|  |
| --- |
| ***Overview*** |
| *Topic/Theme*Please list a brief title for the task | Sequences |
| *Lesson/Activity Goal* | Students will be able to write a program to draw a square. |
| *Rationale and Unit Placement*Please provide a few sentences that describe how this lesson or activity might fit within an existing unit. | This lesson aligns with an accompanying math lesson on shapes. Students can apply the properties of a square (congruent sides, congruent right angles) to their drawings. This project can be extended to shapes other than squares and put within the greater context of parallelograms and quadrilaterals. |

|  |
| --- |
| ***Standards Alignment***Please list the standards aligned with this task (e.g. K-2PA.2) |
| Computer Science Standards3-5.DI.1 Understand and use the basic steps in algorithmic problem solving (e.g., problem statement and exploration, examination of sample instances, design, implementation, and testing).3-5.DI.2 Develop a simple understanding of an algorithm (e.g., search, sequence of events, or sorting) using computer-free exercises.3-5.PA.3 Implement problem solutions using a block-based visual programming language.3-5.DI.5 Understand the connections between computer science and other fields.3-5.PA.1 Use technology resources (e.g., calculators, data collection probes, mobile devices, videos, educational software, and web tools) for problem-solving and self-directed learning, and general-purpose productivity tools and peripherals to support personal productivity, remediate skill deficits, facilitate learning, and individual/collaborative writing, communication, and publishing activities.Math StandardsPS.1: Make sense of problems and persevere in solving them.**3.G.2** Understand that shapes (e.g., rhombuses, rectangles, and others) may share attributes (e.g., having four sides), and that the shared attributes can define a larger category (e.g., quadrilaterals). Recognize and draw rhombuses, rectangles, and squares as examples of quadrilaterals.  Recognize and draw examples of quadrilaterals that do not belong to any of these subcategories. |

|  |
| --- |
| ***Attributions*** |
| *Created by:* | Catherine Floyd |
| *Date:* | May 22, 2019 |

---------------------------------------------------------------------------------------------------------------------

|  |
| --- |
| ***I. Introduction/Anticipatory Set***How might you make connections to students’ own experiences/ideas or other content to set the stage for the lesson/activity? |
| There are many real-world examples of squares, and computer-generated squares. One way to motivate this lesson might be to compile a collection of computer-generated square patterns or pictures to show the students. Ask them what they notice, how they think these were made, and what skills computer scientists might need to create something like this.  |

|  |
| --- |
| ***II. Summary Description***Please describe the procedures or parts of the lesson/activity. If you are using an existing activity, you can include a link to the instructions. Feel free to provide any further instructions for how this lesson/activity might be adapted for be integrated with other parts of the unit.  |
| **Day 1**Task 1: (Unplugged) Walk a Square In front of the room, the teacher demonstrates the action of walking in a square. Ask students to write down the algorithm (directions) to describe these actions. Have students compare their directions in partners, looking for similarities and differences in their codes. Class discussion:* What qualities of a square are important to highlight in the code? [congruent sides, right angles]

Depending on time, have students brainstorm how they would describe drawing other quadrilaterals such as rectangles, parallelograms, etc.**Day 2**Task 2: (Plugged) Investigate the tools in ScratchDemonstrate how to find the control blocks of code in Scratch. Have students experiment with each of the pen tools to figure out what they do and how they might be used. While teachers can use a [guide](https://en.scratch-wiki.info/wiki/Motion_Blocks), give students a chance to experiment with the tools and report to the class about what each one does.Demonstrate how to find the pen block in the Scratch extension blocks. Have students experiment with each of the pen tools to figure out what they do and how they might be used. While teachers can use a [guide](https://en.scratch-wiki.info/wiki/Pen), give students a chance to experiment with the tools and report to the class about what each one does.Class discussion:* If we want to have our sprite draw a square on a computer, what tools might we need? [pen, movement, turning, straight lines, etc.]

**Day 3**Task 3: (Plugged) Program a sprite to draw a square in ScratchGrouping students in pairs, ask them to write a program where the sprite draws a square. Have students annotate (using the comments function) to describe important aspects of their program. Once students have finished their programs, have them do a short gallery walk.Class discussion: * How were the programs of the other students similar/different from your own program?
* What (optional) tools helped customize the squares? [change colors, backgrounds, etc.]
* What tools might we need to make our programs more efficient? [might lead to a motivating conversation around the idea of needing something similar to a loop]
 |

|  |
| --- |
| ***III. Whole Group Discussion Central Questions***What 1-2 central questions might be used with the whole class to solidify the main idea of the lesson/activity? |
| What qualities of a square are important to highlight in our code? |
| What programming tools might we need in order to have the sprite draw a square? |
| What tools might we need to make our programs more efficient? |

---------------------------------------------------------------------------------------------------------------------

|  |
| --- |
| ***Evidence of Proficiency***Please list 3-5 categories and brief descriptions of what proficiency looks like for those categories (this can serve as the precursor to a rubric). You do not have to use all of the rows. |
| Category | Description |
| 1. Sequencing | Students create a program that includes sequences of steps to draw a square successfully. |
| 2. Collaboration | Students work together to create a program, sharing the responsibility of thinking and coding. |
| 3. Annotations | Students provide annotations/comments to prove that they understand the purpose and utility of their chosen program functions. |