Teaching and Learning through Creating Games in ScratchJr

Who needs variables anyway!

Aye Thuzar
Computer Science & Mathematics
The Pingry School
Basking Ridge, NJ, United States
athuzar@pingry.org

Aung Nay
Zatna LLC
Martinsville, NJ, United States
anay@zatna.com

***Abstract*— This paper presents an idea for teaching and learning through creating games in ScratchJr, which is an extension of the existing curriculum available for ScratchJr. It discusses our experience of running half-day camps with K-2nd graders using ScratchJr and how essential skills such as designing and planning, sequencing, problem solving, thinking about thinking (metacognition), and sharing can be developed through creating games in ScratchJr. Since ScratchJr does not have variables but rather events and messaging, we used a sprite (an object that performs actions) moving towards a goal to represent positive integers and away from that goal to represent negative integers. It is the same concept as using a number line. The sprite is at zero at the start of the game, and as the game progresses, the sprite moves either to the left or right goal triggering the winning or losing page. This way of scoring tracking is highly visual, and younger students can understand and incorporate it into their games even without understanding addition, subtraction, or negative integers properly. With this idea of tracking progress, we can create a variety of games that are very entertaining to play while still being understandable to younger students who will be able to code on their own.**

Keywords— ScratchJr; K-2; Games; No Variables; Progress Bar; Number Line

I. Introduction

We have been teaching Computer Science summer camps since 2009 and for the summer of 2015, we wanted to target the younger K-2 age group using ScratchJr in our half-day week-long camps. After reviewing the ScratchJr curricula that were available on scratchjr.org, we came to the conclusion that we would not be able to maintain the students' excitement just using the storytelling and animation elements for a week. We needed to offer something more. The natural inclination was to include games. However, creating games in ScratchJr was a challenge since ScratchJr does not have variables. Because of this, we developed the idea of a sprite moving towards a goal as an ongoing visual tracker/indicator of the player's progress, which allowed us to create a variety of games. The game creation allowed us to move beyond the interactive storytelling phase and maintain excitement throughout the week.

II. Background

ScratchJr is a graphical programming language that allows children from age five to seven to create "interactive and animated scenes and stories" [1]. It addresses the lack of programming tools that focus on "content creating or higher level thinking" [1] for kindergarten to second grade students. ScratchJr software deployment comes with the curriculum and online community, and the design goal of the ScratchJr software is to "provide young children with a powerful new educational tool as well as guidance for teachers and parents to implement it to the benefit of diverse areas of early learning, from math and literacy to interdisciplinary knowledge structures" [1].

III. Curriculum Deployment

Our ScratchJr camp curriculum is designed in such a way that a concept is revisited throughout the duration of the camp with additional concepts being added to the sum of knowledge as the camp progresses. Our planned learning experience starts with animated scenes and stories, using motion, looks, and sound blocks, and progresses to interactive animations, followed by game creation, using all block categories. Approximately 60% of the camp time was spent on creating games. We used iPad Mini2s with ScratchJr in our camps. Each student and instructor had an iPad. The instructor to student ratio was kept to 1:5 or lower. The camp day consists of modeling, project planning and creation, and social interaction (feedback and assessment). The camp week culminates in a final project, which is also an assessment.

A. Modeling

At the beginning of each day for each project, students see a model or example of what they are supposed to achieve for the day as well as for the individual phases that help lead the students to that day's overall outcome(s). For example, prior to a project, we might talk briefly about which concept(s) they need to learn and we might show them a sample project or two of what could be achieved with this new concept(s), along with what they already know. To introduce new concepts or to review concepts, we might use unplugged activities, such as coming up with a sequence of commands for someone to

978-1-4673-8367-7/15/$31.00 ©2015 IEEE

perform a particular task in real life, acting out a given sequence of commands or playing a card game with ScratchJr block images.

Fig. 1. Progress bar as a score tracker. (*Wizard & Tac*)

Fig 1 shows a sample project we modeled on the second day of camp. The game has the progress dot that starts at the middle and has the red vertical bars on the left and right sides of the screen. As time progresses, the red dot moves to the left periodically; however, if the fireball hits Tac, the little creature, the red dot will move to the right. Winning or losing is defined by whether the red dot reaches either the right or the left red bar. Here, we introduced the idea of the number lines, as well as positive and negative integers with this visual progress bar.

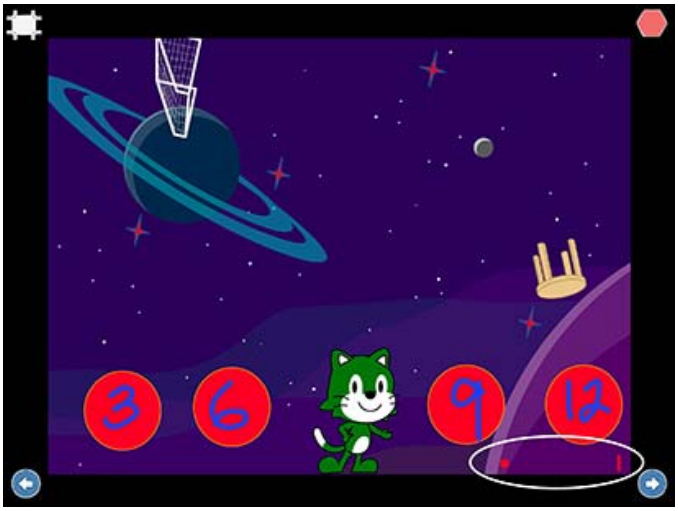

Fig. 2. Example of a progression tracker. (*Space Jump*)

Fig 2 shows a sample project we demonstrated on the fourth day of camp. Progression through the game is tracked by the red dot sprite moving towards the vertical red bar at the bottom right part of the screen that is highlighted by the white oval. As the game progresses, the red dot moves towards the red bar. When the red dot reaches the red bar, the player is greeted by a "You Win" scene.

Along with the achievement tracking, this game also tells a story about the dangers of space junk, and collecting energy stars. The storytelling aspect of the games always draws students in, motivates, and inspires them to create their own games with storyline. The integers that you see on the screen are buttons that allow the Martian Scratch Cat to jump different heights to collect the stars.

Our curriculum for ScratchJr also features the concept of "learn to code and code to learn" [4]. For example, there is always a math lesson to teach or review with every game we introduce to the students, either about numbers, total time frame, or the number line. In the game above, we had conversations about space junk and how to be responsible in space as well as why and how stereotypical Martians are green.

B. Project Planning & Creation

We make planning a regular part of the students' learning process. Before every project, students plan out different elements of it. They are required to draw out their plans and describe their plan to the instructors either in writing or orally. Below is an example of a student's final project planning sheets. This activity encourages them to think about their own thinking process(es).

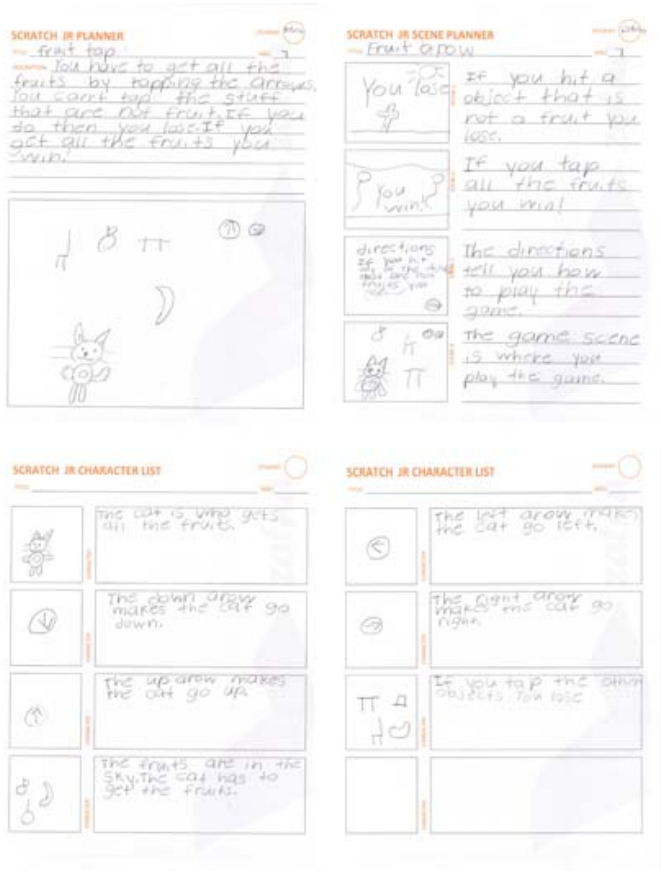

Fig. 3. Student planning sample. (*Fruit Tap*)

The students come up with a general concept, followed by a breakdown of that concept into different scenes, a list of characters and their actions. This planning process asks the students to think about the instructions/commands they need to

give to each of the characters in the game and how those characters will come together to form one cohesive story or game.

One of the biggest tasks in the planning process is the discussion that takes place between the individual student and the instructor with regards to the vision that the student has. This help students clarify their thinking processes. With the approval of an instructor, students can move on to iPads to make their plans come to life on screen.

C. Social Interaction

Our intent for social interaction among our students is three-fold: interaction at camp, interaction online and interaction at home. Interaction at camp is done through "code and tell" [2]. And the way we assess the "tell" part uses the three "levels of describing, demonstrating and imagining projects" [2]: simple, intermediate and complex. This YouTube link is an example of a student describing her game, including the story and her planning process: https://youtu.be/EFbo76jE_sk. When students complete their projects, they should spend a few minutes sharing their unique projects and stories with their campmates and instructors. The students should also give constructive criticism of each other's projects. This promotes cross-pollination of ideas between projects. Online interaction takes place at the end of the day when students vlog (video blog) which is intended as a conversation starter for friends and/or family members. Interaction at home occurs in a few ways: explaining the details of what the students did for the day to their parents (since parents already have prompts from the vlogs); teaching other siblings the great new things that students have learned at camp; and showing-off/bragging about one's achievements to friends who do not attend camp.

D. Final Project

Fridays are open project days. Students are given certain criteria such as creating a game with a storyline, having sound, and using at least four sprites etc. After the planning and creation phase for the final project, their projects are then shared with their peers and instructors at the end of the day. The final projects are assessed based on students' plan for the final projects. The instructors evaluate whether a student was able to create the project that s/he has envisioned and planned.

IV. Reflection

Having a relatively smaller instruction set and needing only a few blocks to create highly entertaining games worked out well with younger students. Individuals who had experiences in other drag and drop environments wished there were variables; however, we believed that we could do without variables, especially for younger students. In our opinion, ScratchJr is a very flexible tool with relatively "low floors" [3], low barrier to entry, and "wide walls" [3]. It supports a broad range of activities, including games. We can teach younger students how to express themselves digitally and be excited about their creations, a task not easily achievable prior to ScratchJr.

Additional benefit of creating games in ScratchJr is giving students "a new context for learning" [4]. Depending on their age, some were able to revisit and apply mathematical concepts such as positive and negative integers that they had already learned. For some younger students, it became a great way to get introduced to those concepts.

Some students feel restricted by the limitation on number of messages that could be sent and number of scenes they could use and wished there were more messages and scenes. We also experienced increased delays in our games with the increased number of sprites, regardless of the iPad model we were using.

Outside of the daily and sustained camp experience, the ephemeral use of ScratchJr might preclude the students' memory of integral aspects of the program which might require the teacher to re-expose the students to components that they have already seen. In the camp setting, this does not apply as often due to daily exposure to and use of the program.

V. Conclusion

We found that creating games in ScratchJr is a great next step for campers after they made their interactive and animated scenes and stories. Creating games with storylines ignited and maintained the interest of K-2 students throughout our half-day week-long camp. Our progress bar idea opened up opportunities to create different types of games that kept students' interest piqued. We were able to keep students excited throughout the week with more variety than what was possible with the ScratchJr cards and existing ScratchJr curricula at scratchjr.org. In addition, students were also exposed to a new context for learning other topics. Most importantly, we were able to create an environment where younger students could explore and learn about designing and planning, sequencing, problem solving, thinking about thinking, and sharing.

References

[1] Flannery, Louise P., Elizabeth R. Kazakoff, Paula Bontá, Brian Silverman, Marina Umaschi Bers, and Mitchel Resnick. "Designing ScratchJr: Support for Early Childhood Learning through Computer Programming." DevTech Research Group. Proc. of 12th International Conference on Interaction Design and Children, ACM, New York, NY. Tufts University, n.d. Web. 29 Aug. 2015

[2] Portelance, Dylan J. "Code and Tell: An Exploration of Peer Interviews and Computational Thinking With ScratchJr in the Early Childhood Classroom." Thesis. Medford/Somerville, MA/Tufts University, 2014. Code and Tell: An Exploration of Peer Interviews and Computational Thinking With ScratchJr in the Early Childhood Classroom. Tufts University, 2015. Web. 26 May 2015.

[3] Resnick, Mitchel et al. "Scratch: Programming for All." Communications of the ACM 52.11 (2009): 60-67. MIT Media Lab. Massachusetts Institute of Technology. Web. 21 July 2015.

[4] Resnick, Mitchel. "Learn to Code, Code to Learn (EdSurge News)." EdSurge. EdSurge, 08 May 2013. Web. 26 May 2015.

[5] Strawhacker, Amanda, Melissa Lee, Claire Caine, and Marina Bers. "ScratchJr Demo: A Coding Language for Kindergarten." DevTech Research Group. Proc. of 14th International Conference on Interaction Design and Children, ACM, Boston, MA. Tufts University, n.d. Web. 29 Aug. 2015.