Hack Your Treasure

Overview:
In this unplugged coding activity, participants will use problem solving skills to program their ship to navigate to the treasure while avoiding obstacles.

Goals:

- Understand the importance of Optimal Efficiency as it applies to writing computer code.
- Design programs for a board game that simulates an automated pirate ship.

Time Required: 45 minutes

Materials:

- Hack Your Treasure Rules
- Hack Your Treasure game board
- Hack Your Treasure game pieces
- Dry Erase Markers and wipes (optional)

Vocab: Optimal Efficiency, Program, Programming, Programming Language

Procedures:

1. Set Up – Print one copy of the game board for each student. You may wish to laminate them for easy reuse.

2. Game pieces are on page 5 of the Hack Your Treasure Instructions pdf. Print tokens on cardstock and cut out. There are enough tokens on the page for 14 Students. Each student needs:
   a. 1 Ship (14 on sheet)
   b. 3 Doubloons (42 on sheet)
   c. 3 Sea Monsters (42 on sheet)
   d. 6 Coral Reefs (84 on sheet)

   Use a ½ inch hole punch to punch out the tokens.

3. Use the slide deck to help walk students through the activity.

4. Ask students,

   “What comes to mind when you hear the term “efficient” or “efficiency”?”

5. Explain the following definition:
   "Optimal Efficiency means to achieve maximum productivity with minimum waste. There are different kinds of efficiency depending on what kind of "waste" you are concerned about. For example, building a car efficiently might mean conserving materials to use the smallest amount of metal or plastic, or it could mean utilizing an assembly line to reduce the total time it takes to build a large number of cars. A given task can be efficient in one way, but inefficient in another. Can you think of examples where this might be the case?" Examples might include:
   i. Hiring more workers might result in a project being completed in less time but costs more money.
   ii. Walking or riding a bike to school is healthier and better for the environment but can take longer than driving.

6. Tie "optimal efficiency" into Computer Science and coding by explaining:
   "Efficiency is important in computer science because computers have limited resources, including time and memory. To ensure they do not overload a computer and that they get results back in a reasonable time, computer scientists must write efficient programs."

7. Introduce the activity by telling students:
   "The action or process of writing computer programs is called programming. In these activities we are going to program our pirate ship to reach the treasure. We want to write efficient programs to make sure we complete the task using the fewest possible instructions. A programming language is used to tell computers exactly what to do, in these puzzles we will use a special set of instructions to move your ships up, down, left, right (but not diagonally). Enter one instruction per spot. Start by writing a program to navigate the ship to the treasure. Remember to use the fewest instructions possible."

8. Give students a chance to write their programs, then discuss:
   "You just wrote a program – you used a set of instructions to tell a device how to do a task. How many of you solved the puzzle using 15 instructions or fewer?" *Allow students to raise hands.*
   "Great! Can anyone tell me what the fewest number of instructions is for this puzzle?" *(10 is the best solution for this puzzle – keep in mind there are multiple correct solutions for each puzzle, this is just an example!)*

1	2	3	4	5	6	7	8	9	10
→	→	→	→	→	↓	↓	↓	↓	↓

9. Now move on to puzzles 2-3:

   “For Puzzles 2 and 3 we are going to add some new components to our game board. Now there are some obstacles in the way of our pirate ship. The first obstacle is a coral reef, ships cannot move onto spots where there is a coral reef. The second obstacle is a sea monster. The pirate ship is equipped with a harpoon that will allow it to fight off the sea monster. To use the harpoon, we will add a new element to our instructions an asterisk ∗. This means you will need to use the ∗ and then the arrow to move into a spot where there is a sea monster. For example in puzzle 2, if you want to move to the spot where there is a sea monster you would put a ∗ in instruction 1 and a → for instruction 2.”

10. Give students a chance to work out a solution for puzzle 2. Remind them to try and solve with the fewest instructions possible.

11. Reflection for puzzle 2:

    “What was the minimum number of instructions needed to solve this puzzle? (10) Was it more efficient to fight the sea monster or go around it? (go around)” Possible solution for puzzle 2:

1	2	3	4	5	6	7	8	9	10
→	→	→	→	→	↓	↓	↓	↓	↓

12. Reset puzzles for puzzle 3 and give students a chance to work out an efficient solution.

13. Reflection for puzzle 3:

    “What was the minimum number of instructions needed to solve this puzzle? (11) Was it more efficient to fight the sea monster or go around it? (fight) Can you think of an instance in the real world where you might chose to go around an obstacle instead of removing it, even though it is more efficient to remove it?” Possible solution for puzzle 3:

1	2	3	4	5	6	7	8	9	10	11
∗	→	→	→	→	→	↓	↓	↓	↓	↓

14. For this next section, we will introduce another new element.

    “In puzzles 4-5 we have added additional treasure in the form of gold doubloons. Use the $ instruction after you move your ship into a spot where the doubloon is to collect it. You must collect all the treasure before you go to the island (finish). You may collect the treasure in any order you want, but remember your goal is to use optimal efficiency”.

15. Give students a chance to solve puzzles 4 and 5. Possible solution for puzzle 4:

1	2	3	4	5	6	7	8	9	10	11	12	13	14
↓	↓	↓	$	→	→	↑	$	→	→	→	↓	↓	↓

Possible solution for puzzle 5:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
↓	↓	↓	↓	↓	$	↑	→	↑	→	↑	→	$	↑	→	↑	→	$	↓	↓	↓	↓	↓

16. Discuss puzzles 4-5:

“Which puzzle (4 or 5) was more challenging to solve? Why? You may have noticed the problem gets more difficult to solve when more doubloons are added to the board. In Puzzle 4 there were only 2 routes you had to check, but in puzzle 5 there were 6 possible routes. If there were 4 doubloons, there would be 24 possible routes. With 10, there would be 3.6 million possible routes! This kind of problem is called the “traveling salesman problem”. Computer scientists are interested in problems like this because it can take a computer an extremely long time to find the optimal solutions when there are a lot of possibilities to check. This problem can be restated to apply to many real-world scenarios, such as mail delivery and airline flight planning, so it is one that computer scientists would like to be able to solve efficiently. Using some clever tricks, mathematicians and computer scientists have found ways to not have to check every possible path. Even using those tricks, however, the problem can still take a long time for a computer to solve, and computer scientists often settle for a “good enough” solution.”

Extensions:

1. Try puzzles 6 and 7. Puzzle 6 incorporates all the elements we learned and in puzzle 7 students will be creating their own challenge to trade with a partner and solve. Use no more than 3 doubloons, 3 sea monsters and 6 coral reefs for your puzzle.

2. Research and learn more about the “traveling salesman problem”. Check out these resources:
   a. bit.ly/NYSDsalesman1
   b. bit.ly/NYSDsalesman2
   c. bit.ly/NYSDsalesman3

Reflection:

1. Why is it important to understand efficiency?
2. What are some challenges computer programmers face?
3. What is another real-world example of the “traveling salesman problem”?
4. How might you use this what you learned?

Adapted from the following Lesson:

Activity adapted from *Game Changers - Hack Your Harvest* created by WVU Extension https://extension.wvu.edu/stemcare