Getting Started With JavaScript Blocks: Magic 8

Magic 8
Overview
The sixth project asks the user to recreate the Magic 8 ball game. The micro:bit runs two blocks, one asking the user to “ask a question” and the other to randomly generate an answer following a shake of the micro:bit.

.hex File
Final version of the .hex file called “06 Magic 8.hex”

POS Reference
Designs simple algorithms using loops, and selection i.e. if statements. (AL)
Declares and assigns variables. (AB)
Uses a variable and relational operators within a loop to govern termination. (AL) (GE)
Uses logical reasoning to predict outcomes. (AL)
Detects and corrects errors i.e. debugging, in algorithms. (AL)
Creates programs that implement algorithms to achieve given goals. (AL)
Understands that programming bridges the gap between algorithmic solutions and computers. (AB)
Uses nested selection statements. (AL)

I can statements
The students can:
• Use a gesture input to control the micro:bit.
• Create a randomly generated variable
• Use an IF...THEN...ELSE
• Display own patterns on the LEDs

Challenges
A number of challenges are based upon this task.
• Adding additional answers
• Add a thinking animation following a shake gesture.

Next Steps
Lesson 7 uses the LEDs to display a picture on a second micro:bit using the radio block.
Getting Started With JavaScript Blocks: Magic 8

Step by Step

This program recreates the “magic 8” ball. It asks the user to ask a question and it will give a random response.
The first state is to tell the user to ask a question. A string is to be displayed on the screen and this is achieved using the “show string” block found in the basic option of the block menu.

Drag the “show string” block onto the workspace and place it within an “on start” block. Click on the text “hello!” and change it to “Ask a question!”
The next stage is to display the number “8” on the screen to represent the magic 8 ball logo. The steps are repeated again.

Now the program displays “ask a question” and “8”, the user needs to get a response from the micro:bit. This is achieved by using the “Shake” gesture as an input. On the block menu choose the “input” option and select the “on shake” block and drag it onto the workspace.
Getting Started With JavaScript Blocks: Magic 8

Once on the worksheet the “on” block can be changed to a number of gestures, such as tilt left or right. For this project, the shake gesture is selected.

The display of the micro:bit needs to be updated first by using a “Clear” block. This will clear any pattern currently being displayed on the micro:bit whether it be the sentence “Ask a question” of the number “8”. The “clear screen” block is found in the “basic” option under the “More” option of the block menu. Drag the block on to the workspace and place within the “on shake” block.
Now the micro:bit is programmed so it will respond to a shake gesture the next part is to create a variable that a random number will be stored in. From the block menu select the “Variable” and click “Make a Variable”.

A popup dialogue box will then ask you to name your variable. In this example name it “Random”
Getting Started With JavaScript Blocks: Magic 8

When the micro:bit is shaken we would like the random number to be generated and stored in the variable “Random”. To achieve the setting of the variable, we select the “Set variable” and drag into the workspace.

Place the “Set variable” block under the “clear screen” block and click on the down arrow next to the word item. This will allow us to select the variable “random” which will currently be set to “0” when the micro:bit is shaken.
Getting Started With JavaScript Blocks: Magic 8

Now that the variable is set to “0” when the micro:bit is shaken, we now need to assign a random number to it. From the block menu select the “Math” option and select “pick random 0 to 4”.

Drag the “pick random” block onto the workspace and join to the “set random” block. A random number between 0 and 2 will give us the 3 options needed.

The next section is to including a selection process using the IF statement found within the “logic” option within the block menu.
Drag the “IF” block into the workspace and connect it to follow the “random” block. So if the micro:bit is shaken the screen is currently cleared, an random number is generated and stored in the variable “Random” and now we will set what happen for each of the different variable conditions.

As there is more than 2 conditions within this selection another selection statement is required, a “else if”. This is achieved by clicking on the cog on the “if” block and dragging an “else if” block on the “if” block.
Next there we need to create a logical statement for the IF statement to be based upon to allow for selection. From the “logic” option on the block menu select the “equals” block.
Once selected drag the “equals” block onto the “IF” statement block.

To ensure the micro:bit bases the IF statement on the random number, the value stored in the variable “Random” needs to be checked. The “Random” variable can be found from the variable option in the block menu.
The “Random” variable block is placed in the “equals” block to test the IF statement logic. If the “Random” variable contains a “2” the first option is selected otherwise the second or third option is chosen.

So the micro:bit can chose whether it is the second or third option the decision statement needs to be duplicated, by right clicking on the “equals” block, and place following the “else if” part of the “If” block.
Getting Started With JavaScript Blocks: Magic 8

Now the micro:bit has 3 possible options based upon the random number generated, the next step is to give the user an answer to their question. This is achieved by selecting the “show string” block from the basic option of the block menu and drag it onto the workspace.

For each “then” part of the “if” block a “show string” block is required. Drag 3 from the menu or right click the block once in the workspace to duplicate. The strings should be modified in order to read “yes”, “no” and “I don’t know”
Once the answer has been generated the final part of the program is to rest the micro:bit to display the number “8” again. Another show string block is needed and placed after the “if” statement.

Save the project and download to the micro:bit.