



Merchant Venturers School of Engineering  
Outreach Programme

## **Micro:bit Space Invaders**

### **Block Editor - 1 Hour**

Created by

*Ed Nutting*

Organised by

[Caroline.Higgins@bristol.ac.uk](mailto:Caroline.Higgins@bristol.ac.uk)

Published on February 5, 2017

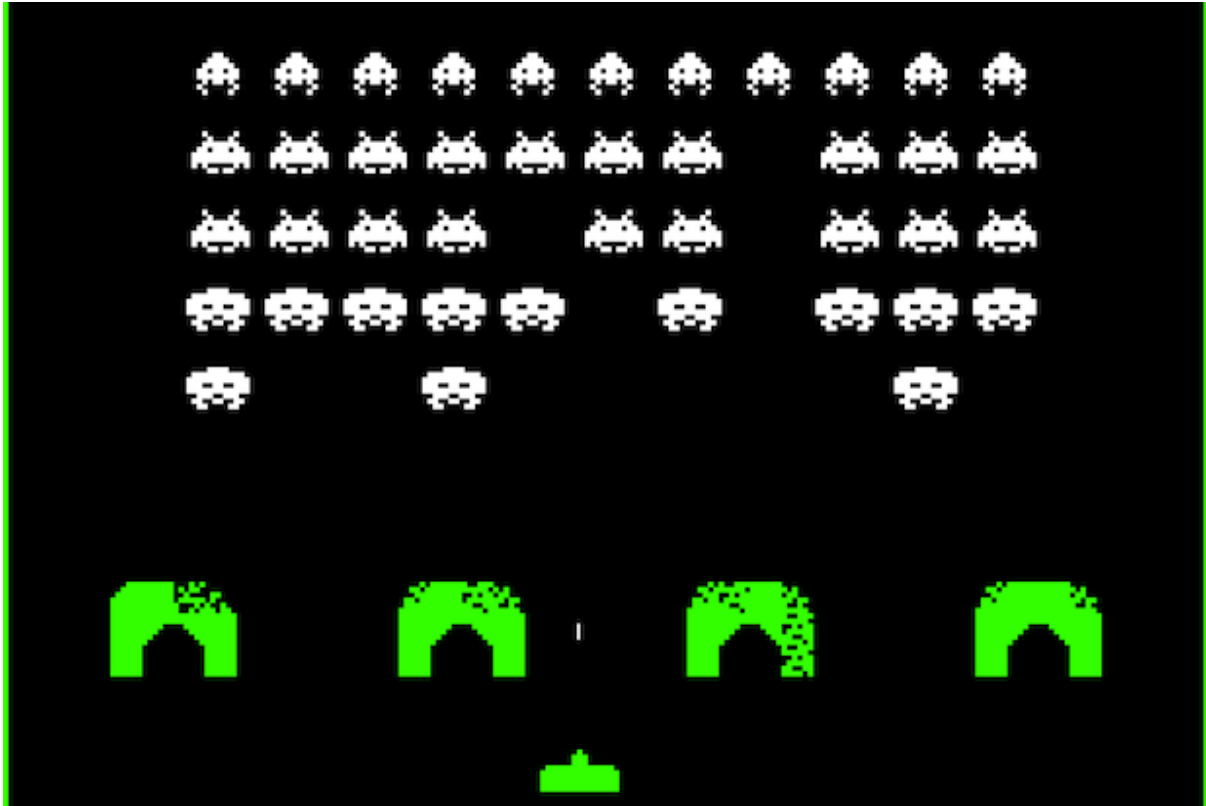
## Notes to Teachers & Helpers

- This workshop is intended to last 1 hour.
- This workshop is intended for ages 9<sup>+</sup> (years 5<sup>+</sup>).
- The content is intended to be learnt through teacher-directed, pair work, using this worksheet as a guide.
- The learning platform is the BBC Micro:bit (and the online Block Editor - very similar to Scratch).
- Students can use their own Micro:bit or we can provide a Micro:bit for the session.
- Students should be comfortable using the computer, including knowing how to click/drag blocks in the editor to join them together.
- This workshop teaches the following skills:

Items marked with an asterisk are directly relatable to the National Curriculum.

- Basic operations: Turning on/off LEDs
- Use of sensors/input: Buttons
- \* Conditional blocks (if-blocks)
- \* Basic logic: AND, OR, Equal-to
- \* Basic mathematical programming: Addition, Subtraction
- \* Basic game programming: Events

# 1 What is Space Invaders?



The classic game, Space Invaders!

Space Invaders is a classic game - one of the earliest ever made.

The invaders come down from the top and the green defender at the bottom has to move left and right, shooting the invaders.

If the invaders reach the bottom of the screen, the defender loses a life. Three lives only! The aim is to survive as many levels as possible.

## 2 What's in this worksheet?

Hi! In this short workshop we're going to try to introduce some of the concepts that Computer Scientists use every day to design everything from your games on your mobile to controllers for self driving cars.

Let's get started. Each section is made up of four parts:

**Actions** Stuff for you to do. They are highlighted in blue.

**Notes** Notes about important stuff you need to be aware of (and possibly remember!). They are highlighted in red.

**Questions** Questions you should try to answer. Sometimes you'll need to write things down; other times you'll need to build something in the game. They are highlighted in yellow.

**Ask a helper or the teacher to check your answers.**

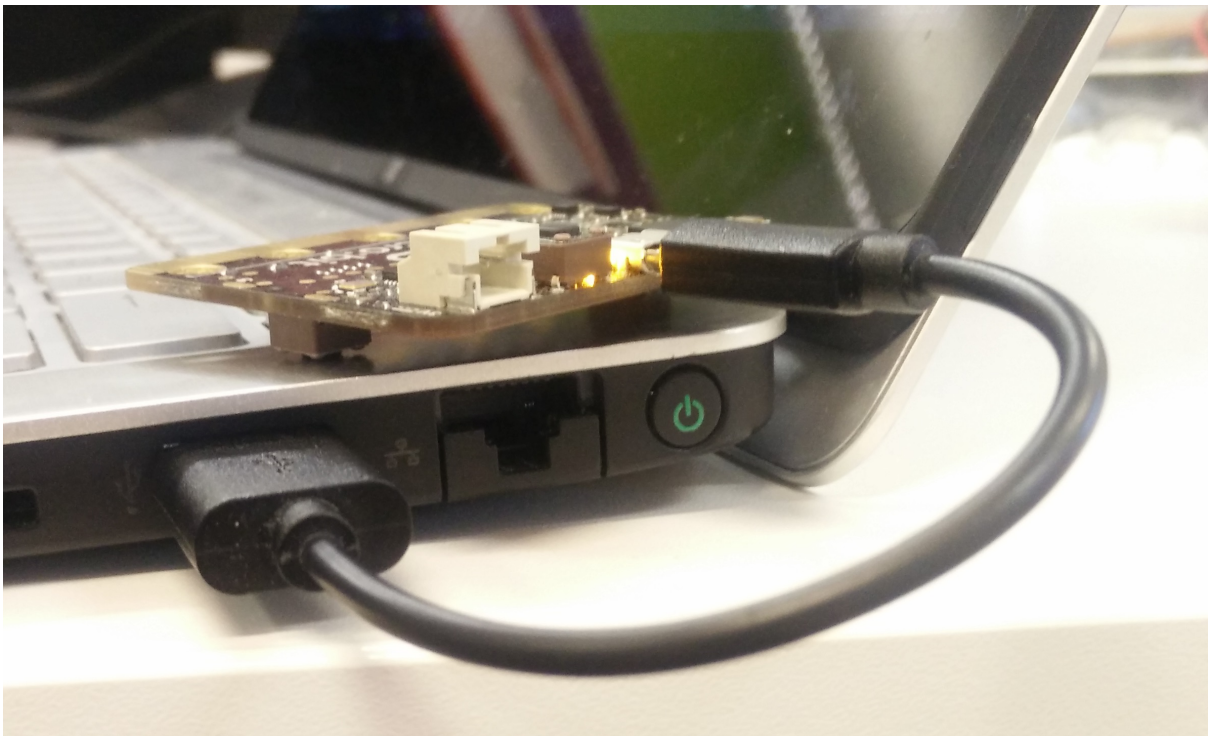
**Goals** Stuff you should have completed at the end of each section. They are highlighted in green.

We'll also write some information between parts and include plenty of screenshots to help you out.

### 3 Getting Started

#### Actions

1. Your workshop organiser has already plugged in your Micro:bit, opened the website, and created a new project for you in the editor.
2. In your project, you will see some code already exists. There should be 2 empty blocks: a “forever” block and an “on button A pressed” block. You will be filling in the code for these.
3. In your project, you will also see some big sections of code - don't worry, you don't need to edit this! These have been made for you so you can concentrate on making the game.



Micro:bit plugged into laptop using the supplied USB cable

#### Goals

Great! You should now have a new Block Editor project open and ready to start creating your game.

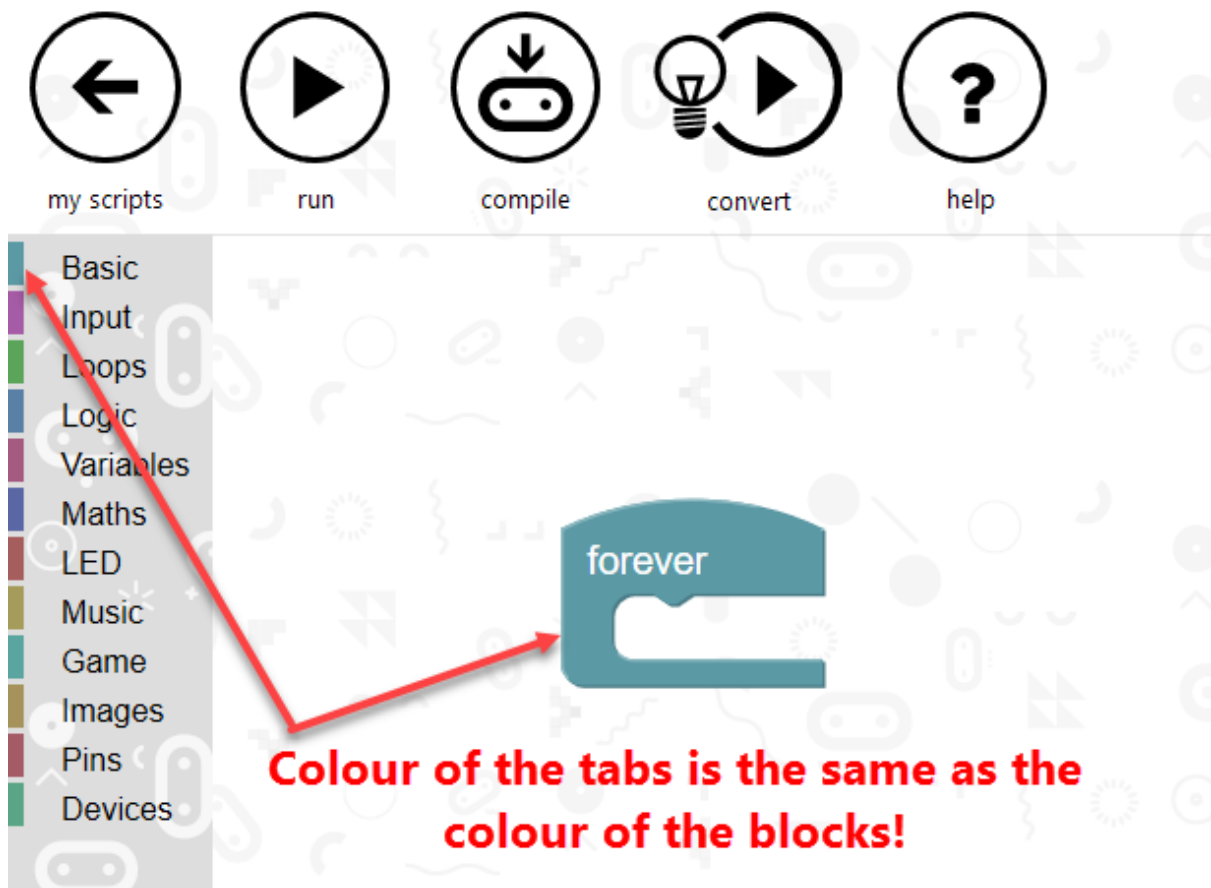
## 4 Lighting the LEDs

We're going to start simple, by just getting an LED to display on the screen. Then we'll see how we can make the LEDs look like our defender.

First, we need a loop. A loop is a block of code which repeats. Some loops repeat forever, others repeat for only a certain number of times. A "forever" loop has already been added to your project.

### Actions

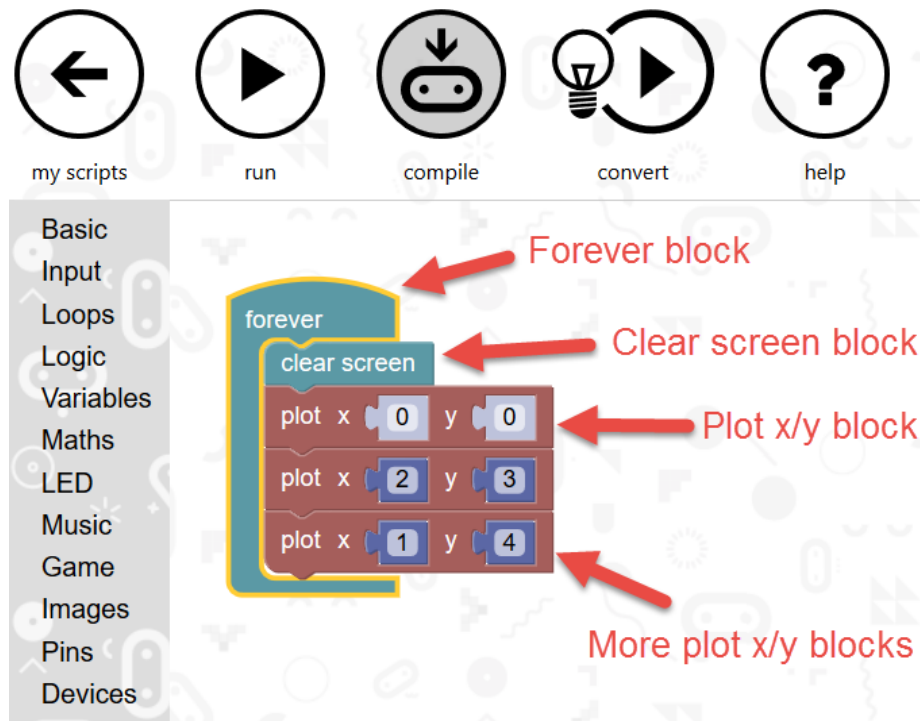
1. Zoom in on the "forever" block (use the plus and minus buttons in the bottom-right corner of your project or use the scroll wheel on your mouse).



Forever block in your project

Each time around the loop, we're going to clear the screen and then light up some LEDs

again. Later, we'll see how this allows us to move the defender and the space invaders around the display.



Plotting pixels

#### Actions

2. Click and drag a “Clear screen” block from the “Basic” tab to the inside of your “forever” block
3. Click and drag a “Plot x/y” block from the “LED” tab to just after the “Clear screen” block
4. Add more “Plot x/y” blocks and change the x/y values to plot more pixels. You can choose any number between 0 and 4 (including 4)!
5. Click “compile” at the top of the screen and download the file to your Microbit.

#### Goals

You should now see your pattern of LEDs on the display of your Micro:bit!

## 5 Defender Position

Now we know how to make the LEDs light up, we want to be able to use the bottom row of LEDs to display the defender. The defender will be represented by a single LED in the column the defender can shoot in.

We need to track the defender's position - 0 for left-most column, 4 for right-most and 1,2,3 for the columns in between. To keep track of which column the defender is in, we can use a "variable".

### Notes

A variable is a named thing which can be set to a whole number (0,1,2,3,.....) or a logic value (true/false). We can look at the value of a variable or change the value of the variable at any point in the code. A variable lets us store information for later.

### Actions

1. Remove all the "plot" blocks from your "forever" loop
2. Add a new "plot" block to your "forever" loop
3. Click and drag a "DefenderX" block from the "Variables" tab into the "x" value of a "plot" block
4. Set the "y" value of the "plot" block to 4





my scripts



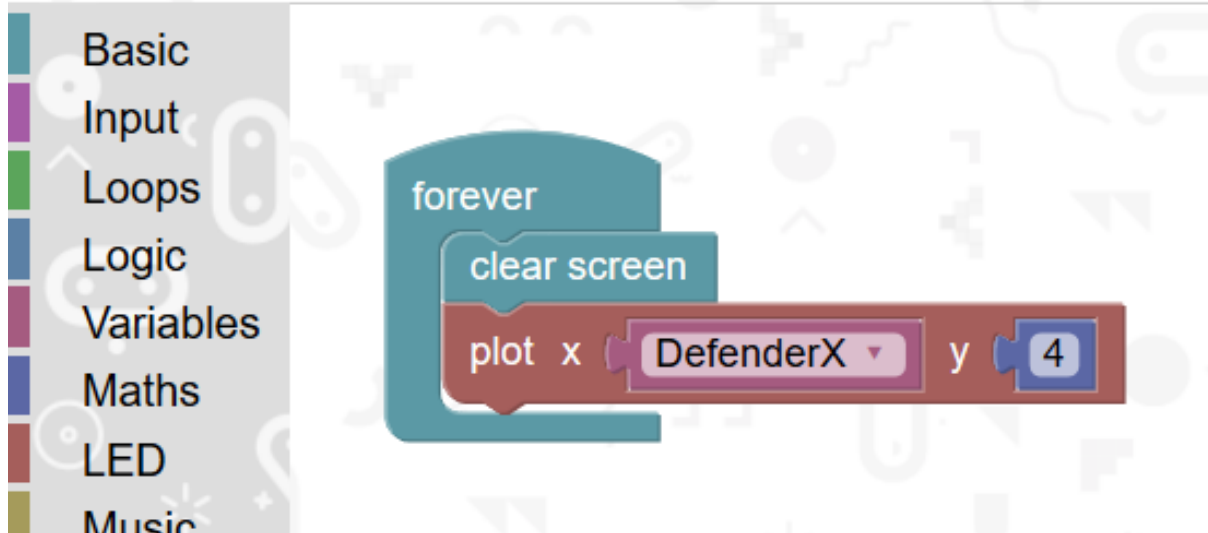
run



compile



convert



Plotting the Defender

#### Goals

Click compile and save your new code to your Micro:bit. You should see the bottom-left LED light up. If you tilt your Micro:bit left and right, you should see the defender move left and right along the bottom.

## 6 Invaders Positions

There will be three invaders, that will start at the top-left and work their way across as a block from left to right. When they reach the right hand side, they will jump back to the left again.

We will need to keep track of which invaders are alive and which are dead. We can do this using three variables named “Invader1”, “Invader2” and “Invader3”. If an invader variable is “true”, it will mean the invader is alive. If it is “false”, the invader is dead.

We will also need to know where the invaders are. We will use two variables for this: one for the X-location (column number) that the left-most invader is in, and one for the Y-location (row number) that the invaders are in.

We need to plot the invaders - i.e. to display them on the LEDs. We can do this by using the current location of the invaders and whether each invader is alive. If the invader is alive, we turn on the relevant LED. Otherwise, we leave the LED off.

### Actions

1. Add blocks to your game's code (extending the section we made earlier for displaying the defender) so that it matches the screenshot on the next page.

### Notes

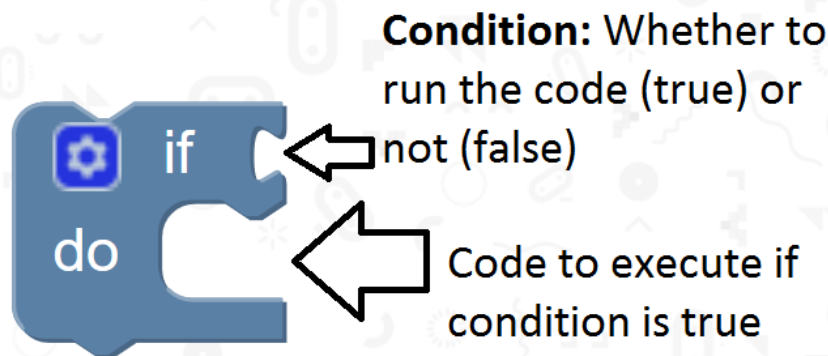
Remember, the colour of the blocks matches the colour of the tab on the left you can find them in!

### Notes

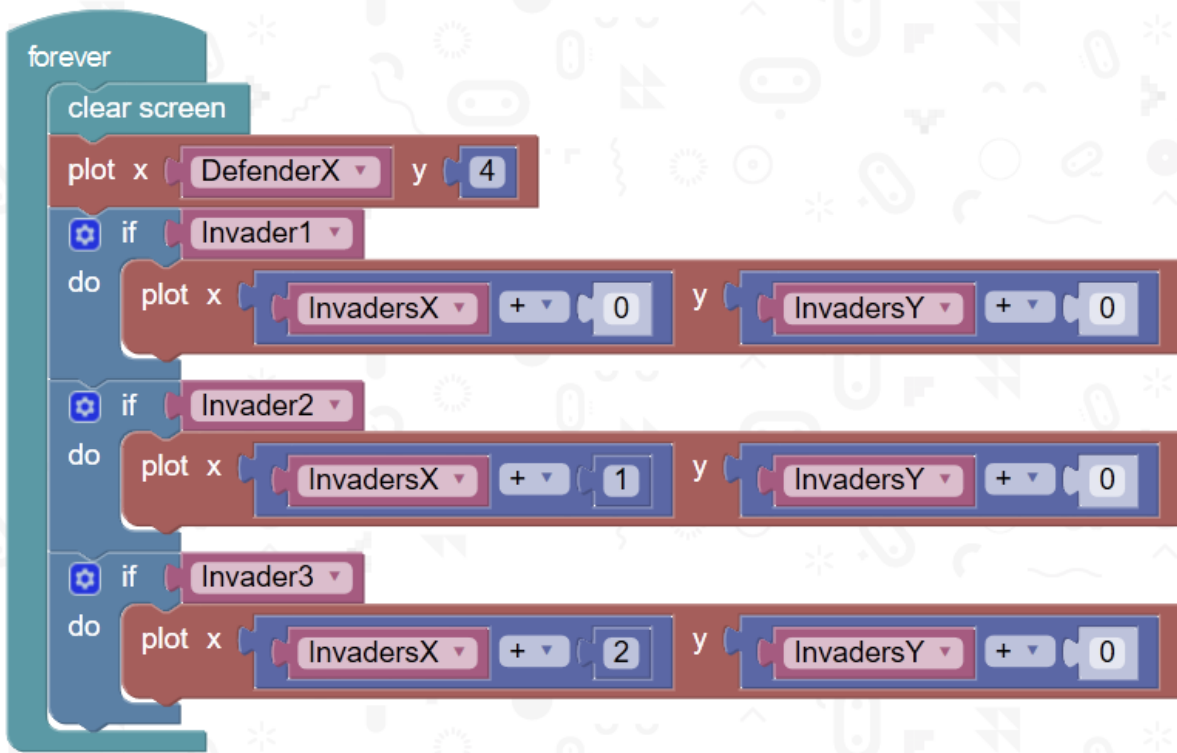
An “if” block is a decision block (often called a Conditional block). You test something, and if that test passes, then the code inside the block runs. Otherwise, the code inside the block doesn't run and the programs goes to the block after the if block.

### Notes

The “test” the if-block uses is called the “condition”. It is a logic equation. For example, “is five greater than four” is a condition (to which the answer is always “yes” - usually called “true”). If the condition is “true” the code inside the if block happens, otherwise the condition is “false” so the code inside doesn’t happen.



An if-block: Condition determines if the code inside runs.



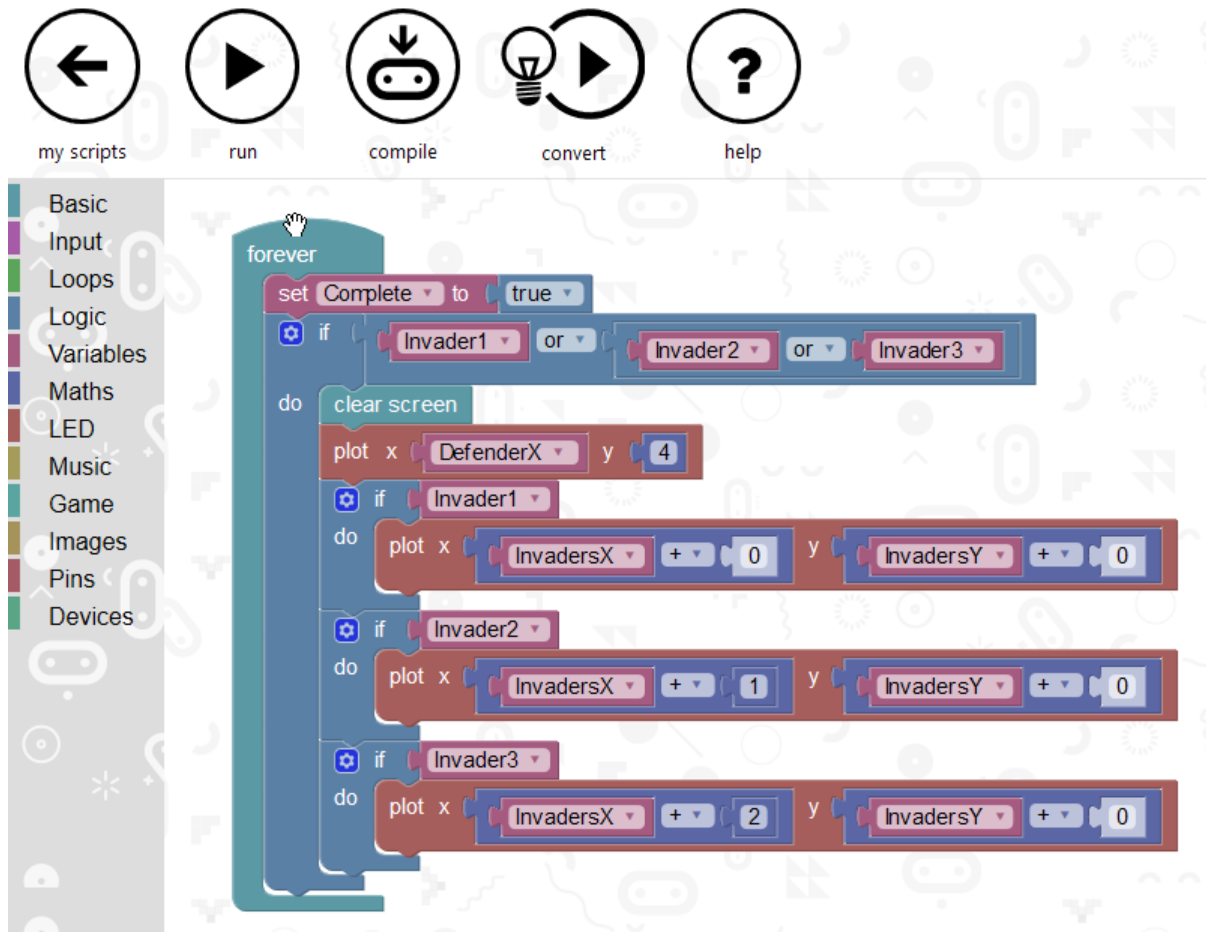
Code for displaying the invaders and the defender

### Goals

Compile and save your game to your Micro:bit game to your Microbit. When the device powers on, you should now see the defender moving as before and the three invaders (staying still) in the top-left corner. (We'll make them move in the next section!)

## 7 Making the invaders move

Half our game is nearly complete! We just need to add a couple of extra blocks. Edit the code within your “forever” block to make it match the code in the screenshot below.



Half our code is complete!

The “Set Complete” block sets a variable called “Complete” to “true”. This enables the other code to know that you’ve completed the Plotting section of the code.

The extra “if” block prevents the invaders and defender from being drawn at the start and end of the game.

### Goals

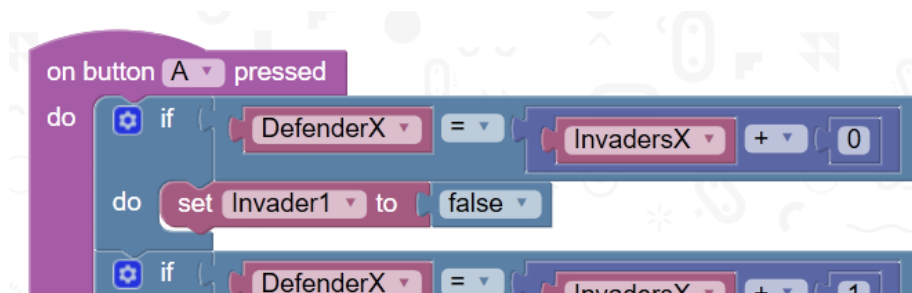
Compile and save your code to your Micro:bit. You should now see the word “Go!” go across your screen and then the game begins.

## 8 Shooting the invaders

This is the last part to our code - shooting the invaders! We're going to program our game so that, after the game has started, when button A is pressed, the invader that is in the same column as the defender will be shot.

### Actions

1. You already have an “on button A pressed” block from the “Input” tab in your project.
2. See if you can work out what if-blocks and conditions you will need to add to the “on button A pressed” block, to determine which invader the defender is shooting at. Hint: A screenshot of a small part of the code is shown below.



See if you can work out the last section of code!

### Notes

Hint: Think about the position of the defender (DefenderX) and the position of the invaders (InvadersX). You will want to compare them somehow!

### Goals

Test your new code. Does it work? If not, keep trying - code, test, repeat - that's how real computer scientists and software engineers work.

### Actions

3. Try to get a working version of shooting the invaders. If you're having problems, ask a workshop helper.

### Goals

Congratulations! You've finished making the game!

### Questions

1. Are there any improvements to the game you can think of? Have a go at programming them. Remember: code-test-repeat.