|  |  |  |
| --- | --- | --- |
| **Name:** John McClellan | **Contact Info:** jmcclellan@readingschools.org | **Date:**11/01/17 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Lesson Title :** Coding with Scratch | **Unit #:**1 | **Lesson #:**2 | **Activity #:**4 |
| **Activity Title:** EDP – Design a Computer Program |

|  |  |
| --- | --- |
| **Estimated Lesson Duration:** | 4 Days |
| **Estimated Activity Duration:** | 3 Days |

|  |  |
| --- | --- |
| **Setting:** | Classroom |

|  |
| --- |
| **Activity Objectives: The student will be able to:**  |

1. Sketch a plan in pseudo-code.
2. Use Scratch proficiently enough to create an automated system.
3. Refine their code in the model of the Engineering Design Process
4. Read and explain their code

|  |
| --- |
| **Activity Guiding Questions:**  |

* How can you check for errors/bugs in Scratch?
* How much control does Scratch give the user?
* What can be automated in Scratch?
* How can code be refined?

| **Next Generation Science Standards (NGSS)**  |
| --- |
| **Science and Engineering Practices (Check all that apply)**  | **Crosscutting Concepts (Check all that apply)** |
| ☒ Asking questions (for science) and defining problems (for engineering) | ☐ Patterns |
| ☐ Developing and using models | ☒ Cause and effect |
| ☒ Planning and carrying out investigations | ☐ Scale, proportion, and quantity |
| ☐ Analyzing and interpreting data | ☒ Systems and system models |
| ☒ Using mathematics and computational thinking | ☐ Energy and matter: Flows, cycles, and conservation |
| ☐ Constructing explanations (for science) and designing solutions (for engineering) | ☒ Structure and function.  |
| ☐ Engaging in argument from evidence | ☐ Stability and change.  |
| ☒ Obtaining, evaluating, and communicating information  |  |

| **Ohio’s Learning Standards for Science (OLS)** |
| --- |
| **Expectations for Learning – Cognitive Demands (Check all that apply)** |
| ☒ Designing Technological/Engineering Solutions Using Science concepts **(T)** |
| ☐ Demonstrating Science Knowledge **(D)** |
| ☒ Interpreting and Communicating Science Concepts **I** |
| ☐ Recalling Accurate Science **I** |

| **Ohio’s Learning Standards for Math (OLS) and/or** **Common Core State Standards – Mathematics (CCSS)** |
| --- |
| **Standards for Mathematical Practice (Check all that apply)** |
| ☐ Make sense of problems and persevere in solving them | ☒ Useappropriate tools strategically |
| ☒ Reason abstractly and quantitatively | ☒ Attendto precision |
| ☒ Construct viable arguments and critique the reasoning of others | ☒ Look for and make use of structure |
| ☐ Model with mathematics | ☒ Look for and express regularity in repeated reasoning |

|  |
| --- |
| **Unit Academic Standards (NGSS, OLS and/or CCSS):** |

Standard for Mathematical Practice #2 Reason Abstractly and Quantatively; Standard for Mathematical Practice #5 Use Appropriate Tools Strategically; CCSS High School Modeling Domain: *“When making mathematical models, technology is valuable for varying assumptions, exploring consequences, and comparing predictions with data.”*

|  |
| --- |
| **Materials**: (Link Handouts, Power Points, Resources, Websites, Supplies) |

* Chromebooks
* Blank Program link
* EDP packet for guidance

|  |
| --- |
| **Teacher Advance Preparation:** |

* Make sure you are in a place where the Wi-Fi is strong. This a web-based compiler, and will not work if there is not access to the internet.
* Break into groups based on skill, as it might make sense to have a proficient coder with a creative individual.
* In terms of “research” for EDP, I would suggest finding a couple of links yourself to get the students started.
* On Day 1, limit the Chromebook use to one chromebook per group, and only after they’ve completed the EDP packet to step 4.
* Have the students who worked the least on the actual computers lead the presentations on day 3.

|  |
| --- |
| **Activity Procedures:** |

Day 1:

1. Since the students were formally introduced to the challenge in a previous lesson, they should already be aware of the challenge. However, it should be reiterated on the board. Take the first few minutes of class talking about automation in the context of code (here is where global impacts should be discussed from CBL).
2. Hand out the packets and divide the students into their predetermined groups. Groups need to be signed off by the teacher in order to move to the next page of the EDP packet page. Chromebooks are taken away until step 4 of the EDP (implementing solution).
3. Students should have some character movement on their screens by the end of the bell.

Day 2:

1. Continuing from yesterday, if students have the ability to continue working on their chromebooks. Each individual can work on it now, but encourage them to have one “master” chromebook that will keep their changes. This allows the students to play around on their own.
2. Have the students finish their project as far as they can, and keep pushing the refinement stage of EDP.

Day 3

1. Students should present their code and the animation to the class. Talk about the growth of the program and where they would go if they had more time. At least two of the students should talk, and pick them based on who was on the computers the least.
2. After all the presentations, give the post-test, which should be the same as the pretest from lesson 1 activity 1.

**Formative Assessments:** Link the items in the Activities that will be used as formative assessments.

* Whole-class and small-group discussion… monitoring small group work to make sure they understand and stay on track.

**Summative Assessments:** These are optional; there may be summative assessments at the end of a set of Activities or only at the end of the entire Unit.

Post-test (same test as the pre-test one lesson 1 activity 1)

Presentation of Code

|  |
| --- |
| **Differentiation:** Describe how you modified parts of the Lesson to support the needs of different learners.Refer to Activity Template for details. |

The students who had success were left to their own devices, besides just probing questions. The group with the struggling students were given new expectations given their time constraints. The individual with the 504 was given prompting questions come presentation time, so that he had structure to talk about his code.

|  |
| --- |
| **Reflection:** Reflect upon the successes and shortcomings of the lesson. |

The challenge was an incredible success. Students wanted to build their own mazes as backdrops (instead of my idea of creating “Sprites” for them). The structure of drawing out their pseudo-code first and doing research through the tutorials was incredibly helpful – it prevented students in groups driving ahead of their group and not communicating; instead, it fostered a team approach and a good way to develop their ideas.

The presentations were also great – students had different strategies with their code, but the final results came in that two out of the three groups’ AI could navigate through any maze given to them. The third group ran out of time and did not function well together because of the number of absences.

Should this occur again, I would have rotated the top student in each group through for an “idea sharing session” where the groups talked strategies with other groups. The purpose of this would be to get students talking and maybe get the group who didn’t get much traction rolling through the project.