

**Lesson created by the GMU-ODU CSforAll Team. For more information about this lesson and our CSforAll initiative, contact Dr. Amy Hutchison at** [achutchison1@ua.edu](mailto:achutchison1@ua.edu)

|  |
| --- |
| **Lesson 4: Planning & Coding an Animation (45 min)**  *Grades 5-6* |

|  |  |
| --- | --- |
| **Concept:** | |
| **Vocabulary:**   * Bugs * Debugging | |
| **Summary:**  In this lesson, students will plan and code an animation in Scratch with the help of the CoCo graphic organizer. They will be introduced to the concept of debugging and new sound blocks in Scratch, as well as discuss what makes for a good animation of their writing. | |
| **Lesson Objectives (learning targets): I can…**   * Review familiar Scratch blocks with a scavenger hunt * Identify new Scratch sound blocks: play sound, record/upload a sound, change the pitch effect of a sound, stop all sounds * Practice using Scratch sound blocks * Define code, bugs, and debugging * Understand how to debug an algorithm * Discuss and understand the features of a good animation * Finish animating my recipe in Scratch * Evaluate my work * Share my animation with a partner | |
| **Content Standard(s)** | **Computer Science Standard(s)** |
| The student will write in a variety of forms to include narrative, descriptive, opinion, and expository.  a) Engage in writing as a process.  b) Identify audience and purpose.  c) Use a variety of prewriting strategies.  d) Use organizational strategies to structure  writing according to type.  e) Use transition words to vary sentence structure. | The student will construct sets of step-by-step instructions (algorithms) both independently and collaboratively using sequencing and using loops.  The student will construct programs to accomplish a task as a means of creative expression using a block or text based programming language, both independently and collaboratively using sequencing and using loops.  The student will analyze, correct, and improve (debug) an algorithm that includes sequencing, events, loops and variables. |

|  |
| --- |
| **Materials** |
| **Lesson materials:**   * Chromebook/Laptop * Internet Access * Teacher slide deck * [Student slides](https://www.dropbox.com/scl/fi/xikraqtwsrwqkkpawvm9z/Student_Lesson-4-slides-Grades5-6.pptx?rlkey=8dl1jhonikest4ykvx2szxd6j&dl=0) * [CoCo Link](https://wego.gmu.edu/scratchgo/login.php) * Written recipe from last lesson * Blank paper for brainstorming   **Supplemental resources:** |

|  |
| --- |
| **Lesson Structure and Activities** |
| **(5 min) Warm-up & Introduction:**  **NOTE: All slides for this lesson are scripted so that, if needed, you can see exact definitions and instructions for teaching this lesson in the notes at the bottom of the teacher slide deck.**   * Optional: Introduce lesson, learning goals, and resources (Slides 2-5) * Share out the link to the [student slides](https://www.dropbox.com/scl/fi/xikraqtwsrwqkkpawvm9z/Student_Lesson-4-slides-Grades5-6.pptx?rlkey=8dl1jhonikest4ykvx2szxd6j&dl=0) with your students. Instruct them to open Scratch for reference and answer the questions on the slides (slides 5-7) * Share the correct answers (slide 8) |
| **(20 min) Direct Instruction & Guided Practice:**  Introduce Scratch sound blocks   * Introduce students to the following sound blocks: play sound; choose, upload, or record a sound; change the pitch effect of the sound; stop all sounds (slides 10-15) * Pause and instruct students to practice using new blocks by choosing a sprite, changing its sound, and voice-recording the sprite to say at least one line of the recipe from last lesson (slides 16-17)   Introduce Debugging   * Introduce and define concept of debugging in Computer Science (slides 19-23) * Explain the process for debugging an algorithm (slide 24) * Have students practice debugging with the three challenge problems in their [student slides](https://www.dropbox.com/scl/fi/xikraqtwsrwqkkpawvm9z/Student_Lesson-4-slides-Grades5-6.pptx?rlkey=8dl1jhonikest4ykvx2szxd6j&dl=0) * Share possible solutions for debugging challenges (slides 25-30) * Note: if you are pressed for time, feel free to choose only one or two of these debugging activities. Check off learning objectives as lesson proceeds (slide 31)   Discussion: what makes a good animation?   * Share that today students will begin animating their work in Scratch. But first, invite students to share their thoughts on what makes a “good” animation (slide 32). * Give an overview of some tips for creating good animations: visuals should match written text (slides 33-38) * Show examples of a good and bad animation (slides 39-40) |
| **(15-20 min) Independent Practice:**  Animating recipes in Scratch   * Instruct students to brainstorm ideas for animating their recipes with a partner (slide 41) * Instruct students to open their recipe, which should already be filled in on CoCo. They should also open Scratch. Then, instruct students to answer the questions in CoCo column 2 to identify the Scratch blocks they will need (slides 43-45).   + There is an optional video on how to use CoCo columns 2 and 3 on slide 45 * Once CoCo is filled in, instruct students to begin coding! As they begin finding those blocks in Scratch, they should check off the self-monitoring column to track their progress (slides 46-48)   + Optional videos on moving between CoCo and Scratch, and using the self-monitoring column can be found on slides 47 and 48. * Ask students to pause and reflect on whether or not they caught any errors (bugs) in their code, and if their animations match their written recipes (slides 50-51) |
| **(5 min) Wrap up:**   * Instruct students to fill out the self-evaluation section at the bottom of CoCo, rating how they felt about their written project, CoCo, coding, and Scratch animation today (slides 52-53) * **Important:** Instruct students to share their Scratch projects to the teacher studio using the appropriate naming scheme (student name + lesson 4 story); [video available](https://www.dropbox.com/s/6o6iu58m61nyctq/Student%20-%20How%20To%20Add%20A%20Project%20To%20A%20Studio%20In%20Scratch.mp4?dl=0) on slide 55 if needed. * If time: instruct students to share their animations with a partner (slide 56) |
| **Assessment Strategy:**  Did the student…   * Review familiar Scratch blocks with a scavenger hunt * Identify new Scratch sound blocks: play sound, record/upload a sound, change the pitch effect of a sound, stop all sounds * Practice using Scratch sound blocks * Define code, bugs, and debugging * Understand how to debug an algorithm * Discuss and understand the features of a good animation * Finish animating my recipe in Scratch * Evaluate their work * Share their animation with a partner |