

# OzoBlockly Editor—**Computer Science via the Buzz-Buzz Game**

## Teacher's Guide

### **Grades 6 – 12**

To gain a feeling for how students can learn a variety of computer science concepts through the “Buzz-Buzz” game, please read through the four-page student guide.

***This lesson involves the use of several computer science programming concepts including:***

- The use of variables to hold the values of program constants
- Subroutines to break up code into logical segments
- The use of conditionals, as needed, to control the flow of logic
- Subroutine and variable names that support self-documentation

***Here are some ways to use this lesson with your students:***

- Begin with asking the students to break up into groups of six, *avoiding groups of 7---why?* Have each group play the game for several minutes—*no calculators or pen and paper allowed*. Things get interesting once they get to the numbers 70 through 79, as they need to keep track of what the current count is in their heads!
- You can then have the students work at their computers with Ari or Evo, building and testing the program presented in the student guide.
- If class time is limited, you can give them the share code for the program **q7xvtjv** and have them do the program modification exercise presented in the student guide. The share code for the modified program allowing numbers through 199 is **aq4gp5h**.

#### ***Applicable CSTA (Computer Science Teachers Association) Standards***

|         |                                                                                                                            |
|---------|----------------------------------------------------------------------------------------------------------------------------|
| 2-CS-02 | Design projects that combine hardware and software components to collect and exchange data.                                |
| 2-AP-11 | Create clearly named variables that represent different data types and perform operations on their values.                 |
| 2-AP-12 | Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. |
| 2-AP-13 | Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.            |
| 2-AP-19 | Document programs in order to make them easier to follow, test, and debug.                                                 |

- 3A-DA-11 Create interactive data visualizations using software tools to help others better understand real-world phenomena.
- 3A-AP-17 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
- 3B-CS-02 Illustrate ways computing systems implement logic, input, and output through hardware components.
- 3B-AP-14 Construct solutions to problems using student-created components, such as procedures, modules and/or objects.