Evaluating the Effect of Using Physical Manipulatives to Foster Computational Thinking in Elementary School

Ashish Aggarwal
Dept. of Computer & Info. Science & Engineering
University of Florida
Gainesville, FL 32611
ashishjuit@ufl.edu

Christina Gardner-McCune
Dept. of Computer & Info. Science & Engineering
University of Florida
Gainesville, FL 32611
gmccune@ufl.edu

David S. Touretzky
Computer Science Dept.
Carnegie Mellon University
Pittsburgh, PA 15213
dst@cs.cmu.edu

ABSTRACT

Researchers and educators have designed curricula and resources for introductory programming environments such as Scratch, App Inventor, and Kodu to foster computational thinking in K-12. This paper is an empirical study of the effectiveness and usefulness of tiles and flashcards developed for Microsoft Kodu Game Lab to support students in learning how to program and develop games. In particular, we investigated the impact of physical manipulatives on 3rd – 5th grade students' ability to understand, recognize, construct, and use game programming design patterns. We found that the students who used physical manipulatives performed well in rule construction, whereas the students who engaged more with the rule editor of the programming environment had better mental simulation of the rules and understanding of the concepts.

CCS Concepts

• **Social and professional topics ~ Model curricula; K-12 education; Computational thinking**

Keywords

Kodu; Physical Manipulatives; Tiles; Flashcards; Computational Thinking; Curriculum Design; K-12

1. INTRODUCTION & BACKGROUND

Visual programming environments such as Alice [3] and Scratch [10] have been helpful in introducing younger students to programming and reducing the inherent barriers in learning text-based programming languages. Over the past 5 years, there has been a significant increase in the development of curricula and resources to help K-12 students learn computer science fundamentals and to foster computational thinking using these programming environments [3, 9, 13]. Included in these resources are student activities, assessments, projects, and teaching tips. In many classrooms, teachers use CS Unplugged activities that feature kinesthetic and interactive elements to help students understand the concepts [2]. Several curricula have developed quick reference guides [9, 10, 12] and flashcards [1, 11, 17] to help students quickly identify key concepts or algorithms for implementing common programming design patterns (e.g., save data to a variable, program an autonomous sprite, or draw a shape). The goal of these resources is to help students develop their understanding of computational principles away from the distractions of the programming environment and give them tangible ideas for creating meaningful artifacts [7, 14].

While the use and development of physical manipulatives in CS is growing, physical manipulatives have been around for years in other disciplines. Physical manipulatives like Cuisenaire rods and algebra rods have long been recommended and used in K-12 Mathematics. These tools help students to learn mathematical concepts by making the leap from "intuitive to logical thinking, from the concrete to the abstract" [6]. The affordances of these physical manipulatives, such as size and shape, influence how students use these tools and how they develop conceptual understanding and chunking strategies [8]. In literacy education, "physical manipulatives are physical objects that aid understanding of concepts or processes by allowing students to physically demonstrate and see the concept or process. The use of manipulatives provides a way for students to learn concepts in a developmentally-appropriate, hands-on, experiential way" [4].

As CS educators continue to design and refine physical manipulatives to improve students' CS learning, it is important to develop an understanding of how students interact with these resources, how to best use these resources to support student learning, and how to measure the impact that these resources have on student learning. The goal of this paper is to explore the use of tiles and flashcards by 3rd – 5th grade students within a Kodu curriculum [16]. In particular, we aim to measure the impact of tiles and flashcards [17] on students' ability to understand, recognize, construct, and use game programming design patterns. Based on our analysis, we make recommendations for the optimal use of these resources to support learning and skill development.

2. KODU AND KODU CURRICULUM

2.1 Microsoft's Kodu Game Lab

Kodu Game Lab is a visual programming language made specifically for 3D game development. It is designed to be accessible to children and enjoyable for anyone. The programming environment runs on the Xbox, Windows PC, and tablet platforms. It provides students with a 3D world to visualize the behavior of their programs and a rule editor to design and rapidly iterate on their programs using an Xbox game controller or keyboard for input (Figure 1). Kodu uses a tile-based language based on WHEN-DO conditional rules to control characters and objects in 3D worlds. The rules are organized using a sequence of tiles (e.g., objects, perceptions, and actions) to create conditional statements. For example, the equivalent of a 'Hello World' program for Kodu is the two-rule program "WHEN see apple DO move toward; WHEN bumped apple DO eat it." These two rules will run until every apple in the world has been eaten.

SIGCSE '17, March 8–11, 2017, Seattle, WA, USA.

DOI: http://dx.doi.org/10.1145/3017680.3017791

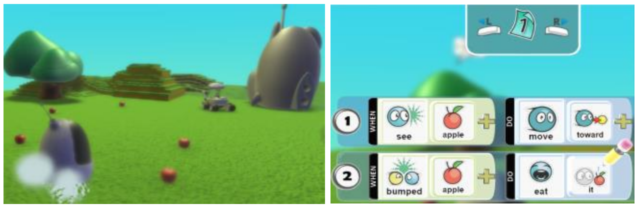

Figure 1: Kodu apple world (left) and rule editor (right).

2.2 Kodu Curriculum

A comprehensive curriculum for elementary through high school students has been developed for Kodu [16]. This curriculum aims to help students to develop mastery of lawfulness in Kodu [15]. It is a structured curriculum that is organized around six modules that teach basic programming and game design patterns (idioms), which are presented on flashcards (Figure 2). In addition, the curriculum includes the use of tile manipulatives (Figure 3) to help students learn how to construct rules prior to navigating through the expansive menu of tiles in the Kodu rule editor.

Kodu Flashcards are a tangible collection of design patterns for programming in Kodu that are laminated and bound together for quick reference and use. The front side of each flashcard provides a conceptual description of the design pattern and a graphical representation of the resulting behavior (Figure 2, left). The back side of each flashcards shows the corresponding rules using the notations that students will see in the rule editor (Figure 2, right).

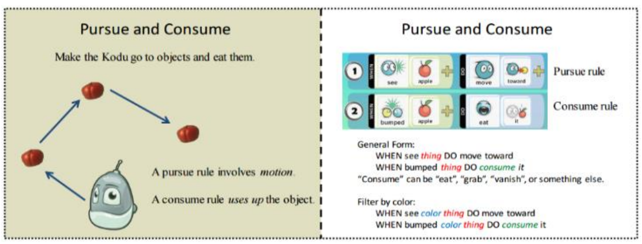

Figure 2: Flashcard showing the Pursue and Consume idiom.

Figure 2 shows the first design pattern students learn in the Kodu curriculum: Pursue and Consume (P&C). This is the 'Hello World' of Kodu described in Section 2.1. The P&C design pattern instructs a character to move toward the closest object that satisfies the rule (e.g., "WHEN see apple DO move toward"), and to consume it upon contact (e.g., "WHEN bumped apple DO eat it".) We consider the Kodu flashcards as physical manipulatives because their shape and size allow students to easily identify them on a work surface and quickly flip through them, referring back to them when needed. While the reference function of the flashcards is similar to that of quick reference guides on single laminated sheets, their compact flip-able structure allows students to quickly navigate through the focal concepts and reduces visual or conceptual distractions when trying to program.

Tiles are the second form of physical manipulative used in the curriculum. They are puzzle-shaped pieces created to model the WHEN-DO template and the graphical tiles in Kodu's rule editor (Figure 3). The WHEN part of each rule is green, and the DO part is blue. The tile set also features a special indentation tile to help students understand indentation in Kodu, which is similar to Python. The primary goal of the tiles is to help students recognize and construct syntactically correct rules. In particular, they help students understand which tiles go in which part of the WHEN-DO template.

In general, the curriculum is designed for instructors to use the tiles to model how to construct rules from the flashcards and discuss the types of objects that are found in the two halves of the WHEN-DO template. The students are then expected to practice constructing rules with the tiles and then implement them in the rule editor. Later when students are working on activities, they are expected to refer back to the tiles as a reminder of how the concepts work and how to construct the rules. In this way, the tiles and flashcards are designed to reinforce the principles of pattern recognition and rule construction in Kodu Game Lab.

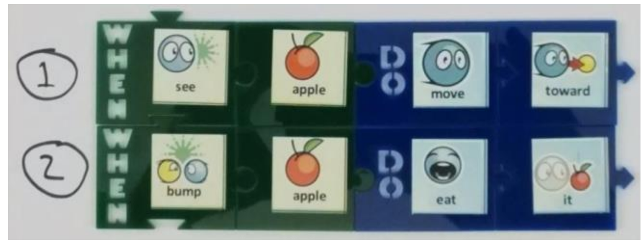

Figure 3: Tiles showing Pursue (1) and Consume (2) rules.

3. EXPERIMENT

We hypothesized that the use of physical manipulatives such as tiles and flashcards improves students' performance in understanding and recognizing design patterns and properly constructing rules in Kodu Game Lab.

Our goal for this study was to explore the effectiveness and usefulness of tiles and flashcards in the Kodu curriculum relative to paper-based alternatives. The paper-based alternatives remove the manipulative nature of these resources while keeping the curricular content intact to isolate the impact of the use of the manipulatives on students' ability to understand, recognize, construct, and use Kodu design patterns.

This intervention was designed to model the recommended use of the resources as described in Section 2.2 by recreating the usage conditions and preserving the learning from these resources. In this condition, printed versions of flashcards and WHEN-DO tile templates were provided to the students.

The work presented in this paper was a mixed methods research study. We used a between-subject study design to isolate the use of physical manipulatives versus paper-based alternatives. Our independent variable was the use or non-use of physical manipulatives. Our dependent variable was student performance on the Module 1 Assessment. We controlled for the instructional time, location, curriculum activities, and instructor. Random variables that we could not control for were the students' states of mind and prior programming experiences. The students were randomly divided into two groups and assigned the following conditions:

The Group A students (with physical manipulatives) were given the Kodu tiles and flashcards and were provided with verbal instruction on how to use them as described in the curriculum (Figure 4 left). **The Group B students (without physical manipulatives)** were given 8.5 x 11 sheets of paper with color prints of the first two design patterns that were relevant for the learning activities. They were also provided with visual representations of the plastic tiles that were printed on black and white 8.5 x 11 sheets of paper (Figure 4 right). In both groups, each student was also given a laptop and an Xbox controller for the experiment.

Figure 4: Test Conditions - Tiles and Flashcards - Group A (left) & Paper Constructs - Group B (right).

Procedure. Each group participated in two 90-minute sessions conducted after school on a Tuesday and Thursday in a given week. The goal was to complete the two curriculum modules: Module 1: Pursue and Consume and Module 2: Color Filters. Prior research suggests that students benefit most from the tiles and flashcards during these introductory modules [14]. Thus, we limited this study to the first two modules of the Kodu curriculum.

In the first session, the students were introduced to Kodu and to the first module of the curriculum, which focused on the Pursue and Consume design pattern. In the second session, they were given a refresher activity to remind them of the concepts learned in the previous session. Then they were given an assessment to evaluate their knowledge of Module 1. Finally, they were introduced to Module 2 and asked to complete an assessment for that module. All sessions were led by the first author, aided by two Teaching Assistants who helped the students as needed.

4. PARTICIPANTS

The students were recruited from the same elementary school and were randomly divided into two groups. Group A had five students: 3 third graders, 1 fourth grader, and 1 fifth grader; 2 girls and 3 boys. Group B had 6 students: 1 second grader, 2 third graders, 2 fourth graders, and 1 fifth grader; 2 girls and 4 boys. None of the students indicated any prior programming experience, except for one student in Group A who had used Minecraft.

5. DATA COLLECTION & ANALYSIS

We collected student pre- and post-surveys, student end-of-module assessments, student artifacts, and researcher field notes. The surveys and assessments were paper-based. The researcher field notes included time spent on session activities, students' overall engagement, use of tiles, flashcards, and alternatives, and interaction with Kodu.

We analyzed the data using quantitative and qualitative techniques. The students were given two assessments during the study; however, due to timing, Group A was unable to complete the Module 2 assessment before the end of the second session. Thus, we only present the results from Module 1, which had 13 questions.

These 13 questions focused on understanding, recognition, and construction of the Pursue and Consume design pattern. We analyzed the data from Module 1 in two ways. First, we used Bloom's Taxonomy [5] to measure concept understanding and skills targeted by the Kodu curriculum and assessments. This taxonomy was used because it considers all the levels of cognitive understanding of the materials covered in Module 1. For the quantitative analysis of the Module 1 assessment, each of the 13 questions were categorized by the authors according to the six levels of Bloom's Taxonomy: Remembering (1 question), Understanding (4 questions), Applying (1 question), Analyzing (3 questions), Evaluating (2 questions), and Creating (2 questions).

In the second analysis, we focused on evaluating the impact of the use of manipulatives on skill and knowledge development. In particular, we qualitatively explored questions that provided evidence of the students' ability to recognize design patterns, construct proper rules for design patterns, and demonstrate concept understanding by use of the flashcards and visualization of program execution in 3D Kodu worlds. This analysis helped provide an additional perspective on our data that was not available from Bloom's Taxonomy analysis. We found this level of analysis to aid us in understanding the conditions and affordances of manipulative use that were beneficial or non-optimal.

6. FINDINGS

The students in both groups demonstrated varying levels of mastery of Module 1 concepts. However, overall, the students answered roughly the same number of assessment questions correctly. In Group A (with physical manipulatives), 4 out of the 5 students individually answered 6 to 9 questions out of 13 correctly. The fifth student in this group answered only 2 questions correctly. This resulted in an overall total of 50% correct answers for Group A. All 6 students in Group B (without physical manipulatives) individually answered 6 to 10 questions out of 13 correctly with an overall total of 64% correct answers for the group. When looking at the levels of Bloom's Taxonomy, however, we can see variations in students' learning across groups.

Similar performance between groups: In both groups, most students answered the Remembering question correctly: 80% (4 of 5) in Group A (with manipulatives), and 83% (5 of 6) in Group B (without manipulatives). This suggests that the students did learn the Pursue and Consume design pattern whether or not they used physical manipulatives.

Slightly differing performance between groups: In this category, the differences between the groups were worth exploring. For the four Understanding questions, students in Group A (with manipulatives) produced in aggregate 8 correct responses out of 20 (40%), while students in Group B (with manipulatives) produced 15 correct responses out of 24 (63%). Similarly, for the three Analyzing questions, Group A produced 8 correct responses out of 15 (53%) while Group B produced 12 correct responses out of 18 (67%). In the Evaluating questions, Group A answered 40% (4 out of 10) correctly, and Group B answered 58% (7 out of 12) correctly. The differences between the groups indicate that Group A did not perform as well as Group B on the Understanding, Analyzing, and Evaluating questions.

Drastically differing performance between groups: We also found that Group A (with manipulatives) significantly underperformed Group B (without manipulatives) on the Applying question, but the reverse was true for Creating questions. For the Applying question, Group A produced 1 correct response out of 5 (20%), while everyone in Group B answered the question correctly (100%). For the Creating questions, Group A produced 7 correct responses out of 10 (70%) while Group B produced only 5 correct out of 12 (42%).

In summary, using Bloom's Taxonomy, we found that Group A performed much better on the Creating questions, while Group B performed much better on the Applying question and slightly better on the Understanding, Analyzing, and Evaluating questions.

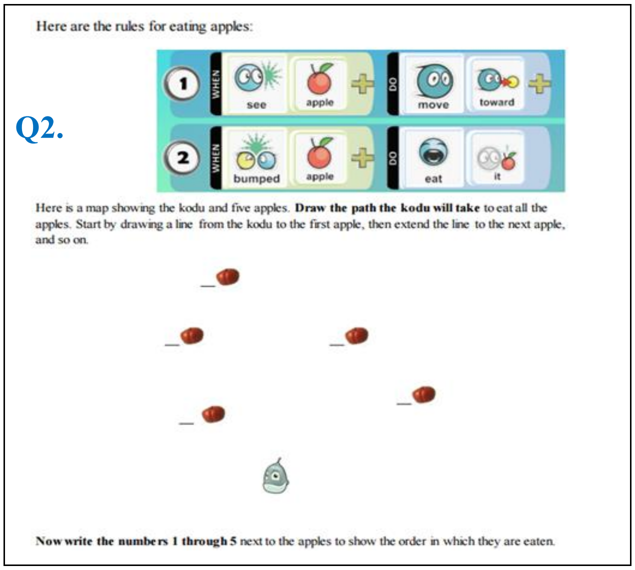

Figure 5: Q2, which was categorized in the 'Applying' category of Bloom's Taxonomy.

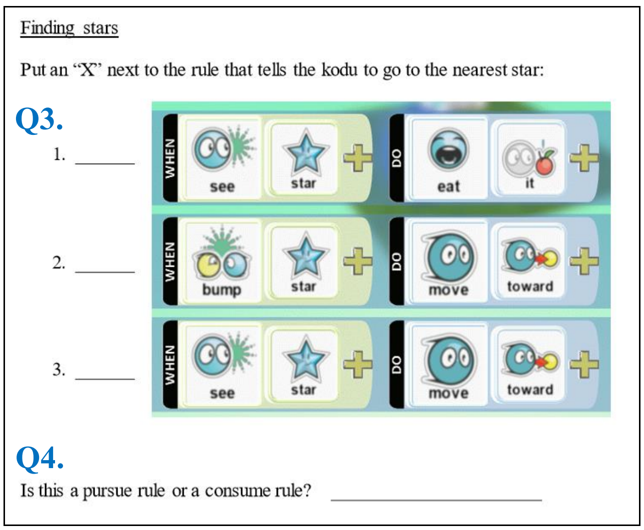

Figure 6: Q3 and Q4, which were categorized in the 'Understanding' category of Bloom's Taxonomy.

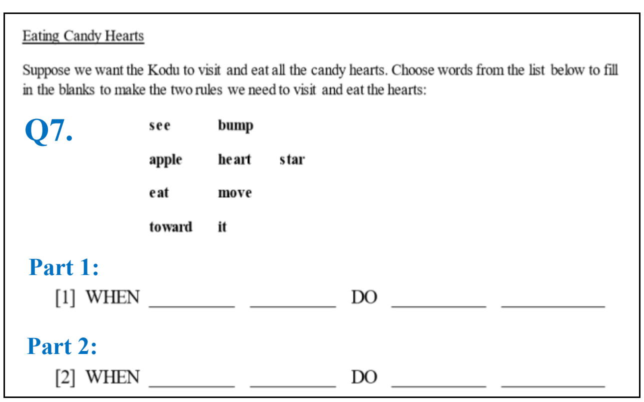

Figure 7: Q7, which was categorized in the 'Creating' category of Bloom's Taxonomy.

6.1 Pursue and Consume Understanding

To better understand the differences between these two groups, we analyzed the concepts and skills that students learn in the Kodu curriculum: recognition of design patterns, proper rule syntax construction, concept understanding, and simulation. We expected the use of physical manipulatives to have a direct impact on the students' learning of these concepts and skills.

The primary concept in Module 1 was the P&C design pattern, which instructs the Kodu character to move toward the nearest apple and eat it, then repeat that action until all of the apples are eaten. In order to test the students' understanding of this concept in the assessment, they were given three separate questions about the P&C rules to independently examine their understanding. Q3 and Q4 (see Figure 6) were based on Pursue. Q3 asked the students to select the correct rule out of the three possible rules by which the Kodu character can move toward an apple. Q4 asked the students to identify the name of the movement, choosing between 'pursue' and 'consume'. Similarly, Q5 and Q6 were based on Consume. Q5 asked the students to select the correct rule that would make the Kodu character eat an apple once it bumped into an apple. Q6 asked the students to identify the name of the action with the same three options available in Q4. Q7 asked the students to write the rules for a Kodu character to Pursue and Consume candy hearts using a fill-in-the-blank format (Figure 7). The students were given a word bank of Kodu tile labels to fill into the respective WHEN-DO rules. The first statement tested students' Pursue understanding, and the second tested their Consume understanding.

The results from Q3 through Q7 revealed that the two groups were similar in concept understanding: 50% of Group A (with manipulatives) responses, and 55% of Group B (without manipulatives) responses were correct. This suggests that the students gained similar P&C design pattern understanding, regardless of the use of physical manipulatives.

6.2 Proper Rule Recognition and Construction

One of the expectations for the students' use of tile manipulatives in the curriculum is for them to be able to recognize and construct design pattern rules correctly. While the students in this study gained similar levels of understanding of the P&C design pattern, an analysis of the questions that were focused on proper rule construction revealed differences between the groups. We used Q3 and Q5 to measure the students' ability to recognize proper syntax of the P&C rules, since the students' ability to construct proper rules was influenced by their use of tiles (with manipulatives) and the rule editor (with and without manipulatives). The questions related to recognizing proper rule syntax also represent a subset of questions from the Understanding level of our Bloom's analysis. We used Q7 Parts 1 and 2 (Figure 7) to measure the students' ability to construct P&C rules. These questions were categorized as Creating questions in Bloom's analysis.

While both groups were able to recognize rules equally (50% each), they differed in rule construction. The Group A (with manipulatives) students produced 70% correct responses, while the Group B (without manipulatives) students produced 42% correct responses (Table 1). This suggests that Group A performed better than Group B on rule construction, which we believe was caused by the Group A students' use of tiles.

Table 1. Analysis of Proper Rule Syntax

Skill	Question Number	Group A (n=5)	Group B (n=6)
Recognition (Tiles & Flashcards)	Q3	2/5	2/6
	Q5	3/5	4/6
Overall Recognition	**Q3 & Q5**	**50%**	**50%**
Construction (Tiles)	Q7 Part 1	4/5	3/6
	Q7 Part 2	3/5	2/6
Overall Construction	**Q7 Parts 1 & 2**	**70%**	**42%**

6.3 Concept Understanding with Flashcards

The Kodu curriculum was designed for students to use flashcards in order to learn design patterns and refer back to them while completing learning activities. We observed that the Group A students actively used the flashcards during the activities. For this analysis, we selected questions that evaluated students on their conceptual understanding of Pursue and Consume (P&C) as described on the flashcards and paper-based equivalent. In this way, the selected assessment questions allowed us to measure the impact of flashcard use on concept understanding.

Q1 tested the students' ability to recognize the direction in which the Kodu character would move based on P&C rules provided. Q4 (Figure 6) and Q6 (not shown due to space), by contrast, tested the students' ability to correctly identify the name of each rule associated with Kodu character's action as described in the question.

Table 2. Concept Understanding Based on Flashcards

Skill	Question Number	Group A (n=5)	Group B (n=6)
Recognizing Pursue and Consume	Q1	4/5	5/6
	Q4	1/5	5/6
	Q6	2/5	4/6
Overall P&C Recognition	**Q1, Q4, Q6**	**46%**	**77%**

Group B (without manipulatives) performed better than Group A (with manipulatives) on these three recognition of Pursue and Consume questions. The Group A students correctly answered 46% of the questions, while the Group B students correctly answered 77% (see Table 2). This suggests that interaction with Kodu's 3D worlds may have helped the Group B students to develop their ability to recognize the actions associated with the P&C concepts.

6.4 Simulation

Another expectation of the Kodu curriculum is to develop students' ability to mentally simulate and predict program behavior [15]. In this study, mental simulation of Kodu rules was gauged using two types of questions. These assessed the students' ability to mentally simulate (1) the basic P&C design pattern, which required only knowledge of the pursue and consume rules; and (2) intermediate P&C programs using multiple pursue or consume rules.

On the basic simulation questions, we found that the Group A (with manipulatives) students correctly answered 50% of the questions, while the Group B (without manipulatives) students correctly answered 91% of questions. On the intermediate simulation questions, Group A students correctly answered 43% of the questions, and Group B students correctly answered 69% of the questions (see Table 3). This suggests that the development of mental simulation abilities requires more interaction with the rule editor and the 3D visualizations of the rules, as experienced by Group B.

Table 3. Analysis of Simulation

Skill	Question Number	Group A (n=5)	Group B (n=6)
Basic P&C Sim. Knowledge	Q2, Q8	5/10	11/12
Intermediate P&C Sim. Knowledge	Q9-12	8/20	14/24
Overall	**Q2, Q8-12**	**43%**	**69%**

7. OBSERVATION

We hypothesized that the use of physical manipulatives such as tiles and flashcards improves student performance. However, we found that while students who used manipulatives did better on rule construction, those who did not use the manipulatives did better on simulation and overall understanding of the concepts.

We turned to our observation and field notes to better understand the differences between the two learning methods in the classroom.

The students of Group A (with manipulatives) extensively used tiles before every Kodu activity that they were asked to complete. They constructed rules using the tiles before constructing them in the rule editor. Some of the students referred back to the flashcards while completing other learning activities, primarily focusing on the back side of the flashcards, which had the Pursue and Consume rule syntax. Whenever the students were in doubt, they checked the syntax of the tiles using the flashcards and also consulted the instructors before finally putting the rules into the rule editor. Discussions between the instructors and the students were often lengthy, as the instructors helped the students to understand why their rules were not logical, or were inconsistent with the type of rule construction that was needed to complete the activity.

The students of Group B (without manipulatives) initially used the paper constructs of tiles and flashcards, but they were reluctant to continue using them beyond the initial group activity. After interacting with the rule editor, the paper-printed tile rules were not used by the students; rather, the students used the Kodu rule editor to directly construct the rules. Students would make their solutions for the activity in the rule editor, run their worlds and make observations, then iteratively change their solutions until they accomplished the required task. After a couple of iterations, the students often called over the instructors without referring back to any material available to them (i.e., the paper-based alternative to flashcards and tiles). But while the students were explaining their problems to the instructors, they often identified their own issues before an instructor could assist them.

8. DISCUSSION & IMPLICATIONS

The results of this study indicate both the benefits and the drawbacks of using physical manipulatives in different learning situations. The students who used physical manipulatives were better at rule construction than the students who did not use physical manipulatives. This might have been because the students who used tiles before completing the activities in the rule editor developed a more refined understanding of the proper rule syntax of Kodu design patterns. We believe that the students who did not have manipulatives focused more on completing the activities iteratively, as they received more dynamic feedback from the programming environment. In contrast, the students with manipulatives focused on constructing the syntax with tiles without getting the same dynamic feedback that the students without manipulatives received. Thus, this difference in feedback altered the direction of focus for the two groups while they completed the learning task. Group B was primarily focused on completing the activity, and Group A was focused on constructing rules for that activity using the tiles and flashcards. This suggests that extended use of tiles may have diminishing returns, as it can be time-consuming to construct the correct syntax without dynamic feedback from the programming environment. These findings also suggest that the Group A students' intense focus on constructing rules also limited their usage of the flashcards to understand the general concept of Pursue and Consume. Therefore, we recommend that the use of tiles should be limited to introducing students to proper rule syntax and construction and to explain more complex syntax configuration, such as indentation.

The students who did not use tiles (Group B) may have acquired a more nuanced understanding of how the rules are executed through completing the activities through trial and error interactions in the programming environment. We believe that the reinforcement provided by dynamic feedback and the visualization of rule execution in the programming environment helped these students to develop the ability to mentally simulate rule execution. The students with manipulatives had limited interaction with the rule editor due to time constraints, resulting in (1) limited exposure to dynamic visualization of rule execution and (2) limited recognition of behavior caused by errors in syntax.

Thus, the findings of this study suggest that iterative development in the programming environment helps students to observe and visualize Kodu's dynamic behavior, which impacts students' ability to simulate and predict Kodu's behavior on paper-based assessments.

9. LIMITATIONS

This study has the following limitations: **Number of Students:** Both Groups A and Group B had a small number of students because of the qualitative nature of the evaluation and other logistical constraints. This limits our ability to generalize these results over a large population using similar materials. **Length of Study:** The study had only two 90-minute sessions with each group. This provided students with a relatively short amount of time to learn and explore these concepts. In addition, the results presented in this study were based on only the first module of the curriculum. For more nuanced results, longer instructional time is needed to allow students to practice and learn the concepts and to see if these results can be replicated.

10. CONCLUSION

This study has implications for researchers and practitioners who are working on developing K-12 CS educational curricula and resources. Our results show that students make active use of flashcards when they are learning new concepts. Our results also show that selective and strategic use of physical manipulatives such as tiles can foster the development of rule construction. However, if the use of these manipulatives is not monitored, students can spend their time unproductively at the expense of reinforcing their conceptual understanding, which is fostered more deeply through students' iterative interaction with the rule editor.

Acknowledgments

Funded by a gift from Microsoft Research. Thanks to Kyuseo Park and Jiyoung Kang, who served as TAs in the Kodu camps. Thanks to Dr. Fred Ball for help in organizing the camps.

11. REFERENCES

[1] Bhagi, A. (n.d.). App Inventor Concept Cards. Retrieved August 26, 2016, from http://appinventor.mit.edu/explore/sites/all/files/ConceptCards/ai2/AI2_ConceptCards.pdf

[2] CS Unplugged; http://csunplugged.org. Accessed 2016 June 21.

[3] Dann, W. P., Cooper, S., & Pausch, R. (2011). *Learning to Program with Alice (w/CD ROM)*. Prentice Hall Press.

[4] Enome, Inc. (n.d.). Literacy Manipulatives. Retrieved August 26, 2016, from https://goalbookapp.com/toolkit/strategy/literacy-manipulatives

[5] Krathwohl, D. R. (2002). A revision of Bloom's taxonomy: An overview. Theory into practice, 41(4), 212-218.

[6] Learning Resources, "Research on the benefits of manipulatives;" http://www.learningresources.com/text/pdf/Mathresearch.pdfAccessed 2016 June 23.

[7] Lifelong Kindergarten Group (MIT), "Blocks;" http://wiki.scratch.mit.edu/wiki/Blocks. Accessed 2016 June 18.

[8] Moore, S. D. (n.d.). Why Teach Mathematics with Manipulatives? Retrieved August 26, 2016, from ETA hand2mind; http://www.hand2mind.com/pdf/Benefits_of_Manipulatives.pdf. Accessed 2016 June 18.

[9] Repenning, A., Webb, D., & Ioannidou, A. (2010, March). Scalable game design and the development of a checklist for getting computational thinking into public schools. In *Proceedings of the 41st ACM technical symposium on Computer science education* (pp. 265-269). ACM.

[10] Resnick, M., et al. "Scratch: Programming for everyone." Communications of the ACM 52, 11 (2009): 60-67.

[11] Rusk, N. "Scratch cards;" http://scratched.gse.harvard.edu/resources/scratch-cards-0. Accessed 2016 July 21

[12] Stolee, K. T., & Fristoe, T. (2011, March). Expressing computer science concepts through Kodu game lab. In *Proceedings of the 42nd ACM technical symposium on Computer science education* (pp. 99-104).

[13] Touretzky, D. S., Marghitu, D., Ludi, S., Bernstein, D., & Ni, L. (2013, March). Accelerating K-12 computational thinking using scaffolding, staging, and abstraction. In *Proceeding of the 44th ACM technical symposium on Computer science education* (pp. 609-614). ACM.

[14] Touretzky, D. S. (2014). Teaching Kodu with physical manipulatives. *ACM Inroads*, *5*(4), 44-51.

[15] Touretzky, D. S., Gardner-McCune, C., & Aggarwal, A. (2016, February). Teaching Lawfulness With Kodu. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education* (pp. 621-626). ACM.

[16] Touretzky, D. S. (n.d.). Kodu Curriculum Modules. Retrieved July 26, 2016, from https://www.cs.cmu.edu/~dst/Kodu/Curriculum/

[17] Touretzky, D. S. (n.d.). Kodu Idiom Flash Cards & Tiles. Retrieved July 26, 2016, from https://www.cs.cmu.edu/~dst/Kodu/