowing into their school work to create a culturally responsive computing environment [11], the opportunity to express themselves within their projects is important for all students. Students must have opportunities to express themselves in their projects with some scaffolds and structures, but too much structure could impede their creativity. Whether or not students used a themed starter project does not appear to affect the rate at which students integrate CS concepts into their final projects. To support students in expressing themselves and add creativity to their projects, especially when they are first learning CS, themed starter projects might not be a beneficial scaffold. Collectively, this work advances our understanding for pedagogical strategies to support student creativity while engaging in CS instruction. These findings contribute to the ever growing knowledge regarding scaffolding within K-12 CS classrooms and strategies for supporting students to learn CS content and represent their full selves in introductory computing experiences.

8 ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 1738758.

REFERENCES

[1] 2018. Scratch Act 1. https://www.canonlab.org/scratchact1modules

[2] 2020. Code.Org. https://curriculum.code.org/

[3] Joel C Adams and Andrew R Webster. 2012. What do students learn about programming from game, music video, and storytelling projects?. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education*. 643–648.

[4] Veno Brennan, Haduong. 2020. Assessing Creativity in Computing Classrooms. *[White Paper]* (2020).

[5] Karen Brennan, Christian Balch, and Michelle Chung. 2014. Creative computing. *Harvard Graduate School of Education* (2014).

[6] Quinn Burke and Yasmin B Kafai. 2012. The writers' workshop for youth programmers: digital storytelling with scratch in middle school classrooms. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education*. 433–438.

[7] Cynthia E Coburn, William R Penuel, and Kimberly E Geil. 2013. Research-practice partnerships: A strategy for leveraging research for educational improvement in school districts.

[8] Merijke Coenraad, Jen Palmer, Diana Franklin, and David Weintrop. 2019. Enacting identities: Participatory design as a context for youth to reflect, project, and apply their emerging identities. In *Proceedings of the 18th ACM International Conference on Interaction Design and Children*. 185–196.

[9] Merijke Coenraad, Jen Palmer, Diana Franklin, and David Weintrop. 2019. Utilizing Participatory Design to Develop a Culturally Relevant Computer Science Curriculum. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. 1261–1261.

[10] Merijke Coenraad, David Weintrop, Donna Eatinger, Jen Palmer, and Diana Franklin. 2020. Identifying Spheres of Influence for a Culturally Relevant Computing Curriculum through Participatory Design. (2020).

[11] Ron Eglash, Juan E Gilbert, and Ellen Foster. 2013. Toward culturally responsive computing education. *Commun. ACM* 56, 7 (2013), 33–36.

[12] Deborah A Fields, Yasmin B Kafai, Anna Strommer, Elissa Wolf, Bailey Seiner, et al. 2014. Interactive storytelling for promoting creative expression in media and coding in youth online collaboratives in Scratch. *Proceedings of constructionism* (2014), 19–23.

[13] Diana Franklin, Merijke Coenraad, Jennifer Palmer, Donna Eatinger, Anna Zipp, Marco Anaya, Max White, Hoang Pham, Ozan Gökdemir, and David Weintrop. 2020. An Analysis of Use-Modify-Create Pedagogical Approach's Success in Balancing Structure and Student Agency. In *Proceedings of the 2020 ACM Conference on International Computing Education Research*. 14–24.

[14] Diana Franklin, David Weintrop, Jennifer Palmer, Merijke Coenraad, Melissa Cobian, Kristan Beck, Andrew Rasmussen, Sue Krause, Max White, Marco Anaya, et al. 2020. Scratch Encore: The Design and Pilot of a Culturally-Relevant Intermediate Scratch Curriculum. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. 794–800.

[15] Yasmin B Kafai, Deborah A Fields, Debora A Lui, Justice T Walker, Mia S Shaw, Gayithri Jayathirtha, Tomoko M Nakajima, Joanna Goode, and Michael T Giang. 2019. Stitching the Loop with Electronic Textiles: Promoting Equity in High School Students' Competencies and Perceptions of Computer Science. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. 1176–1182.

[16] Yasmin B Kafai, Eunkyoung Lee, Kristin Searle, Deborah Fields, Eliot Kaplan, and Debora Lui. 2014. A crafts-oriented approach to computing in high school: Introducing computational concepts, practices, and perspectives with electronic textiles. *ACM Transactions on Computing Education (TOCE)* 14, 1 (2014), 1–20.

[17] Caitlin Kelleher and Randy Pausch. 2007. Using storytelling to motivate programming. *Commun. ACM* 50, 7 (2007), 58–64.

[18] Caitlin Kelleher, Randy Pausch, and Sara Kiesler. 2007. Storytelling alice motivates middle school girls to learn computer programming. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 1455–1464.

[19] Anastasia Kovalkov, Avi Segal, and Kobi Gal. 2020. In the Eye of the Beholder? Detecting Creativity in Visual Programming Environments. *arXiv preprint arXiv:2004.05878* (2020).

[20] Chronis Kynigos et al. 2007. Half-baked logo microworlds as boundary objects in integrated design. *Informatics in Education-An International Journal* 6, 2 (2007), 335–359.

[21] Gloria Ladson-Billings. 2014. Culturally relevant pedagogy 2.0: aka the remix. *Harvard Educational Review* 84, 1 (2014), 74–84.

[22] J Richard Landis and Gary G Koch. 1977. An application of hierarchical kappa-type statistics in the assessment of majority agreement among multiple observers. *Biometrics* (1977), 363–374.

[23] Irene Lee, Fred Martin, Jill Denner, Bob Coulter, Walter Allan, Jeri Erickson, Joyce Malyn-Smith, and Linda Werner. 2011. Computational thinking for youth in practice. *Acm Inroads* 2, 1 (2011), 32–37.

[24] Avril Loveless. 2002. Literature review in creativity, new technologies and learning. (2002).

[25] Brian Magerko, Jason Freeman, Tom Mcklin, Mike Reilly, Elise Livingston, Scott Mccoid, and Andrea Crews-Brown. 2016. EarSketch: A STEAM-based approach for underrepresented populations in high school computer science education. *ACM Transactions on Computing Education (TOCE)* 16, 4 (2016), 1–25.

[26] John Maloney, Mitchel Resnick, Natalie Rusk, Brian Silverman, and Evelyn Eastmond. 2010. The scratch programming language and environment. *ACM Transactions on Computing Education (TOCE)* 10, 4 (2010), 1–15.

[27] John H Maloney, Kylie Peppler, Yasmin Kafai, Mitchel Resnick, and Natalie Rusk. 2008. Programming by choice: urban youth learning programming with scratch. In *Proceedings of the 39th SIGCSE technical symposium on Computer science education*. 367–371.

[28] Jaeger McClure. 2020. Assessing Creativity: A palette of possibilities. *[White Paper]* (2020).

[29] Tom McKlin, Brian Magerko, Taneisha Lee, Dana Wanzer, Doug Edwards, and Jason Freeman. 2018. Authenticity and personal creativity: How EarSketch affects student persistence. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*. 987–992.

[30] Luis C Moll, Cathy Amanti, Deborah Neff, and Norma Gonzalez. 1992. Funds of knowledge for teaching: Using a qualitative approach to connect homes and classrooms. *Theory into practice* 31, 2 (1992), 132–141.

[31] Na'ilah Suad Nasir, Ann S Rosebery, Beth Warren, and Carol D Lee. 2006. Learning as a cultural process: Achieving equity through diversity. (2006).

[32] Richard Noss and Celia Hoyles. 1996. *Windows on mathematical meanings: Learning cultures and computers*. Vol. 17. Springer Science & Business Media.

[33] Karen O'Quin and Susan P Besemer. 1989. The development, reliability, and validity of the revised creative product semantic scale. *Creativity Research Journal* 2, 4 (1989), 267–278.

[34] Seymour Papert. 1980. Mindstorms: Computers, children, and powerful ideas. *NY: Basic Books* (1980), 255.

[35] Seymour Papert and Idit Harel. 1991. Situating constructionism. *Constructionism* 36, 2 (1991), 1–11.

[36] Django Paris. 2012. Culturally sustaining pedagogy: A needed change in stance, terminology, and practice. *Educational researcher* 41, 3 (2012), 93–97.

[37] Kylie Peppler, Erica Rosenfeld Halverson, and Yasmin B Kafai. 2016. *Makeology: Makers as Learners (Volume 2)*. Vol. 2. Routledge.

[38] Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, et al. 2009. Scratch: programming for all. *Commun. ACM* 52, 11 (2009), 60–67.

[39] Jean Salac, Cathy Thomas, Chloe Butler, Ashley Sanchez, and Diana Franklin. 2020. TIPP&SEE: A Learning Strategy to Guide Students through Use-Modify Scratch Activities. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. 79–85.

[40] San Francisco Unified School District (SFUSD) Computer Science Team. 2018. SFUSD's Creative Computing Curriculum. https://sites.google.com/sfusd.edu/3-5cs/green

[41] Kristin A Searle, Deborah A Fields, Debora A Lui, and Yasmin B Kafai. 2014. Diversifying high school students' views about computing with electronic textiles. In *Proceedings of the tenth annual conference on International computing education research*. 75–82.

[42] Megan Smith. 2016. Computer science for all. https://doi.org/10.1145/2524713.2524716

[43] The College Board. 2017. *AP Computer Science Principles*. Technical Report.

[44] Moran Tsur and Natalie Rusk. 2018. Scratch microworlds: designing project-based introductions to coding. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*. 894–899.

[45] Joshua L Weese, Russell Feldhausen, and Nathan H Bean. 2016. The impact of STEM experiences on student self-efficacy in computational thinking. (2016).

[46] Brenda Darden Wilkerson. 2018. The evolution before the revolution. In *SIGCSE'18: The 49th ACM Technical Symposium on Computer Science Education*. ACM, 1. https://doi.org/10.1145/3159450.3166085