Lesson 1.5: Searching Algorithms

Objectives

In this lesson, students will:

- Gain an understanding of searching algorithms used by software
- Discover and discern which algorithms are better and what a better searching algorithm means

Agenda

1. Introduction	10 mins
2. Linear Search	10 mins
3. Binary Search	10 mins
4. Hash Search	10 mins
5. Wrap Up and Reflections	10 mins

Preparation

- Print student activity worksheets (one per student, 1A', 1B', 2A',... are backups in case students accidentally see the location of their partner's ship; hence print a few copies only)

Resources & Links

- A Searching Algorithm Lesson Plan: https://csunplugged.org/en/topics/searching-algorithms/

1. Introduction

Engage students in a brief introduction to searching algorithms:

Computers store a lot of information, and they need to be able to go through it quickly. One of the biggest search problems in the world is faced by Internet search engines, which must search billions of web pages in a fraction of a second.

There are different ways for a program to search for information. Today, we are going to discover different algorithms to search for something and we will discover that some are better than others.

Prompt students to the meaning of an algorithm. An algorithm is a list of steps to accomplish a task.

What does it mean for one algorithm to be better than another in computer programming?
A better algorithm is one that requires fewer steps because then the computer has to do fewer operations. When a computer has to do fewer operations it means the program will run faster and use fewer resources.

Today, we are going to experience using different searching algorithms and discover which ones require fewer steps.

2. Linear Search

Instructions to give to students:

1. Organise yourselves into pairs. One of you has sheet 1A, the other sheet 1B. Don't show your sheet to your partner!
2. Both of you circle one battleship on the top line of your game sheet and tell your partner its number.
3. Now take turns to guess where your partner's ship is. (You say the letter name of a ship and your partner tells you the number of the ship in that letter.)
4. How many tries does it take to locate your partner's ship? This is your score for the game.

Note: Sheets 1A' and 1B' are extras provided for children who would like to play more games or who "inadvertently" see their partner's sheet. Sheets 2A', 2B' and 3A', 3B' are for the later games.

Discussion Points:

- What were the scores?
- What would be the minimum and maximum scores possible? (They are 1 and 26 respectively, assuming that the children don't shoot at the same ship twice. This algorithm is called 'linear search', because it involves going through all the positions, one by one.)

3. Binary Search

Instructions to give to students:

1. Organize yourselves into pairs. One of you has sheet 2A, the other sheet 2B. Don't show your sheet to your partner!
2. Your ships and your partner's ships are sorted in ascending order.
3. Both of you circle one battleship on the top line of your game sheet and tell your partner its number. This is the number of the battleship your partner must find. The number that your partner gives you is the number of his battleship you must find.
4. Now take turns to guess where your partner's ship is. (You say the letter name of a ship and your partner tells you the number of the ship at that letter.)
5. How many guesses does it take to locate your partner's ship? This is your score for the Game.

Discussion Points:

- What were your scores?
- What strategy did the low scorers use?
- Which ship should you choose first? (The one in the middle tells you which half of the line the chosen ship must be in.) Which location would you choose next? (Again the best strategy is always to choose the middle ship of the section that must have the selected ship.)
- If this strategy is applied, how many guesses will it take to find a ship? (Five at most). This method is called 'binary search', because it divides the problem into two parts.

4. Hashing Search

Instructions to give to students:

1. Each take a sheet 3A and 3B as in the previous games and tell your partner the number of your chosen ship.
2. In this game you can find out which column (0 to 9) the ship is in. You simply add together the digits of the ship's number. The last digit of the sum is the column the ship is in. For example, to locate a ship numbered 2345, add the digits 2+3+4+5, giving 14. The last digit of the sum is 4, so that ship must be in column 4. Once you know the column you need to guess which of the ships in that column is the desired one. This technique is called 'hashing', because the digits are being squashed up ("hashed") together (like hash browns).
3. Now play the game using this new searching strategy. You may like to play more than one game using the same sheet—just choose from different columns.

Note: The spare sheets 3A' and 3B' must be used as a pair, because the pattern of ships in columns must correspond.)

Discussion Points:

1. Collect and discuss scores as before.
2. Which ships are very quick to find? (The ones that are alone in their column.) Which ships may be harder to find? (The ones whose columns contain lots of other ships.)

5. Wrap Up and Reflections

Follow Up Discussion:

- Which of the three searching algorithms is fastest? Why?
- What are the advantages of each of the three different ways of searching? (The second strategy is faster than the first, but the first one doesn't require the ships to be sorted into order. The third strategy is usually faster than the other two, but it is possible, by chance, for it to be very slow. In the worst case, if all the ships end up in the same column, it is just as slow as the first strategy)
- Watch the following video: https://youtu.be/iDVH3oCTc2c . Which searching algorithm was used?

For Your Reference: What's it all about?

Computers store a lot of information, and they need to be able to sift through it quickly. One of the biggest search problems in the world is faced by Internet search engines, which must search billions of web pages in a fraction of a second. The data that a computer is asked to look up, such as a word, a bar code number or an author's name, is called a *search key*.

Computers can process information very quickly, and you might think that to find something they should just start at the beginning of their storage and keep looking until the desired information is found. This is what we did in the Linear Searching Game. But this method is very slow—even for computers. For example, suppose a supermarket has 10,000 different products on its shelves. When a barcode is scanned at a checkout, the computer must look through up to 10,000 numbers to find the product name and price. Even if it takes only one thousandth of a second to check each code, ten seconds would be needed to go through the whole list. Imagine how long it would take to check out the groceries for a family!

For more in depth activities on linear search algorithm, check

https://csunplugged.org/en/topics/searching-algorithms/sequential-and-binary-search-unit-plan/how-many-guesses/

A better strategy is *binary search*. In this method, the numbers are sorted into order. Checking the middle item of the list will identify which half the search key is in. The process is repeated

until the item is found. Returning to the supermarket example, the 10,000 items can now be searched with fourteen probes, which might take two hundredths of a second—hardly noticeable.

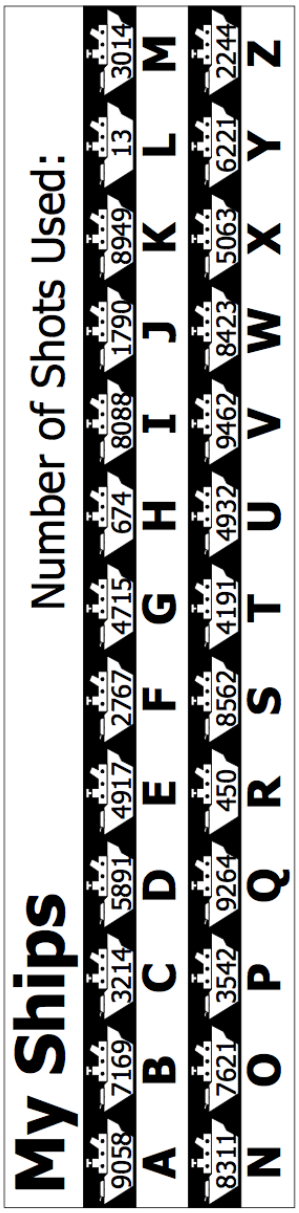
Additional activity:

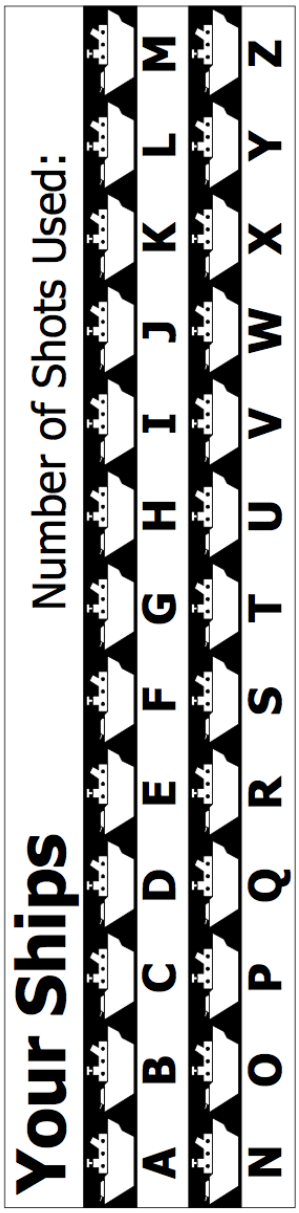
https://csunplugged.org/en/topics/searching-algorithms/integrations/drama-video/

A third strategy for finding data is called *hashing*. Here the search key is manipulated to indicate exactly where to find the information. For example, if the search key is a telephone number, you could add up all the digits in the number and take the remainder when divided by 11. In this respect, a hash key is a little like a small piece of data whose value depends on the other data being processed. Usually the computer will find what it is looking for straight away. There is a small chance that several keys end up in the same location in which case the computer will need to search through them until it finds the one it is seeking.

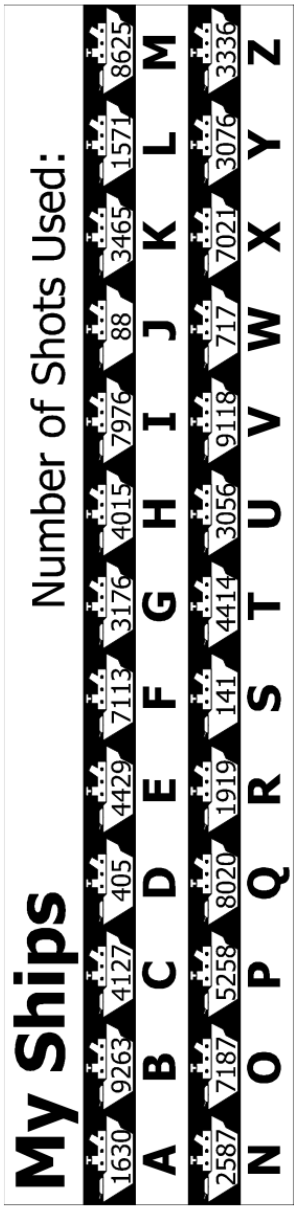
Computer programmers usually use some version of the hashing strategy for searching, unless it is important to keep the data in order, or unless an occasional slow response is unacceptable.

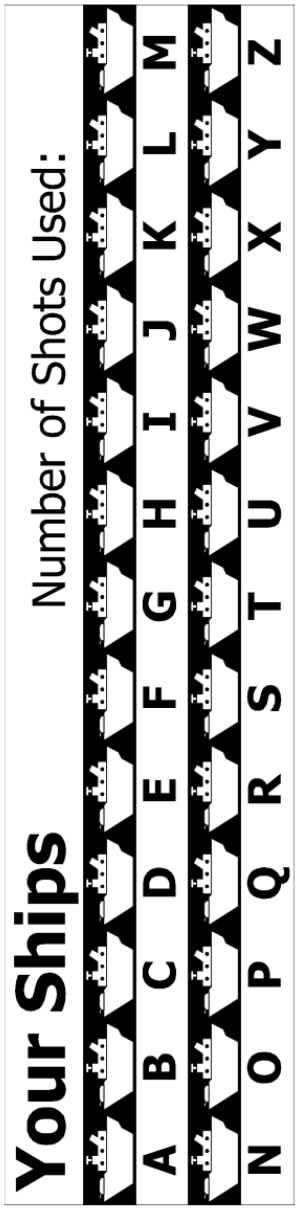
Student Activity Worksheets Follow



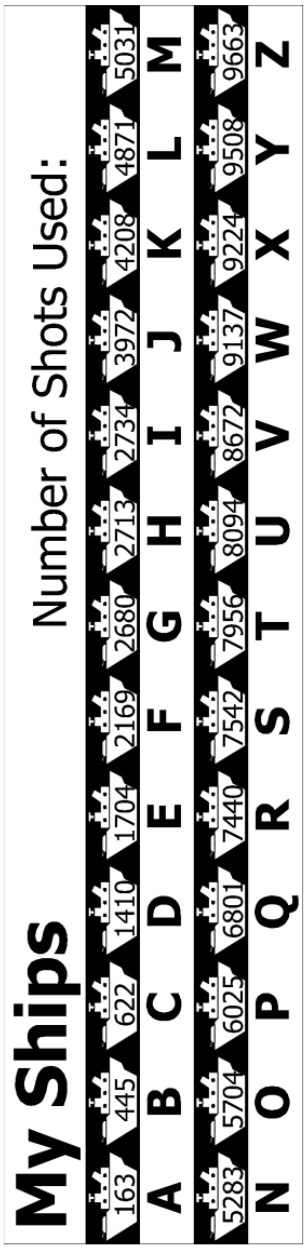


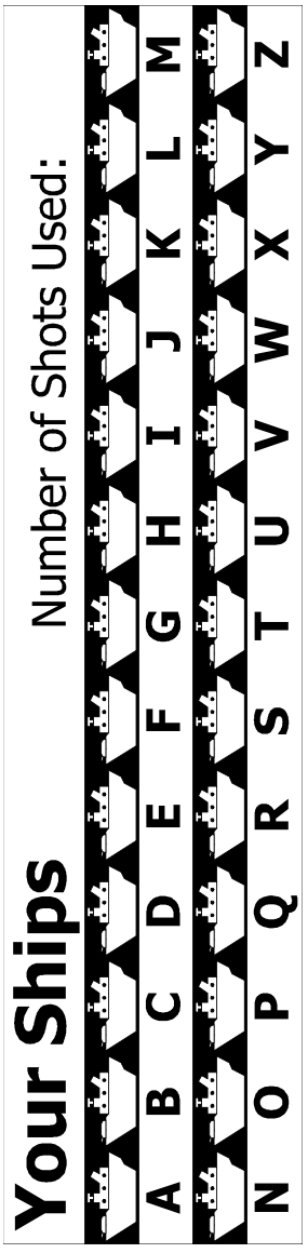
1A



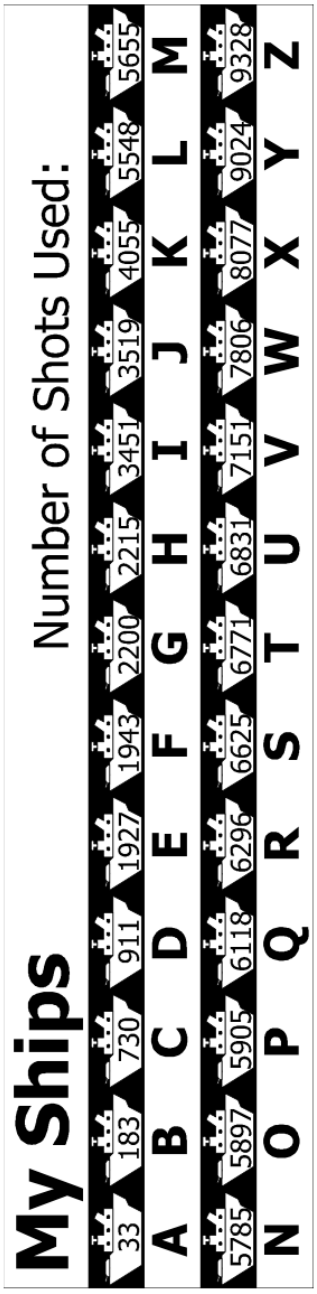


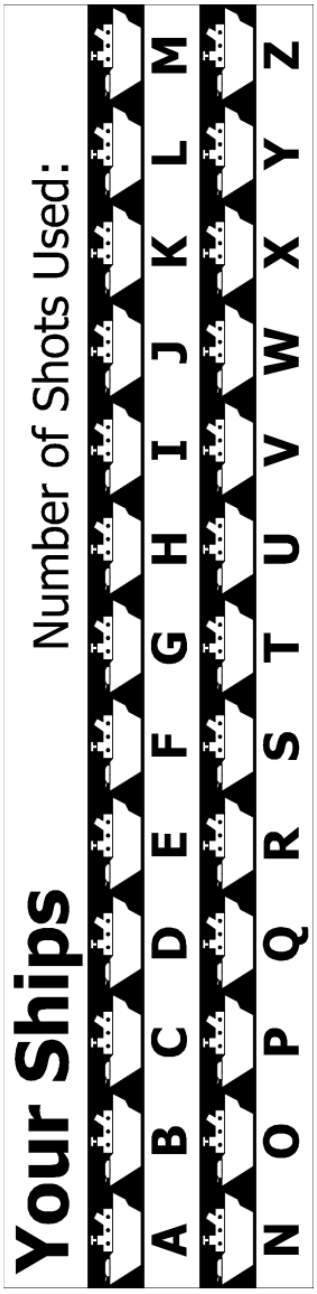
1B





2A





2B

My Ships

Number of Shots Used:

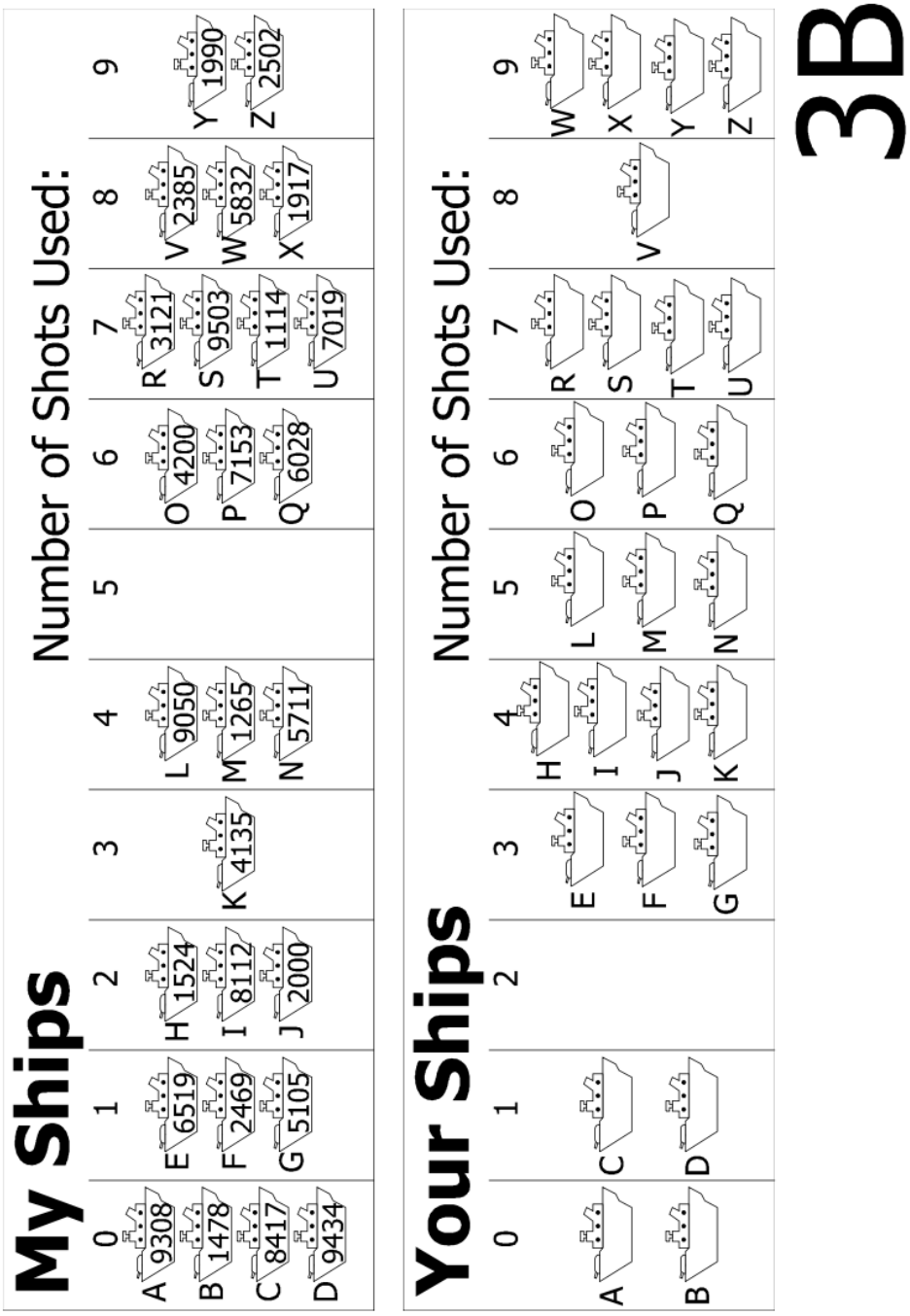
0	1	2	3	4	5	6	7	8	9
A 9047 B 1829	C 3080 D 9994		E 5125 F 1480 G 8212	H 8051 I 1481 J 4712 K 6422	L 7116 M 8944 N 4128	O 6000 P 7432 Q 4110	R 9891 S 1989 T 2050 U 8199	V 4392	W 1062 X 2106 Y 5842 Z 7057

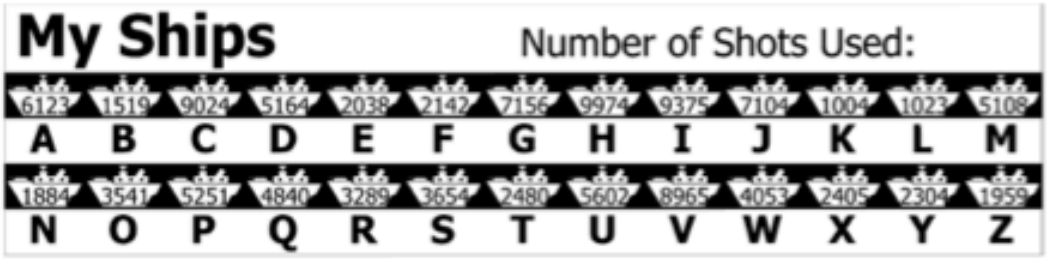
Your Ships

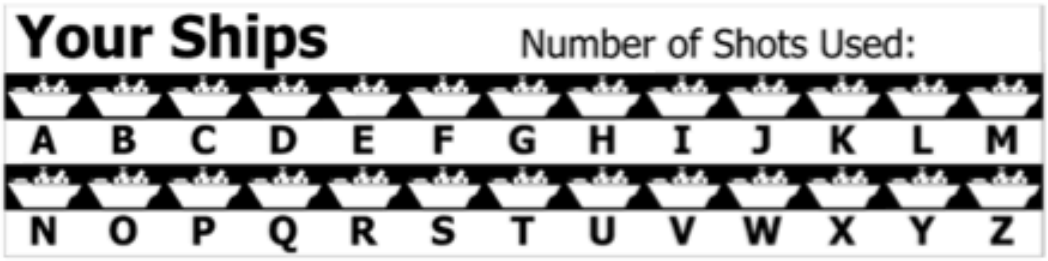
Number of Shots Used:

0	1	2	3	4	5	6	7	8	9
A B C D	E F G	H I J	K	L M N		O P Q	R S T U	V W X	Y Z

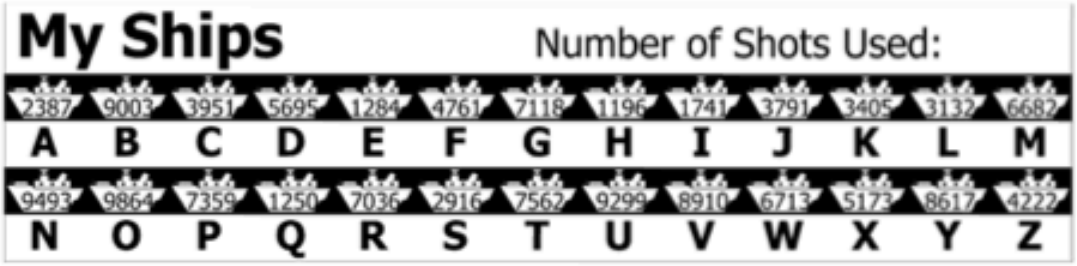
3A

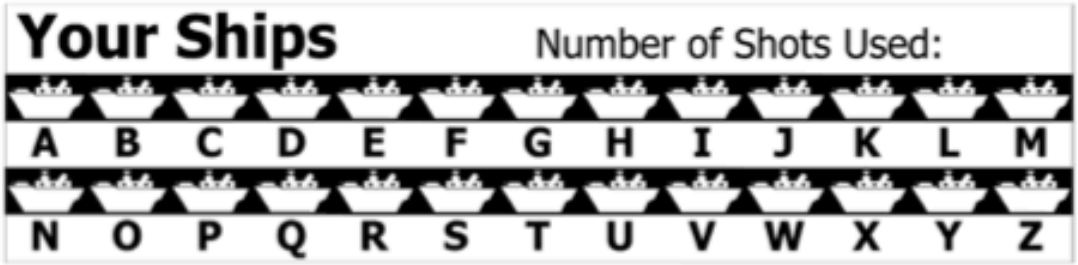




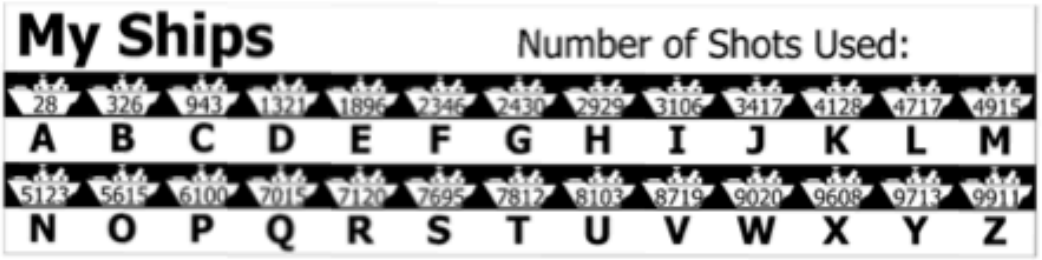


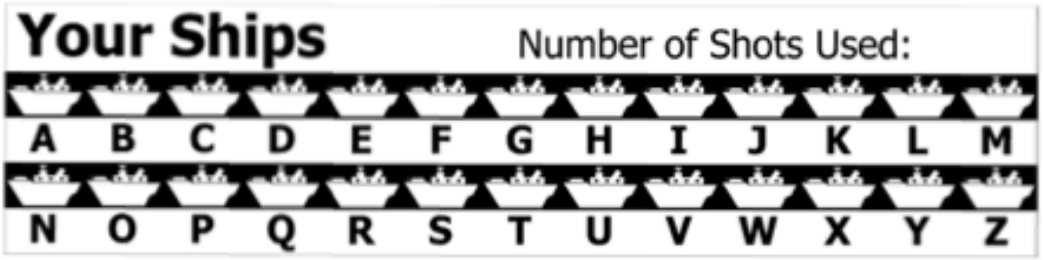
1A′



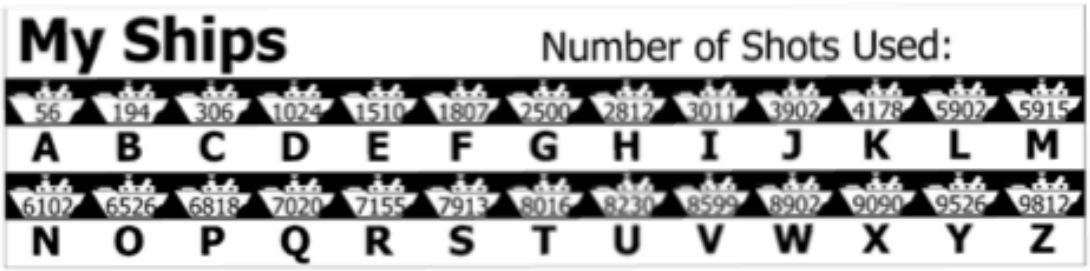


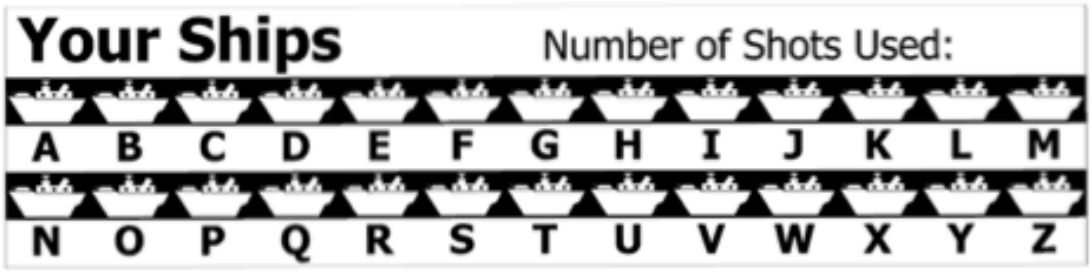
1B′



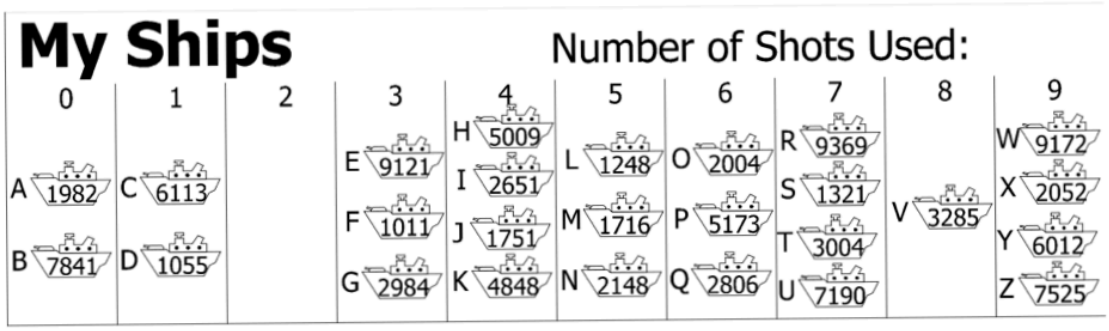


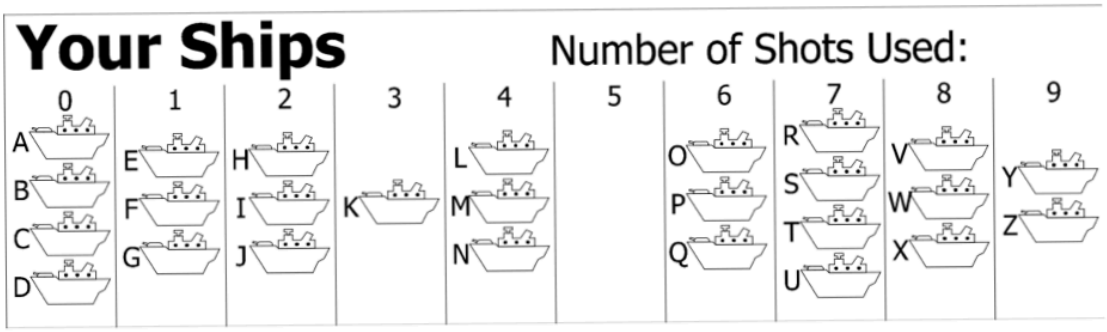
2A′



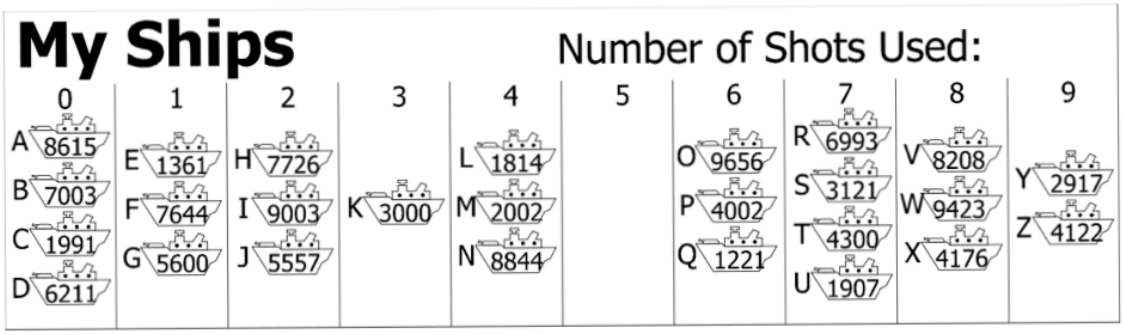


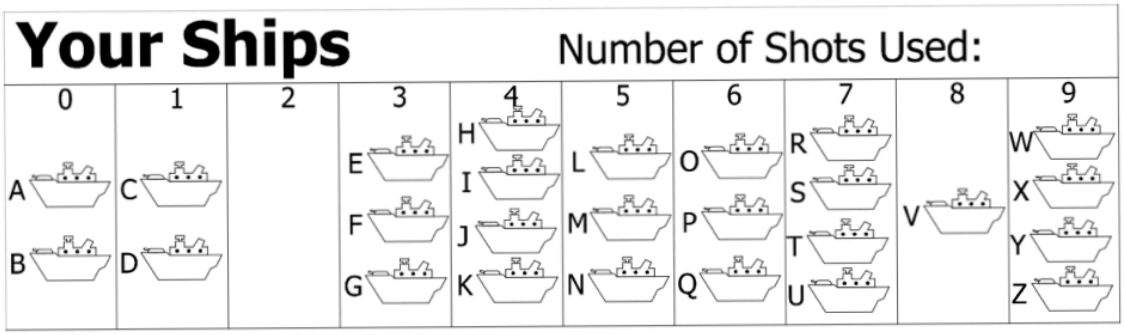
2B′





3A′





3B′