Activity

BUZZ BUZZ GAME

CREATED BY

Richard Born
Associate Professor Emeritus
Northern Illinois University
rborn2@niu.edu

TOPICS

Robotics, Programming

GRADES

7-12

METHOD

OzoBlockly

DURATION

60 minutes

OzoBlockly Challenge: The Buzz-Buzz Game

By Richard Born
Associate Professor Emeritus
Northern Illinois University
rborn2@niu.edu

There are a variety of variations of a game called "Buzz-Buzz". Here is how the variation works that we want to program Evo to play using OzoBlockly. Evo is to count out loud starting at zero and proceeding through 127. For any number containing the digit 7 or any number that is exactly divisible by 7, Evo is to play the note C6 (High C on the piano) for a half second instead of speaking the number. We'll call this response from Evo a ***Buzz***. For numbers containing the digit 7 ***and*** are also divisible by 7, Evo is to play the note C6 twice instead of speaking the number. We'll call this response from Evo a ***Buzz-Buzz***.

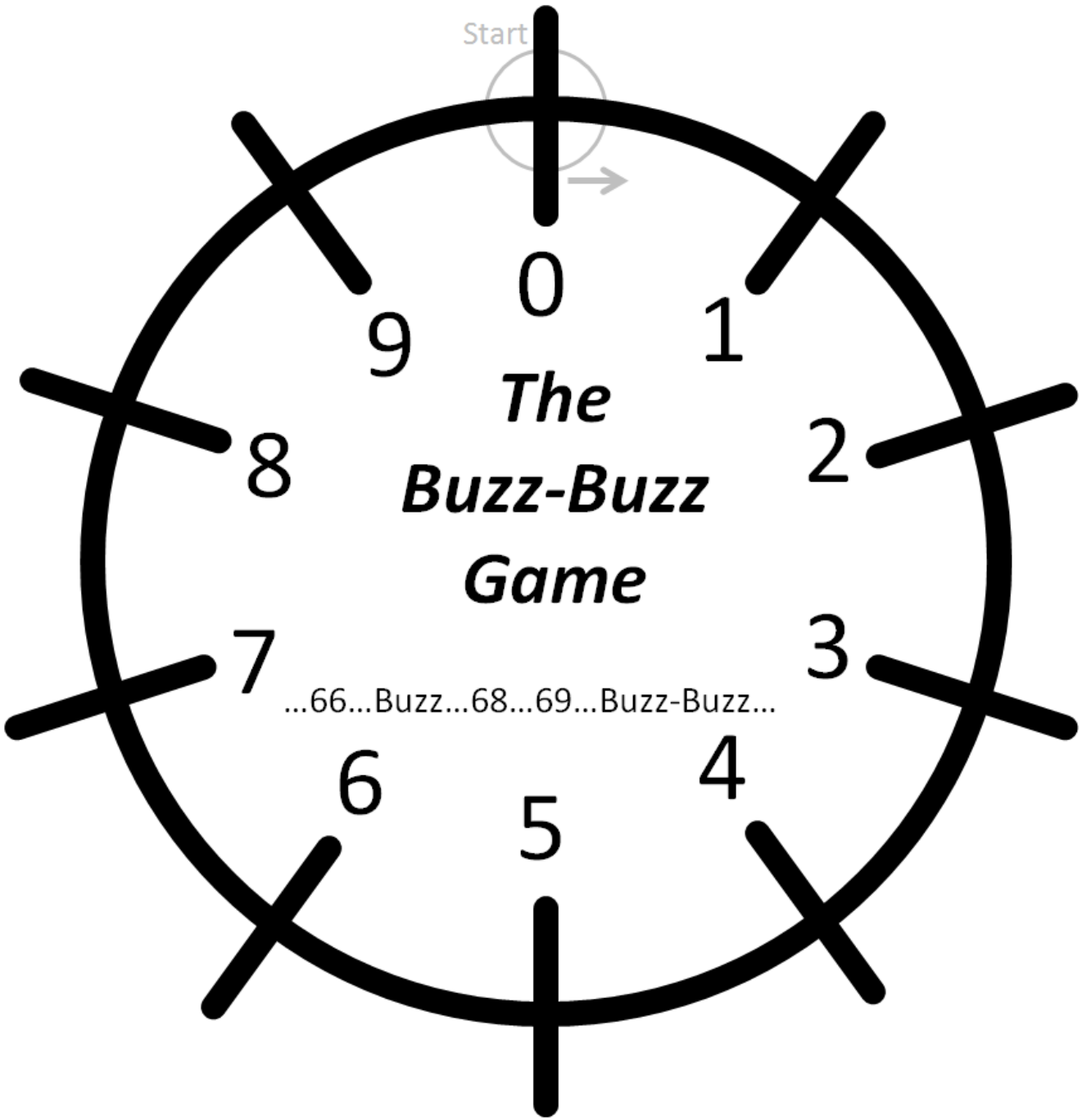
We don't want Evo to just sit in one spot while he is "counting". Rather, we want him to go around the circle on the Ozomap that appears on the next page. The OzoBlockly program should be started after placing Evo on the intersection at 0, facing the direction shown by the gray arrow. The first time around the circle, he will count 0 through 9 by the Buzz-Buzz Game rules. The second time around the circle he will count from 10 through 19, and so on, until he has counted to 127. He should stop at each intersection momentarily while speaking the count, or playing a Buzz or Buzz-Buzz note.

So… put your coding skills and ingenuity to work and see if you can get Evo to play ***Buzz-Buzz***! If you want, you can get fancy and set his top LED one color when speaking the count, another color while playing a Buzz note, and yet another color while playing Buzz-Buzz notes.

Here are just a few suggestions:

1. Design a function with *input* X that *returns* **true** if X is divisible by 7 and returns **false** if X is not divisible by 7.
2. Design a function with *input* X that *returns* **true** if X contains the digit 7 and returns **false** if X does not contain the number 7.
3. In the main program, make use of the two above functions in an *if...else if…else* block that contains compound conditions using **and**/**or** as required by Buzz-Buzz game logic.

When *people* play this game, each player takes his or her turn in the counting process, speaking the number, Buzz, or Buzz-Buzz as appropriate. If a player makes a mistake, then counting starts all over again at zero with the player making the mistake. Oh yes—be careful when starting to count at zero! If interested, you can find some variations in playing this game by searching the internet.


Start
0
1
2
3
4
5
6
7
8
9
The
Buzz-Buzz
Game
...66...Buzz...68...69...Buzz-Buzz...