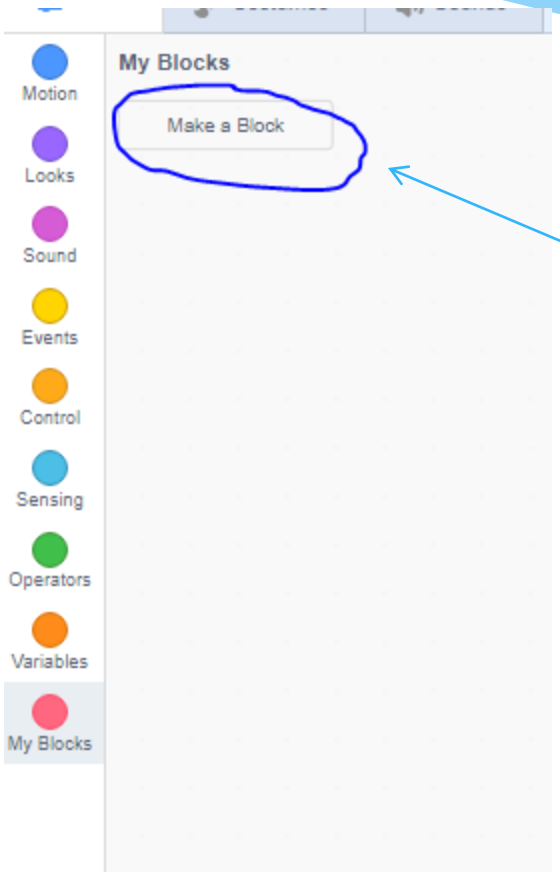
Pen Platformer Part 1 – Creating the arena with My Blocks

Vineet Srivastava

In this lesson, we will …

* Learn how to create and use our own SCRATCH blocks.
* We will learn *why* we create our own blocks and *when* they are helpful.
* To bring these concepts to life, we will build an arena for a platformer game with multiple levels using My Blocks.

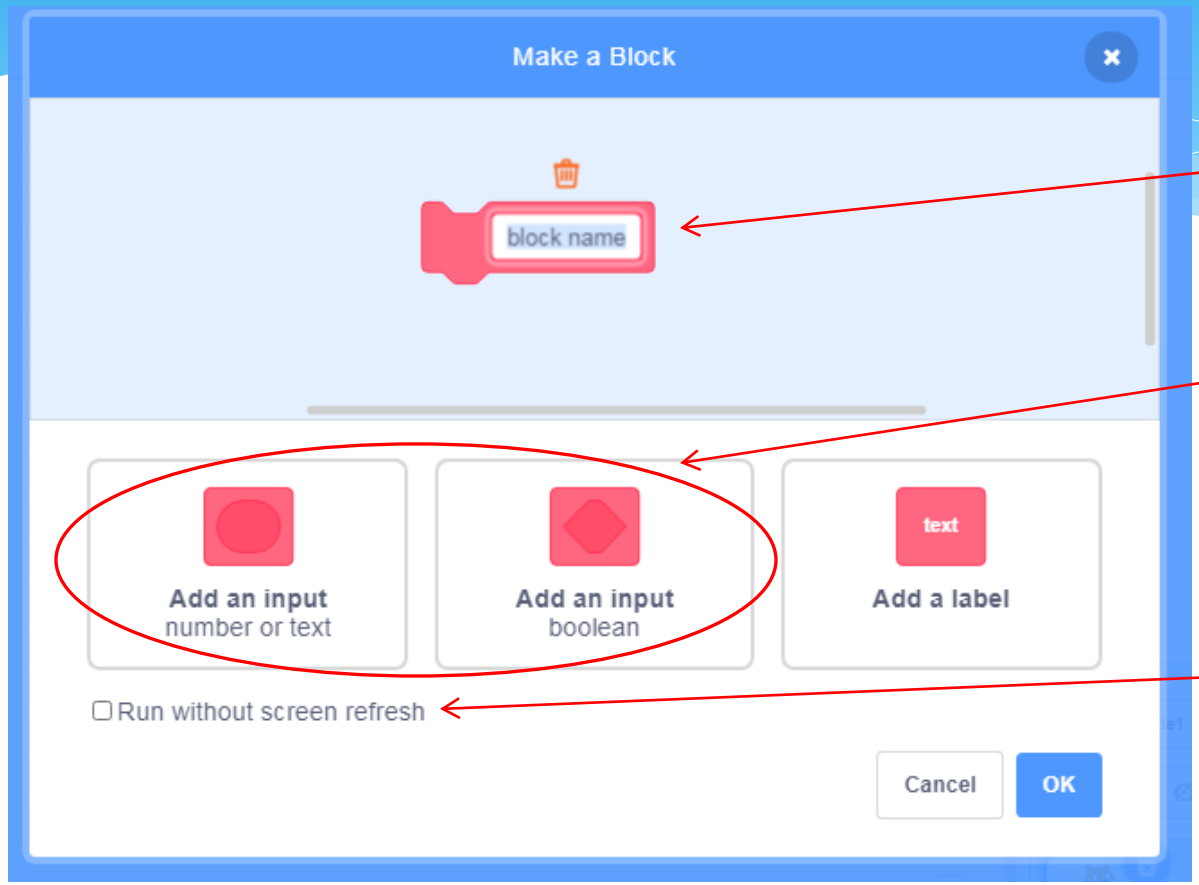
To create our own block



Click on ‘Make a Block’

WIBYTE

When you click ‘Make a Block’



Name of your block (Use a meaningful name).

Inputs to your block

Ignore this for now, we will come back to this later.

We get a screen like this … This allows us to ‘define’ our block.

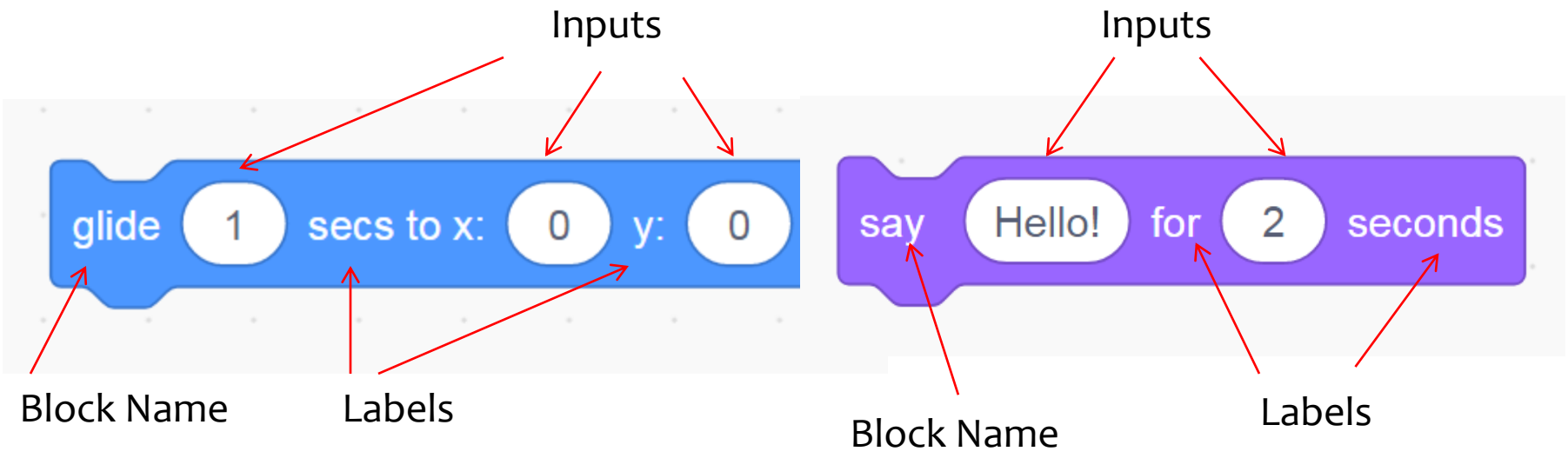
WIBYTE

Two types of inputs

* Notice that there are two types of inputs
  * Number or text
    * This implies a regular variable, which holds a number or a text (string), e.g., 1, 2, 3, … 4, ‘name’, ‘place’ etc.
  * Boolean
    * A special type of variable which takes on only two values – TRUE or FALSE

Add a label

* A label is like a text in the block definition which makes the function easier to read.
* Let's first take some examples from existing SCRATCH blocks

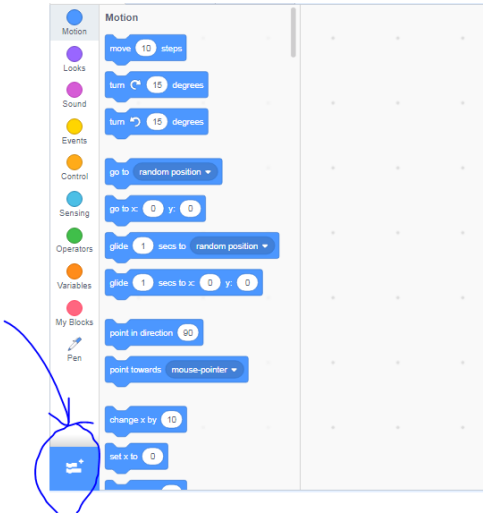


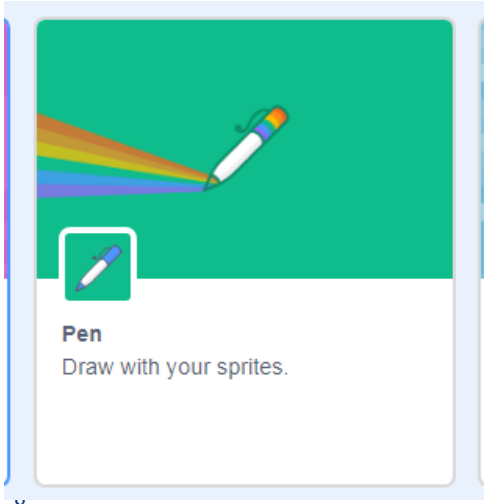
What can we do from inside MY BLOCK

* We can pretty much use all the SCRATCH existing blocks to create a My Block.
* We can *also* use any other block that we have created as a My Block.

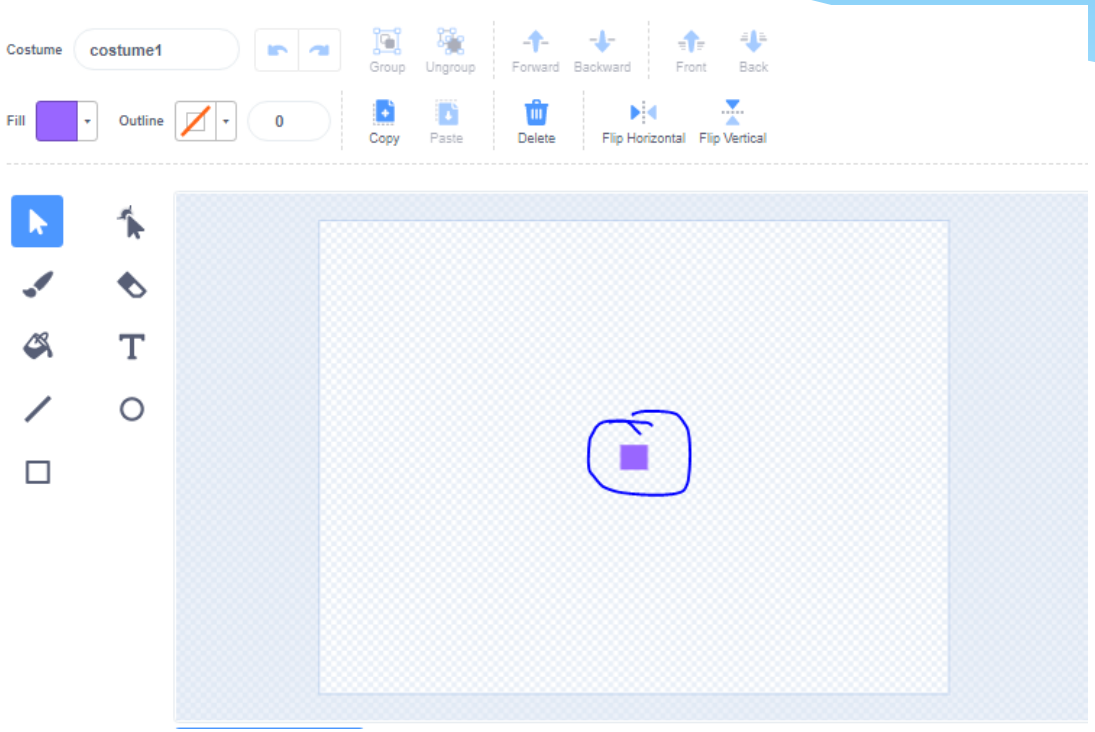
Create My Block to Draw a Line

* Say we create a block to draw a straight line from (x0, y0) to (x1, y1).
* First thing, add PEN extension.



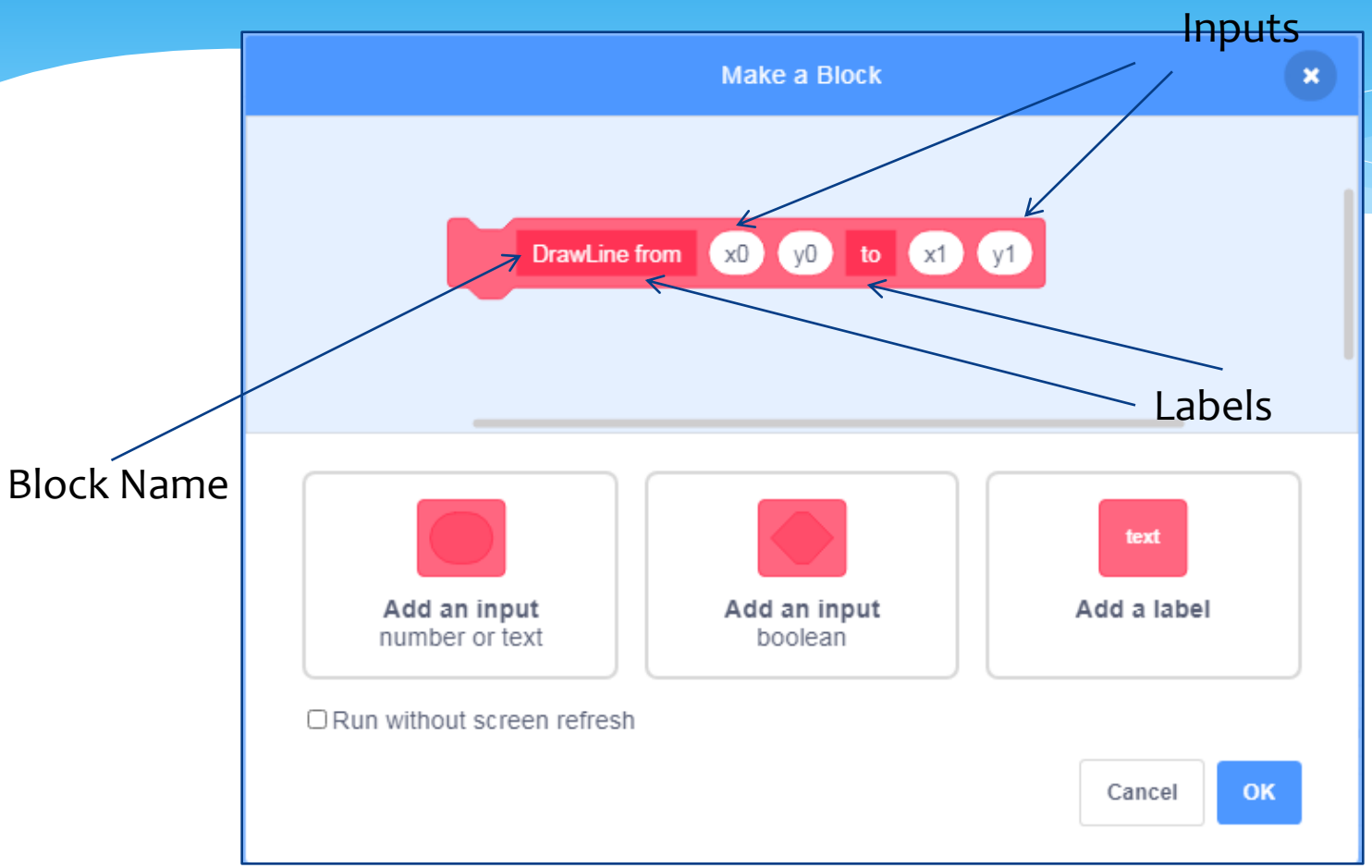


Create a costume for our sprite



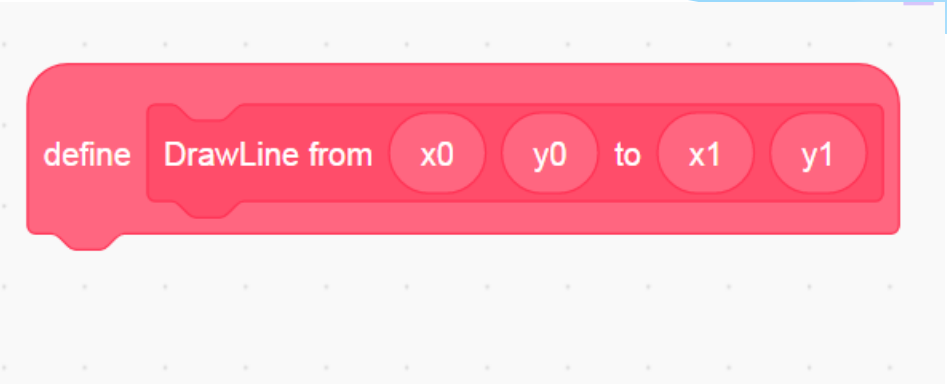
* Just a small DOT.
* This really does not matter, as we are mostly going to keep this hidden in our code.

Create a block to Draw a Line!



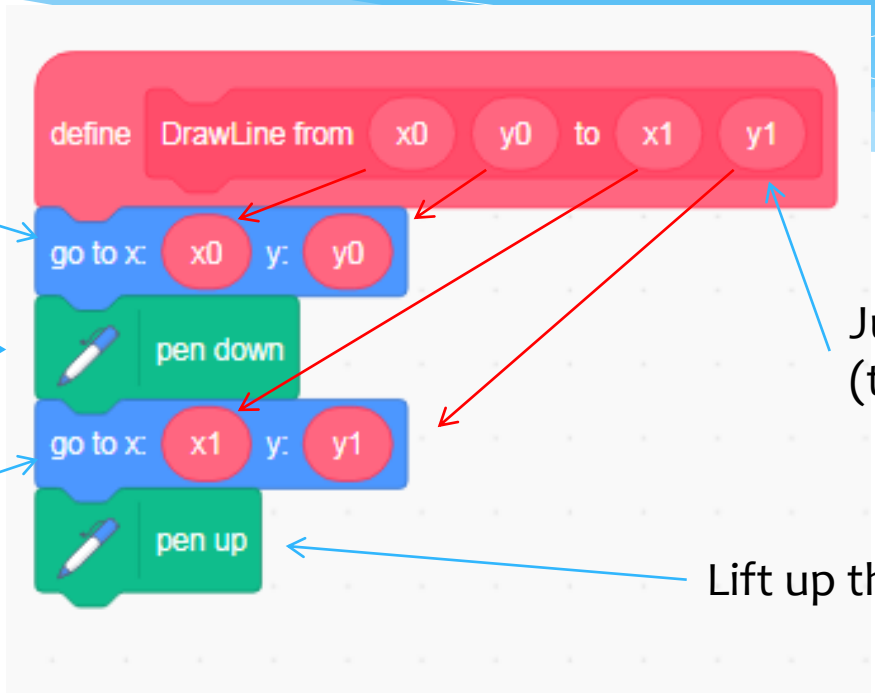
When you click OK

* You will see something like this:



* Now we will write some code to ‘define’ what this block does.

Let's write some code to DRAW a line!



Go to Point 1 (x0, y0)

Start Drawing

Go to Point 2 (x1, y1)

Just drag these (the inputs)

Lift up the pen

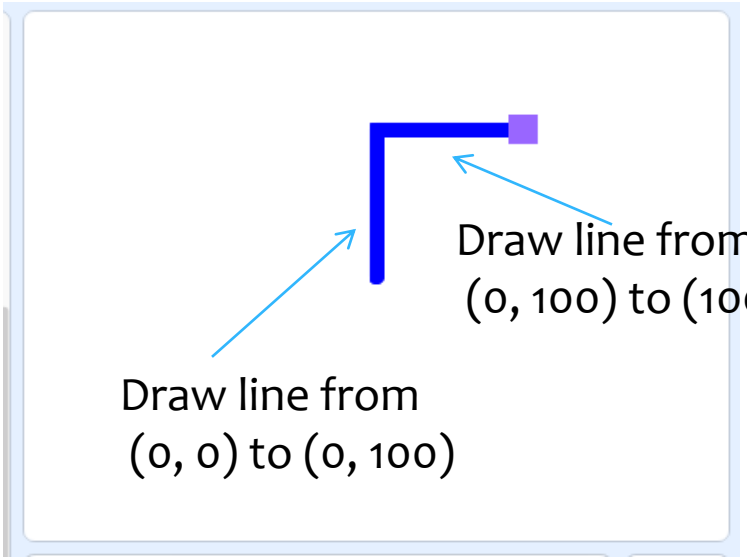
With this, we can draw a line starting from *any* (x0, y0) and ending at *any* (x1, y1) – We just need to provide these as inputs.

Now see this code …

* We call the block DrawLine twice, each time with different inputs.

Code

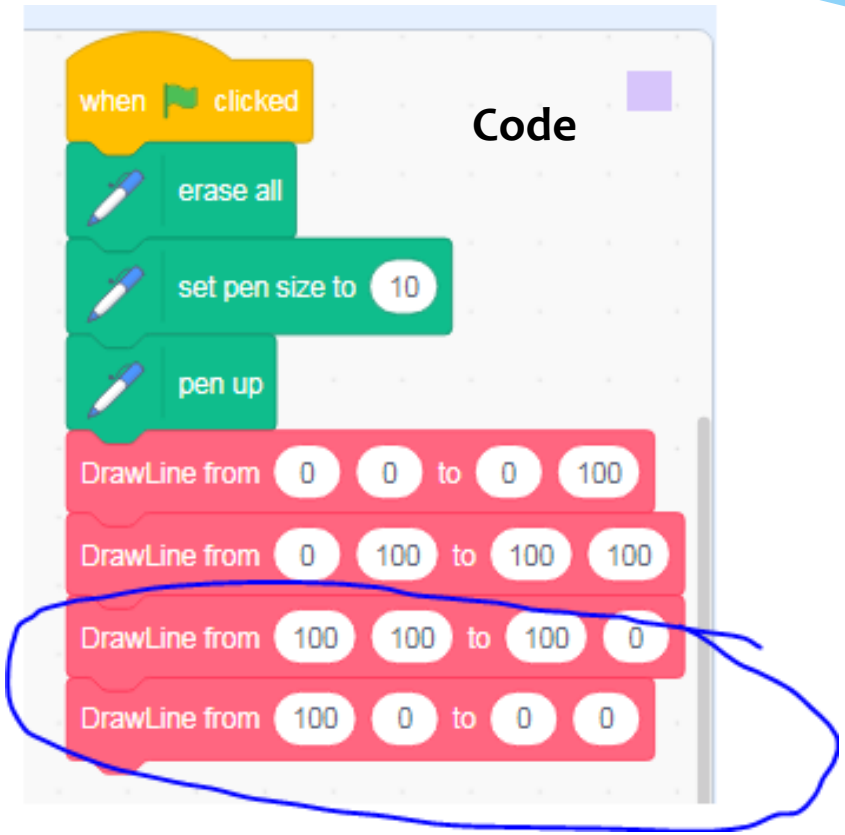
Screen Output



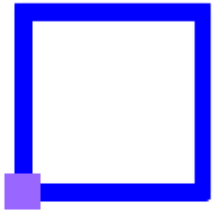
In the first call for DrawLine, the inputs are x0 = 0, y0 = 0, x1 = 0 and y1 = 100 .

In the first call for DrawLine, the inputs are x0 = 0, y0 = 100, x1 = 100 and y1 = 100 .

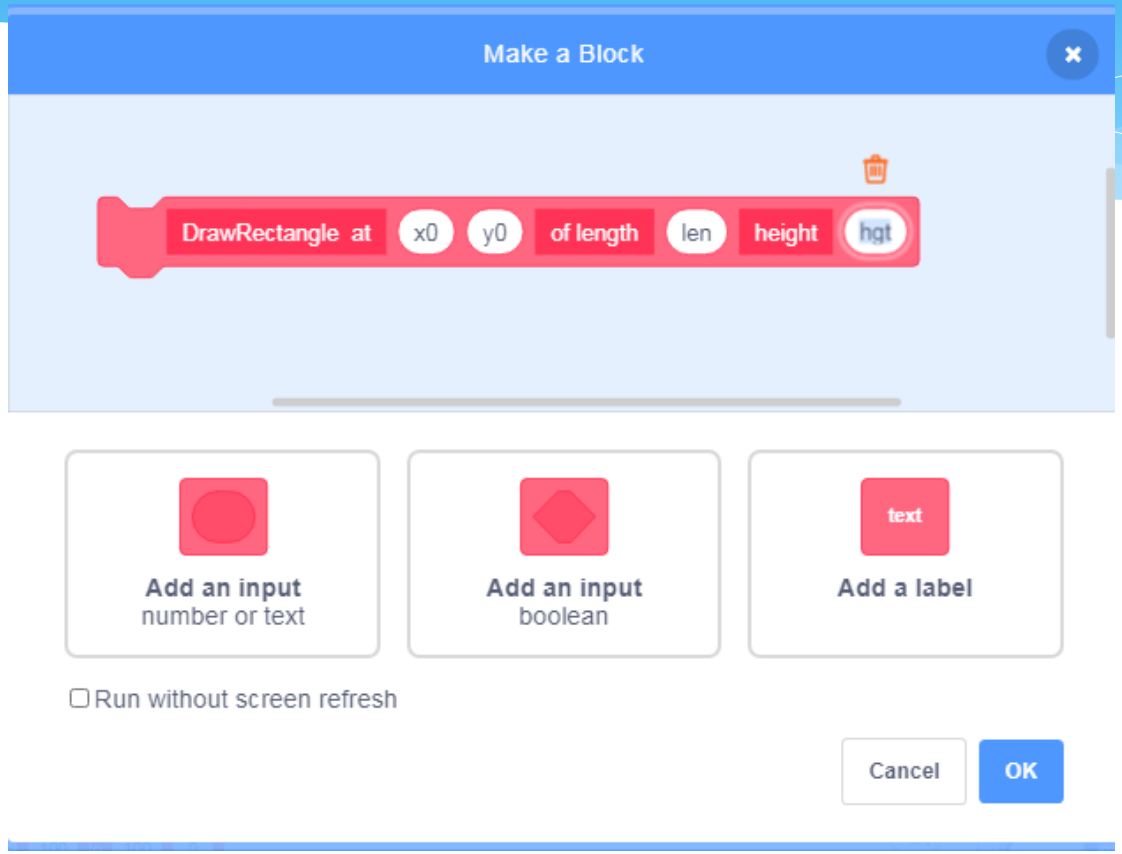
Add two more lines to complete the square!



Screen Output

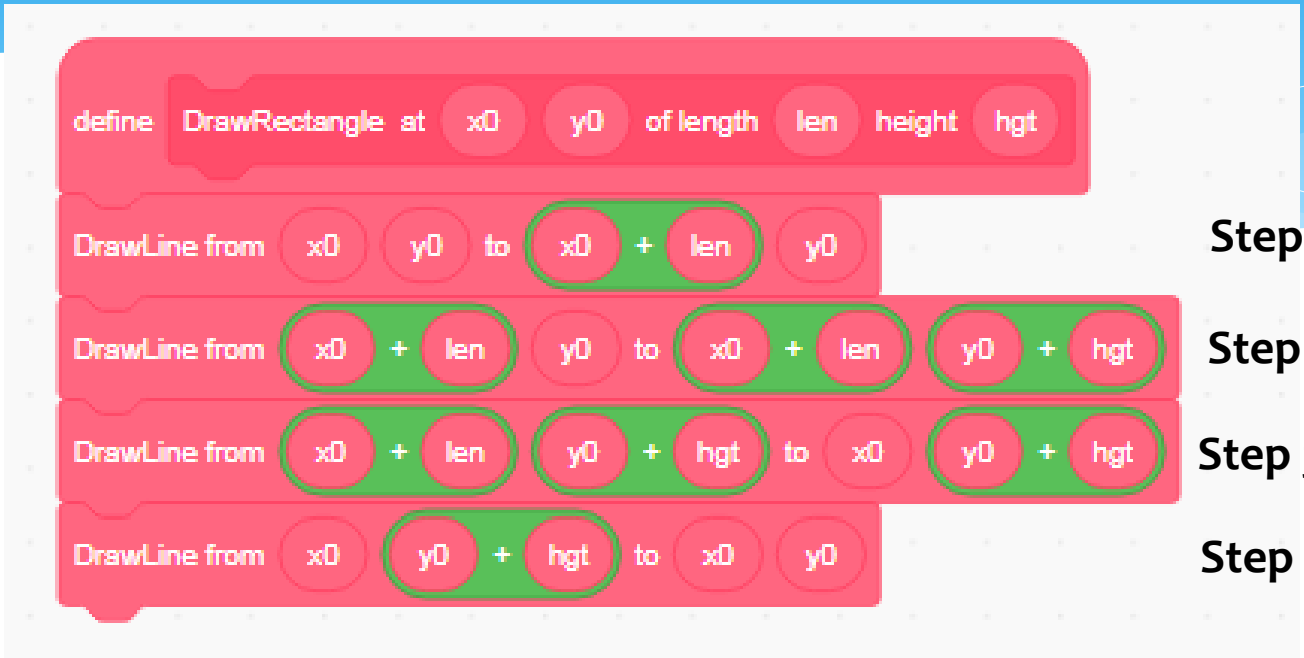


Use MY Block on ‘top’ of another MY BLOCK – Define Draw Rectangle

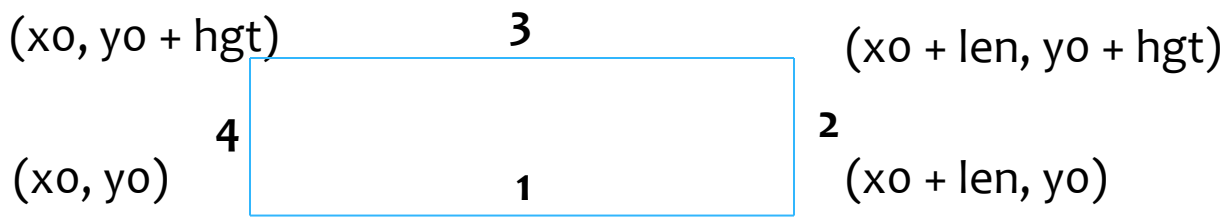


With this block we want to draw a rectangle at (xo, yo) of length ‘len’ and height ‘hgt’

Let's give code to DrawRectangle



See this code very carefully. We are drawing 4 lines, in the order as shown below:



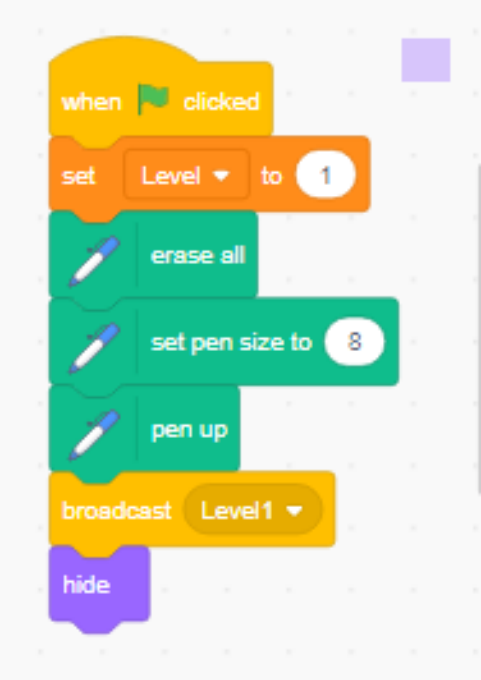
But is this all an overkill?

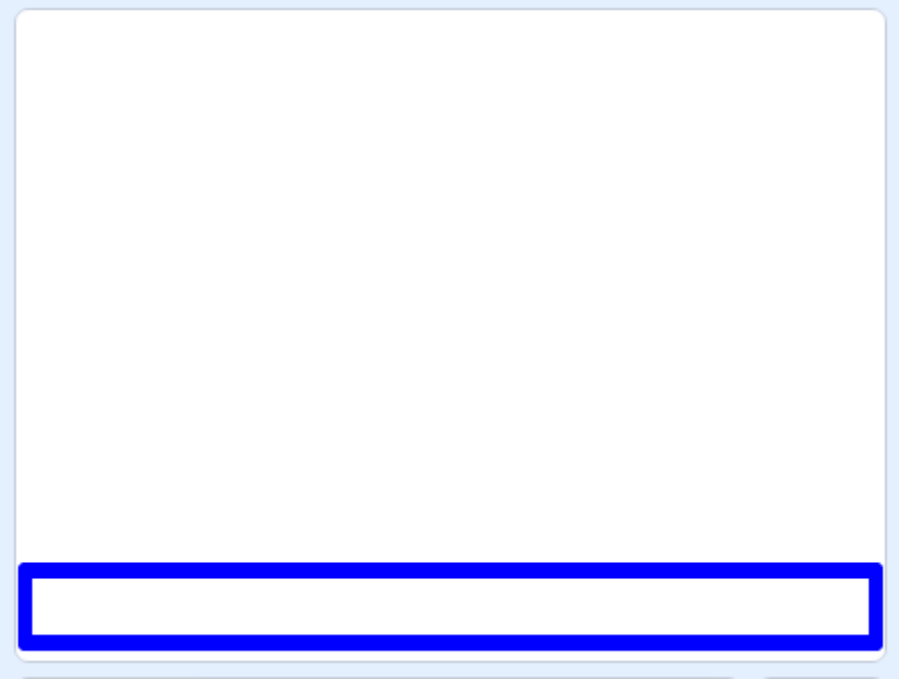
* Why can we not just draw these without MY BLOCKS
  * Answer is, you can, but creating a ‘MY BLOCK’ gives you the **flexibility**.
* However, one has to strike a balance – do not overuse these.
* Use My Blocks for tasks that you need to do **again and again and with different parameter values**.
* Sometimes My Blocks can help to organize the code better and make it more readable.

Back to the PEN Platformer

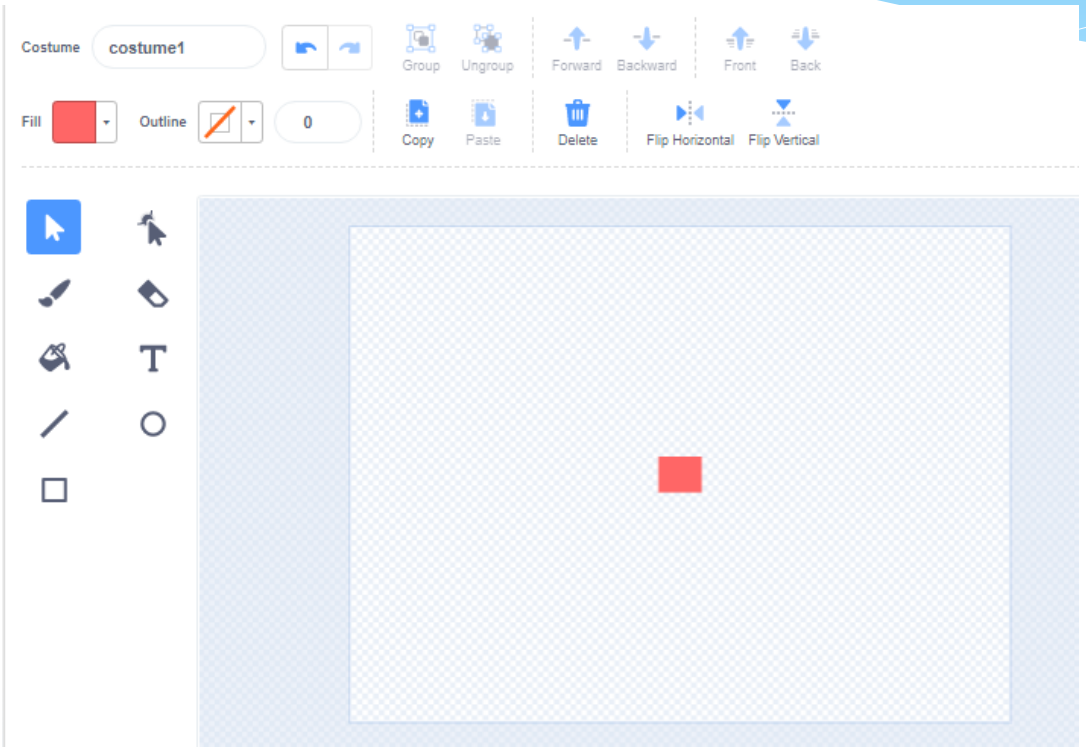
* In the PEN platformer, we use the pen blocks to draw the playing arena.
* We want the playing arena to 'change' every time a new level is reached.
* Basically, the HERO sprite moves from LEFT to RIGHT.
* On the way, it has to cross some obstacles. It may even fall down!
* But if it does reach the end of the screen, it enters the next level.
* In this case, it has to deal with another set of obstacles etc.
* We will use the blocks we have developed so far to create these arenas

Level 1 – Warm Up (Just 1 Rectangle)



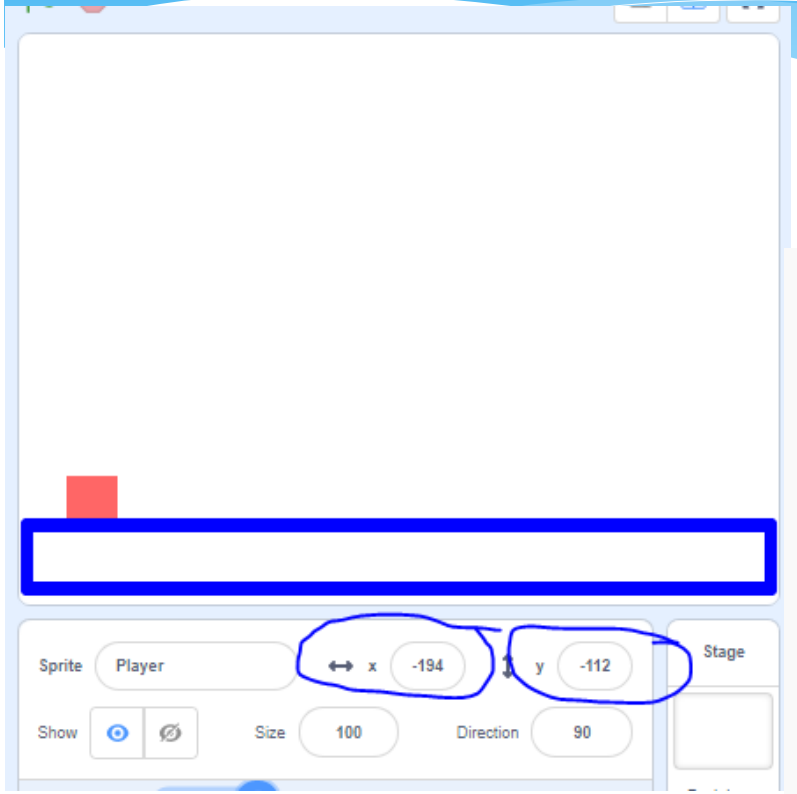


Create a HERO sprite! – A simple player

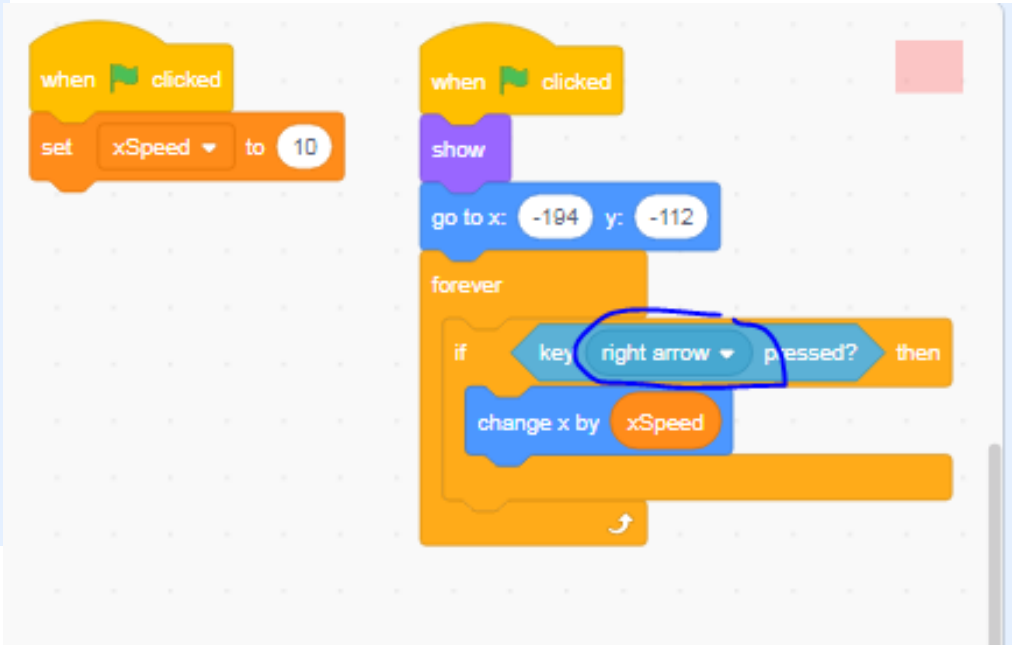


We will treat this small dot as a player.

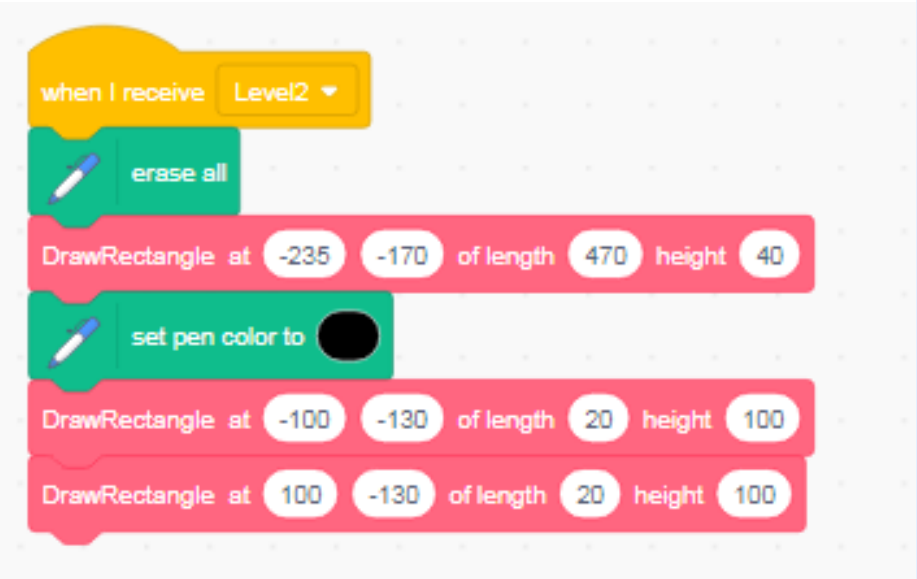
Player Movement

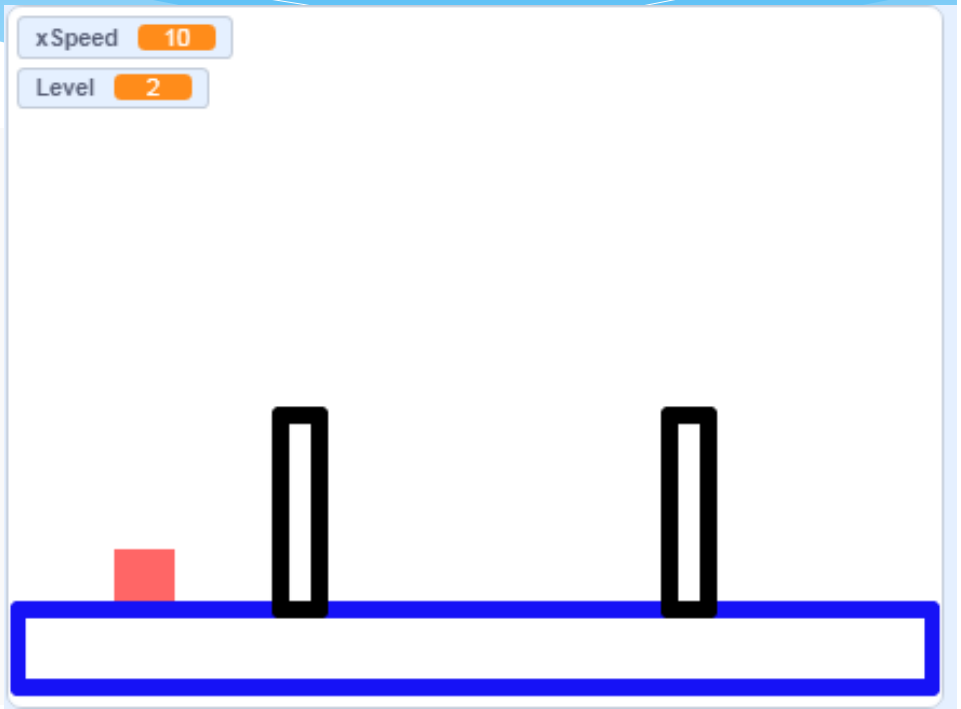


Place the player. (For now, just drag the player so that it is nicely placed on the platform and note the x and y values.)



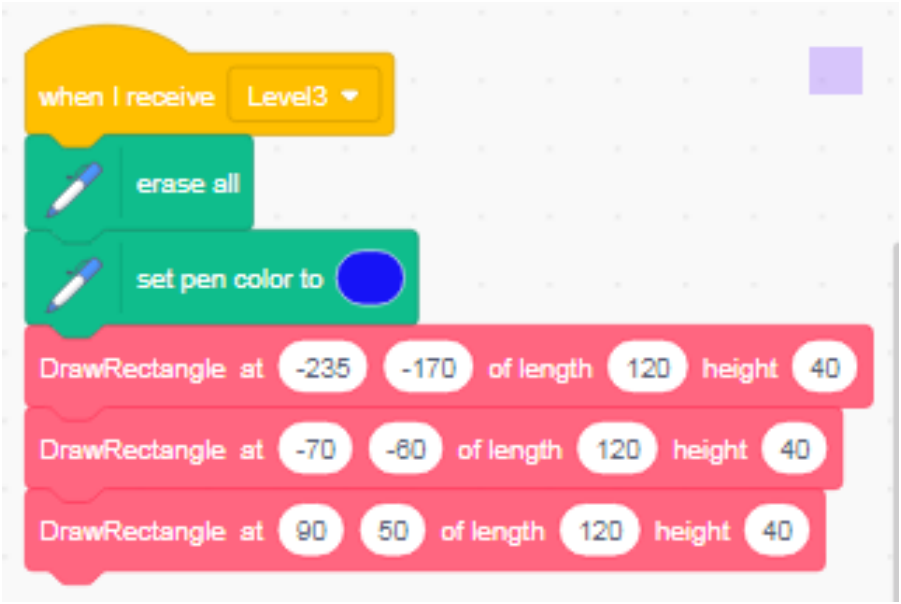
Level 2 – Some Obstacles

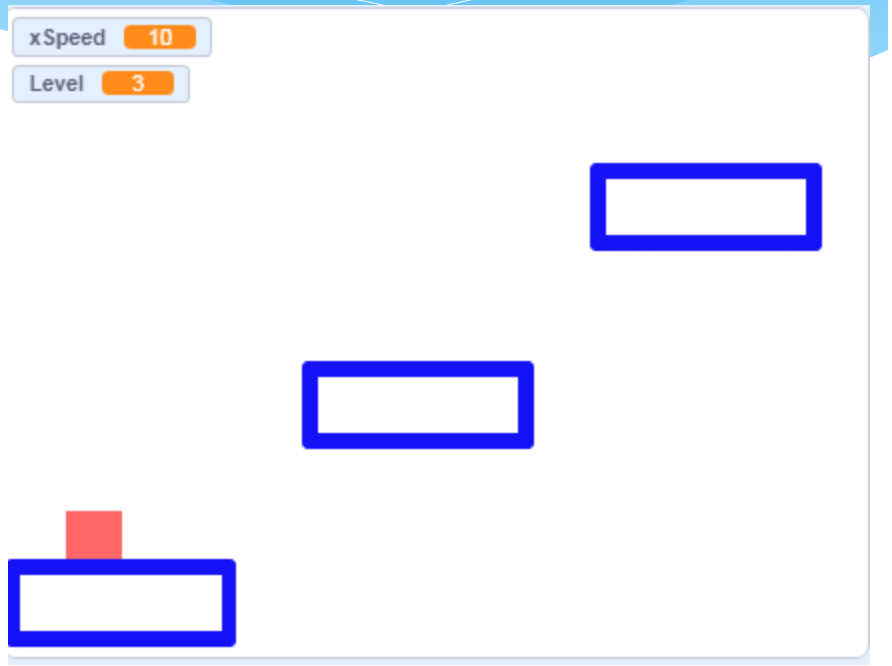




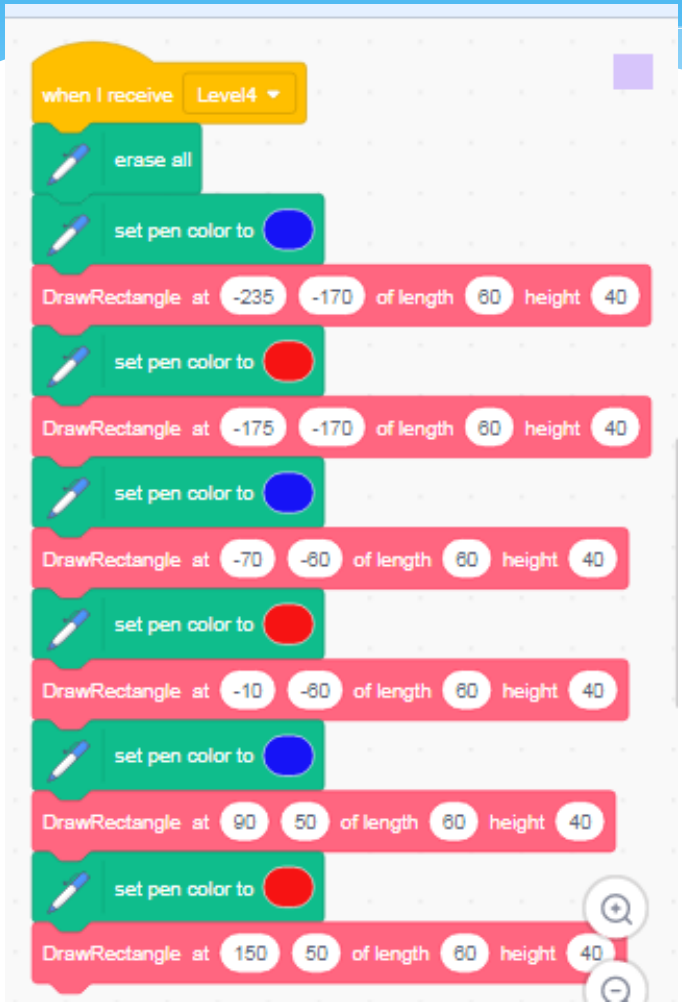
This 'When I receive' is placed in the 'creator' sprite. Level 2 is Broadcast from the Player Sprite – we have covered this code at just one place in a later slide. (Develop the two codes together. Add one level, create its arena and then create next level and so on.)

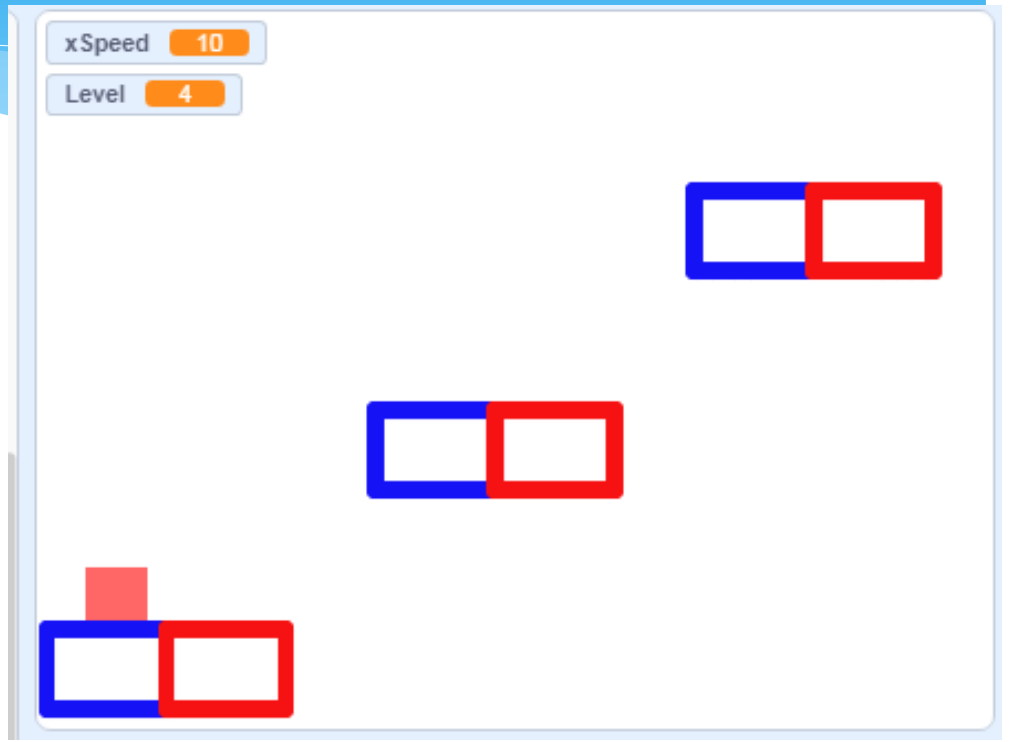
Level 3 – Platforms



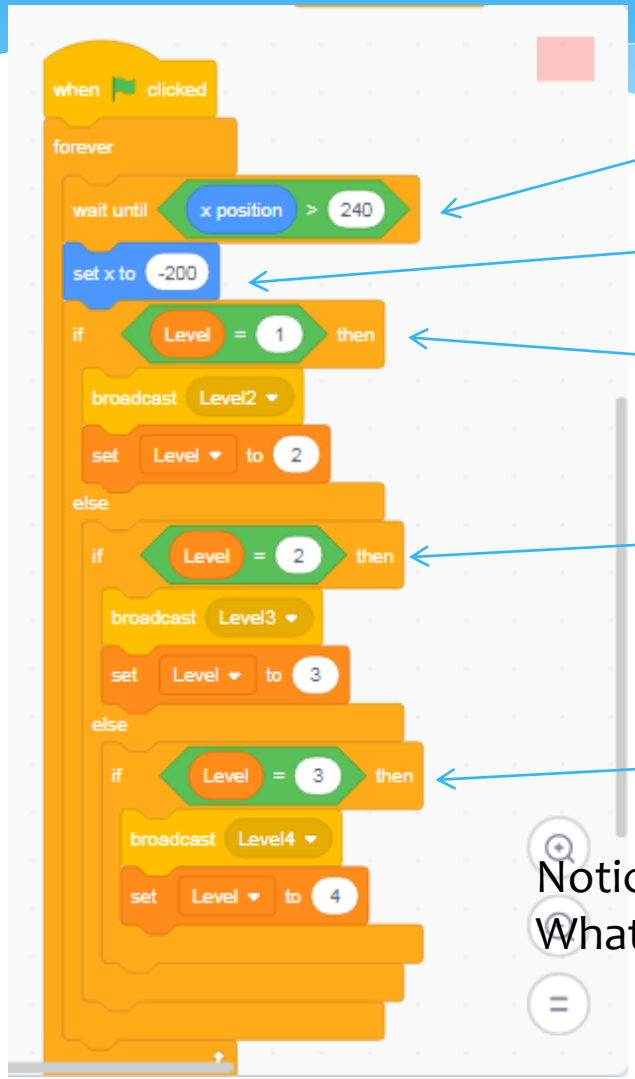


Level 4 – 2 color Platforms





Moving to Next Levels



Wait until (x > 240) to go to next level

On starting Next Level, set the x to -200 again

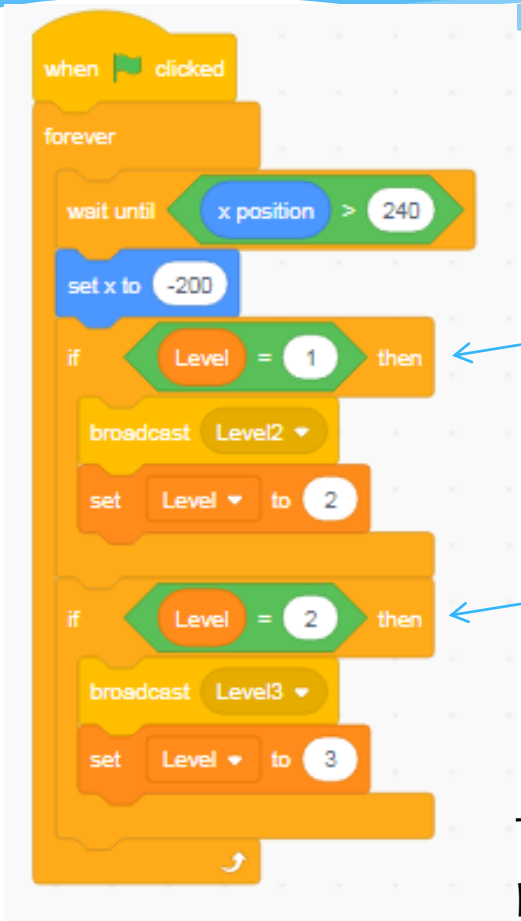
If Level is 1, move to level 2.

Else, If Level is 2, move to level 3.

Else, If Level is 3, move to level 4.

Notice very carefully the use of 'if then else' in this code. What will happen if we did not do so – see next slide.

What is the problem in this code?



If Level is 1, move to level 2.

If Level is 2, move to level 3.

Think carefully. Will this code work? What will be the problem? HINT: Will you ever go to Level 2?

And you are all set

* In this lesson we learn about using MyBlocks to create blocks that otherwise did not exist in Scratch.
* As a practice, we created some simple blocks and used them to create an arena for the PEN platformer game.
* With this you are all set for your independent activity, Pen Platformer – part 1.
* Use this opportunity to get very comfortable with the notion of My Blocks.
* Enjoy!

Extra Innings

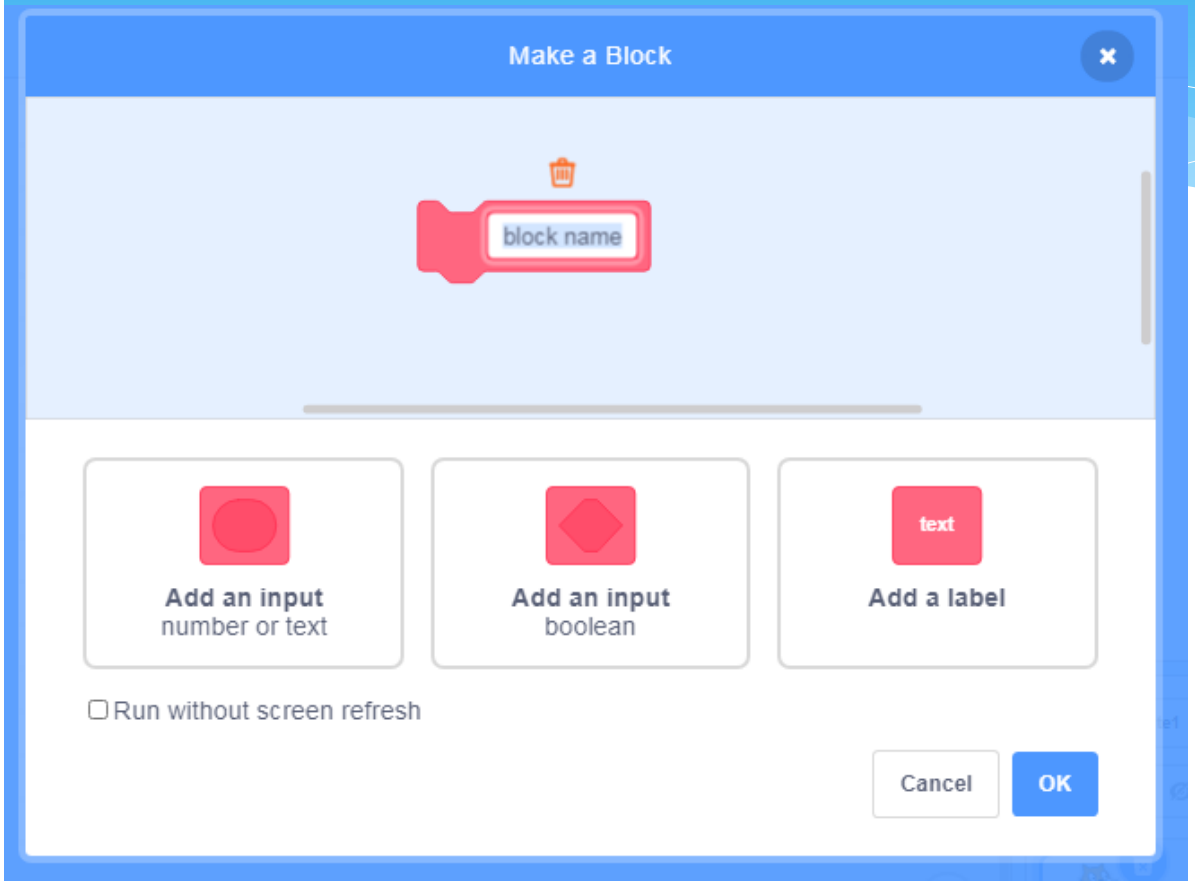
Ideas to spice up the game!

* Be creative.
* Imagine what all kind of obstacles and paths you may want to create for your game.
* You may want to be imaginative with the kind of obstacles you put.
* Use My Blocks creatively to create these!

My Blocks or Broadcast!

* We have seen the ‘BROADCAST’ also helps us in organizing code by putting a ‘repetitive’ action into a separate BROADCAST.
* But with My Blocks, we can also pass in arguments – this can allow us to control the parameters of this action.
* Unlike Broadcast, My Blocks can work in a ‘TURBO MODE’, that is, without screen refresh.

Running without screen refresh



Recall, there is an option called 'RUN WITHOUT SCREEN REFRESH'.

This is useful if there are loops like 'repeat' inside your block. With this option, the loops run faster – without the added delay caused by screen getting refreshed.

Hence, if you are making a circle, using repeat block, this may be helpful.

See these examples:
https://scratch.mit.edu/projects/88005579/editor
https://scratch.mit.edu/projects/10567475/editor

WIBYTE