Scratching the Surface of Video Games

Developed by Sarah Kramer
@kramerwithak | kramerwithak.weebly.com

Grade Level: 4th (could be adapted for 3rd or 5th)

CS Standards:

- **1B-AP-09:** Create programs that use variables to store and modify data.
- **1B-AP-10:** Create programs that include sequences, events, loops, and conditionals.
- **1B-AP-13:** Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.
- **1B-AP-15:** Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.
- **1B-AP-17:** Describe choices made during program development using code comments, presentations, and demonstrations.
- **1B-IC-18:** Discuss computing technologies that have changed the world, and express how those technologies influence, and are influenced by, cultural practice.

Additional Content Area Standards:

This long division standard is related to the in unplugged lesson surrounding algorithms.

- **CCSS.MATH.CONTENT.4.NBT.B.6:** Find whole-number quotients and remainders with up to four-digit dividends and one-digit divisors, using strategies based on place value, the properties of operations, and/or the relationship between multiplication and division. Illustrate and explain the calculation by using equations, rectangular arrays, and/or area models.

Language/Platform Restrictions:

- Students will be restricted to Scratch. My students have zero to very minimal experience with computer programming and using coding platforms. The restriction of choice is so that I can best scaffold their understanding, and hopefully as they progress, we can expand our platform options. For an introduction, I want to be able to model and expect them to use the same program. This is also supported by the research article we read in this course where block-based coding provided students with a powerful introduction to coding concepts and encouraged them to want to continue coding.

Addressing Culturally Relevant Propositions

- *Students must experience academic success:* Students will have a tutorial and be introduced to block-based coding concepts to make CS more accessible for beginners. They will have freedom to make choices, but also the tools they need to make their code work properly.
- *Students must develop and/or maintain cultural competence:* Students will communicate with one another as they share ideas through their coding project. They will also be part of the larger Scratch community of coders and have the chance to interact there. They will learn how to comment, remix, give attribution, etc in order to participate in the community successfully.
- *Students must develop a critical consciousness through which they challenge the status quo of the current social order:* Our ELA unit will tie in the social implication of video games. We will read articles advocating for and condemning student / child access to video games, screen time, etc. Students will have critical conversations about how video games affect the world around them, as well as their own lives. Finally, students will write a letter to their caregivers suggesting how screen time and video games be limited / monitored in their homes.

Plugged in Overview

Standards addressed in the full project, but not in this lesson:

- **1B-AP-13:** Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.
    - Students will interview potential users (other classmates, friends, family members, younger students) to design a game that is fun, engaging, and could be educational for them.
- **1B-AP-15:** Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.
    - Will be included during project workshop time as needed.
- **1B-AP-17:** Describe choices made during program development using code comments, presentations, and demonstrations.
    - Students will demonstrate their game and explain their project. They will also respond to reflection questions upon submission.
- **1B-IC-18:** Discuss computing technologies that have changed the world, and express how those technologies influence, and are influenced by, cultural practice.
    - Students will start the project by exploring what coding is and the history of how video games changed the entertainment world.

Students will create a video game as a "Check Your Learning" for their Coding Module in this Video Games HERO project.

After learning about sprites, backgrounds, sequences, events, loops, conditionals, variables, sensing blocks, and broadcast blocks in Scratch, students will have two options to create their own video game.

MILD	**SPICY**
Follow the Scratch Pong tutorial to create your own Pong game. **After you finish adding all the elements of the tutorial, add at least three more of your own ideas. Here are some to get you thinking:** • A "good game" message when they reach a certain score • Changing backdrops • Sounds when the ball hits the paddle • A countdown to start the game • Multiple balls bouncing • Changing colors / costumes	Create your own video game in Scratch, from scratch! **Your game must include:** • At least 2 sprites • At least 1 backdrop • Key or mouse controls **Your game must include at least three of the following:** • At least 1 loop • At least 1 conditional (if / then) • At least 1 sensing block • At least 1 broadcast block • At least 1 variable **Optional extras:** • Changing backdrops/costumes • Playing sounds • Anything else you can imagine!

Plugged in Student Examples

Mild Example - Beefed up Pong tutorial project

Unplugged Overviews

This unplugged lesson will be included in the math module (long division unit) of the Video Games HERO project. There is a step devoted to learning what an algorithm is as students explore the long division algorithm (divide, multiply, subtract, check, bring down/over, repeat/remainder).

This unplugged lesson will teach students that algorithms are sequences of instructions with a goal of one correct result. Students will realize that any "bugs" (errors) will lead to the wrong / undesired result. Students will realize the importance of taking algorithms one step at a time, and making sure to take the correct steps in the correct order.

PARTNER CODING OPTION

Materials:

- Blindfold per pair of students
- 10-20 pieces of construction paper per pair of students

Instructions:

- Students work in pairs. One is the user and is blindfolded while the other is the computer.
- The "computer" should lay out pieces of construction paper to make a maze for the "user." The more turns, the more fun! Turns should only be 90 degrees (right / left turns). The "user" should NOT watch this part so that the maze is a surprise.
- The "user" starts at the designated starting point of the maze with the blindfold on. The "computer" may not touch the user.
- The "computer" gives instructions to the user to make it through the maze. They may tell the "user" to step forward, turn right, or turn left.
- If the "user" steps off the maze, the computer says, "ERROR!" and the "user" must stop, take off the blindfold, return to the start, and try again.
- If the "user" makes it to the end successfully, they may take off their blindfold and high five the "computer." They succeeded!

MOB CODING OPTION

Materials:

- Blindfold
- Butcher / roll paper

Instructions:

- Choose one student to be the "user" and be blindfolded to complete this activity. The rest of the class is the "computer" and will mob code (one student giving an instruction at a time).
- Have the class roll out and cut butcher paper to make a maze for the "user." The more turns, the more fun! Turns should only be 90 degrees (right / left turns). The "user" should NOT watch this part so that the maze is a surprise.
- The "user" starts at the designated starting point of the maze with the blindfold on. The "computer" (rest of the class) may not touch the user.
- The "computer" gives instructions to the user to make it through the maze. Students may raise their hands to be called on to give instructions, or go in order so each student takes a turn.) They may tell the "user" to step forward, turn right, or turn left.
- If the "user" steps off the maze, the computer says, "ERROR!" and the "user" must stop, take off the blindfold, return to the start, and try again.

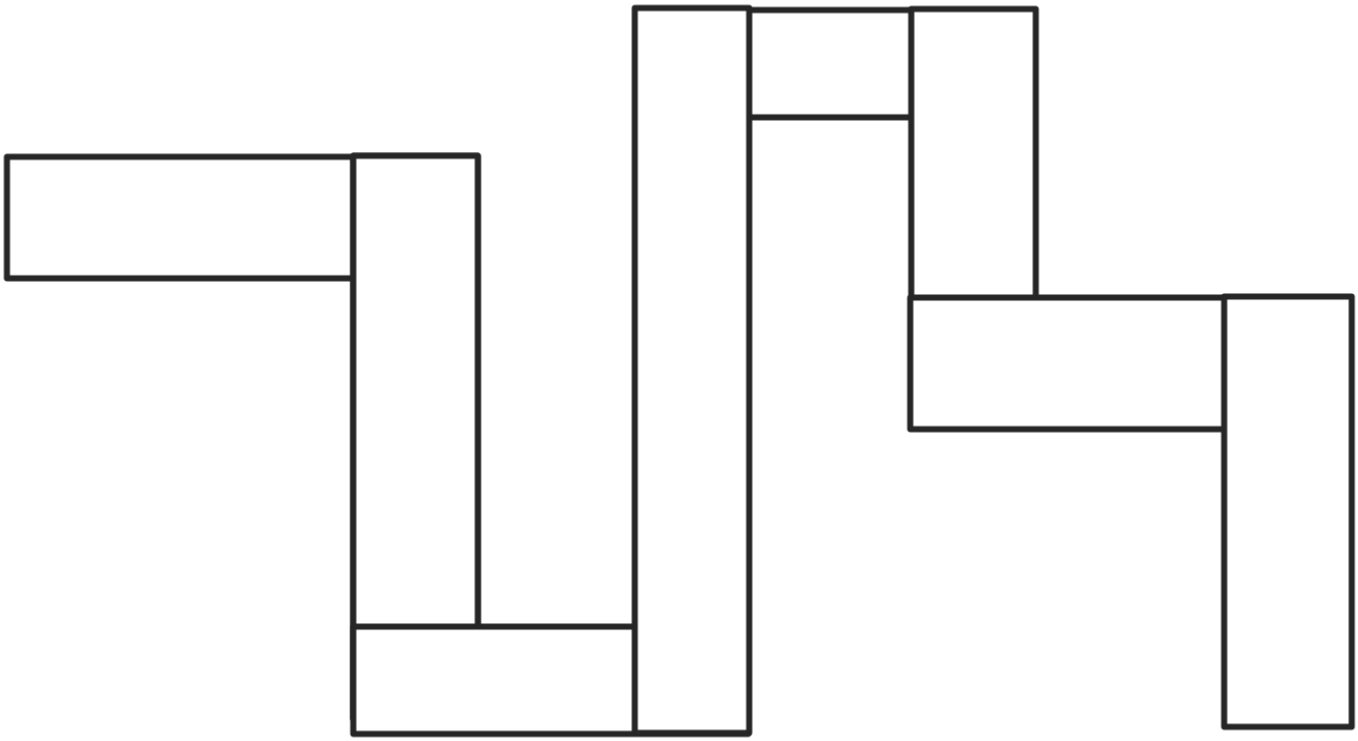
- If the "user" makes it to the end successfully, they may take off their blindfold. They succeeded!

Reflection Questions:

- Were you the computer or user? How did it feel?
- What information were you getting or giving?
- What information were you missing?
- What was most interesting about this activity?
- What was most frustrating?
- What happened to the user if the instructions weren't correct?
- What is an algorithm? What did you learn about algorithms in this activity?

Unplugged Student Example

Example Maze:



Sample Student Responses to Reflection Questions:

- Were you the computer or user? How did it feel?
    - As the "computer," I felt frustrated when the "user" wasn't listening to my instructions. I also felt like I had to give the instructions very carefully. It felt like a big job to guide the user to the end correctly.

    - As the "user," I felt nervous about if the "computer" would give me the right instructions. I had to listen very carefully to make sure I did each step correctly. I had to trust them a lot.
- What information were you getting or giving?
    - As the "computer," I was giving the instructions to the user. I told them when to move forward, left, and right to make it through the maze.
    - As the "user," I was getting the instructions from the computer. I had to listen to what they said and do it one step at a time.
- What information were you missing?
    - As the "computer," I couldn't actually move! I had to wait for the "user" to do that.
    - As the "user," I couldn't see the maze. I had to listen to the "computer."
- What was most interesting about this activity?
    - It was interesting to see how important the right order for the instructions was. It was also interesting how we had to slow down to really focus and get it right.
- What was most frustrating?
    - It was frustrating that I couldn't do both parts! We had to work together and each do an important half of the task.
- What happened to the user if the instructions weren't correct?
    - The "user" would get it wrong and have to restart. That made it really important that the "computer" go step by step and not skip anything. The "user" had to listen to each step.
- What is an algorithm? What did you learn about algorithms in this activity?
    - An algorithm is a list of steps or instructions. If followed correctly each time, it will always lead to the right answer. The "computer's" instructions to the "user" was an algorithm. That's why they had to be so careful to be correct! We didn't want the algorithm to be wrong because then the "user" wouldn't make it to the end.