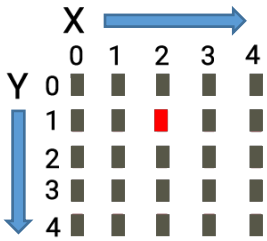
Name: _____

Loopy LEDs!

Professor Bob Brown – Kennesaw State University
Bob.Brown@Kennesaw.edu

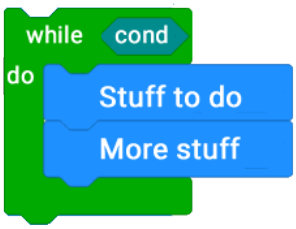
The Micro:Bit LEDs

The LEDs on the Micro:Bit are organized as a grid, five wide and five tall. Although there are five of them each way, they're numbered from 0 to 4. The rows are the X axis and the columns are the Y axis. Unlike graphs in math, the Y axis goes down as the numbers get bigger. We can name any LED by giving its number on the X axis followed by its number on the Y axis. The red LED in the image is 2, 1.



LEDs are turned on by the *plot* operation, which is in the LED bin, and turned off by the *unplot* operation.

Loops



In computing, a *loop* is a way of performing the same instructions more than once, possibly with different data. Looping is also called *iteration*.

The general form of a loop is shown in the picture. The part labeled *cond* is the *loop continuation condition.* The loop will repeat for as long as the continuation condition is true. For example, if the continuation condition were $x < 5$, the loop will repeat as long as the variable x is less than 5 and stop when that's no longer true. It is the responsibility of the programmer to make sure that x eventually becomes equal to or greater than 5. Otherwise, the loop will run forever. That's called an *infinite loop*, and that' usually a programming error.

The block language of the Micro:Bit has several kinds of loops. The *for-loop* changes a variable that can be used within the loop and stops when the variable reaches a given value. The variable is called the *index variable*. The *repeat-loop* runs a given number of times and stops. The most general loop is the *while-loop*, shown above.

KENNESAW STATE UNIVERSITY
COLLEGE OF COMPUTING AND SOFTWARE ENGINEERING

Last update: 2020-03-09

Light Those LEDs!

If we were going to turn on, then turn off each of the LEDs one at a time, we'd need 20 *plot* operations, one for each LED, and also 25 *unplot* operations. Using loops, we can do the same thing with only two plot / unplot operations.

Algorithm for the Top Row Only

- Set *x* to 0
- Turn on the LED at *x*,0
- Pause, then turn it off
- Add 1 to *x*
- If *x* > 4, stop.

Notice that this is an infinite loop. When we do this in MakerCode, we'll use a *repeat-loop* that runs only once, but place it within a *forever* block.

One other thing: All of this happens at computer speed, not human speed. We'll need a pause in there between step 2 and step 3 so that we can see the LED turn on.

All the LEDs

To light all the LEDs one after another, we need two loops. The *inner loop* lights the LEDs in one row, one after another, and the *outer loop* moves to the next row. We move to the next row by increasing the value of the *y* variable.

Algorithm for All LEDs

- Set *y* to 0
    - Set x to zero
    - Turn on the LED at *x, y*
    - Pause, then turn it off
    - Add 1 to *x*
    - If *x* > 4, stop
- Add 1 to *y*
- If *y* > 4, go to top.

(If you're reading this at home and you want to see the pictures of the programming blocks, remember that the slides are on line: drbrown.link/mcnair

Thought Challenges

Can you make the one-row algorithm go from right to left? *Hint:* Do not change x within the loop! You will need a new, temporary variable. You can do arithmetic on it, such as setting it to *x* – *something*. There's an absolute operation in the Math bin.

Can you make the LEDs move down one column, then down the next, instead of across the rows?

What Have We Learned Today?

- Looping, also called *iteration*, repeats the same actions, possibly with different values.
- There is more than one kind of loop.
- Looping is one of the important methods of program control.

The important methods of program control are:

- *Sequence:* Doing one thing at a time. This is the natural way computers work.
- *Selection:* Choosing which instructions to do next. This is how computers make decisions. A way to make decisions with MakeCode blocks is the *if-block*.
- *Iteration:* This is looping, repeating the same instructions more than once, possible with different values.

In MakeCode, *on-blocks* are a kind of selection, and the *forever* block is exactly the same as a *while-true* loop. The *on-start* block is just a set of instructions to be performed in *sequence* when the program starts.