LESSON 1: Flappy Frog

Stages

- Create a flappy sprite
- Show the spirite in a fixed poistion
- Have the sprite fall
- Have the sprite rise when a button is pressed
- Create the tubes sprite
- Move the tube sprites
- Detect collision between flappy and tube
- Increament the score when going through the pipes
- Improvements

Create a Flappy Sprite

Select the default Scratch cat sprite and click the delete icon

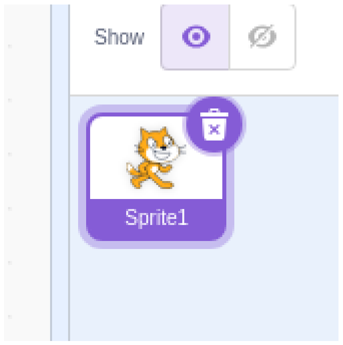


Figure 1: Remove the deault sprite

Click this icon on the bottom right of the page.

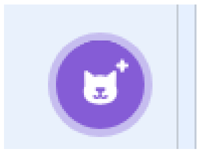


Figure 2: Choose Sprite

Select a sprite from the list (I chose `Wizard-toad`)

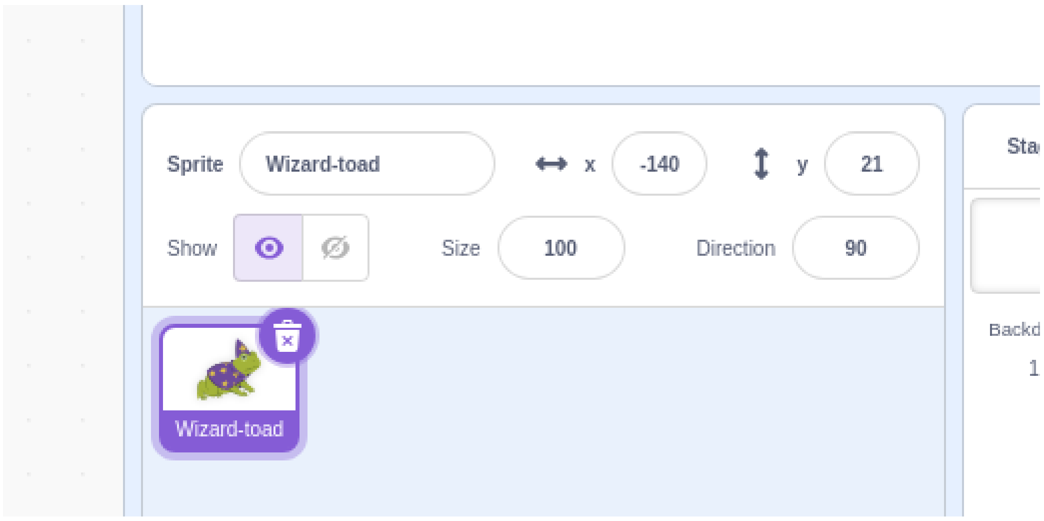
You are able to resize and see the position of the sprite from the `Sprite info` panel.

Show the spirite in a fixed poistion

Drag the sprite to roughly where you want it to be when the game starts and make a note of the location. You can also resize the sprite

Now we can start adding code blocks. We can use the `When Flag clicked` event block to start our logic. So drag that onto our canvas.

We want to ensure the sprite always starts in the same place when the games starts. For this we can use the `go to` Motion block. So drag that onto the canvas, so that it connects to `When Flag clicked`


Sprite Wizard-toad x -140 y 21

Show Size 100 Direction 90

Figure 3: Sprite Location

Have the sprite fall

As the sprite needs to fall all of the time we can use the `forever` Control block to constantly loop through code blocks. Like the `change y by` Motion block.

Now run the game to watch the sprite fall to the bottom of the screen.

Have the sprite rise when a button is pressed

In our forever loop we want to detect if a button has been pressed, if is has, we want to get the sprite to jump up, if not, then keep falling. So we need and `if then else` Control block.

To detect when a key is pressed, we use the `key pressed?` Sensing block. Notice the shape of the gap in the `if then else` Control block. This matches the space of a few other blocks including the `key pressed?` Sensing block.

(If it also the same shape as the `and`/`or`/`not` Operator blocks, so we can trigger on different keys if we wanted to.)

Create the tubes sprite

Hover over the `Choose a srite` and select `Paint` to start creating our obstacles

I have created a coupe of simple pipes that will act as our obstacles.

Move the tube sprites

Now lets setup the pipes constantly moving when we start the game. Make sure we are on the `Sprite1` sheets. and drag in the `When flag clicked` Event block.

Then lets set the tubes to start as far right as we can get it;

We want to constantly move the pipes through the game. For this we can use the `forever` Control block

We can use the `glide secs to x y` Motion block, to move the pipes with a constanyl speed (this will also help us in the future to make the game more flexible)

This will only move the pipes once. A simple trick to reset them is to move the `go to x y` Motion block inside the for loop

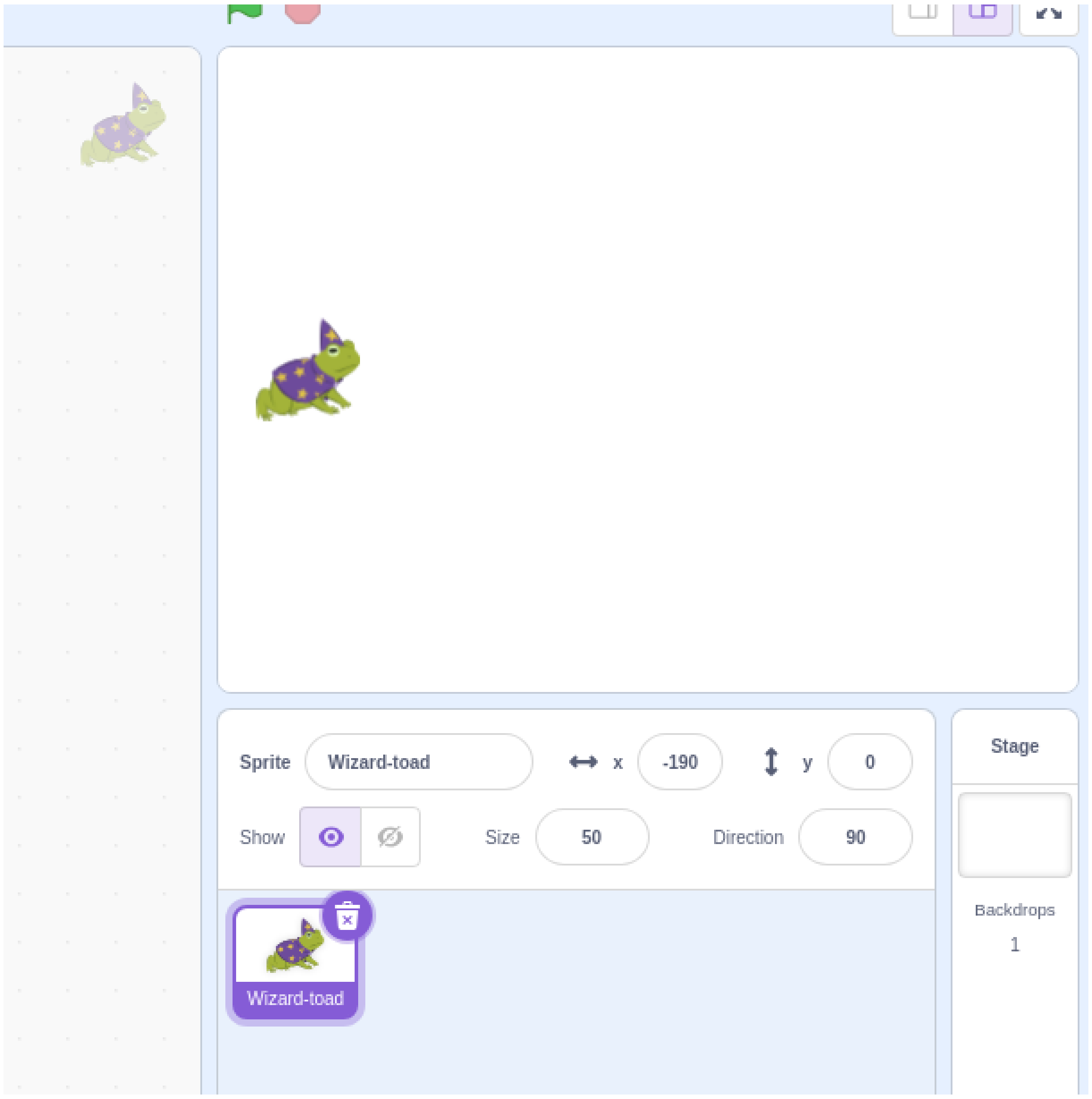


Figure 4: Sprite Starting Location

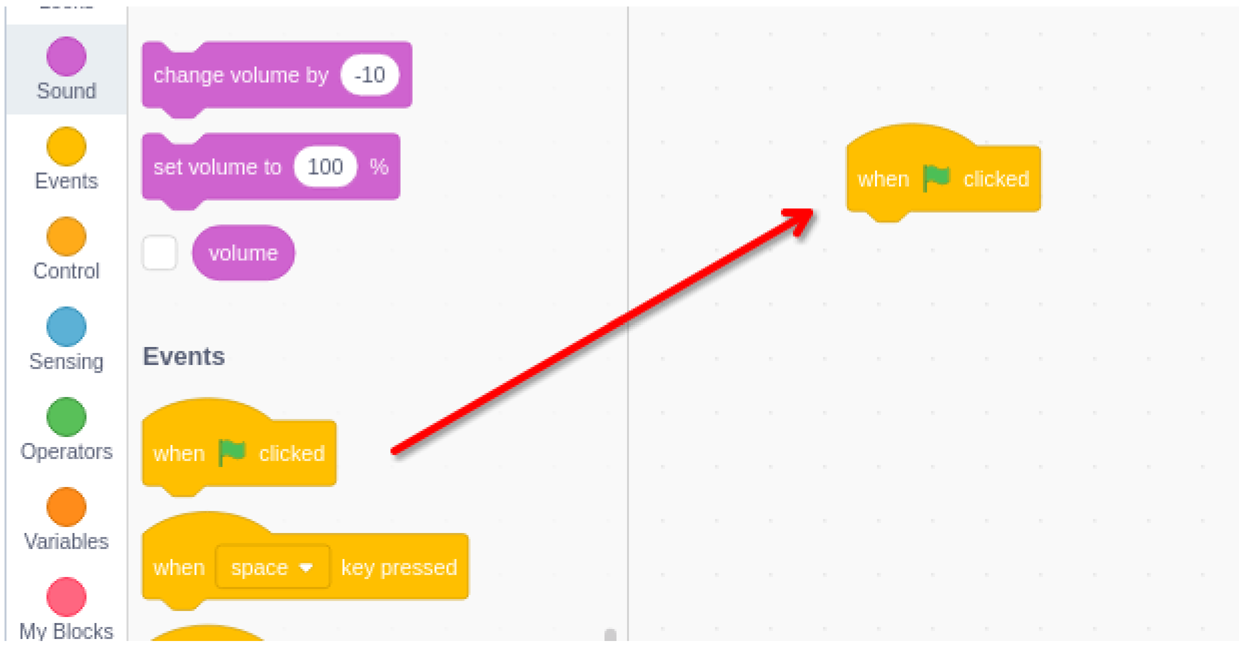


Figure 5: Event Start

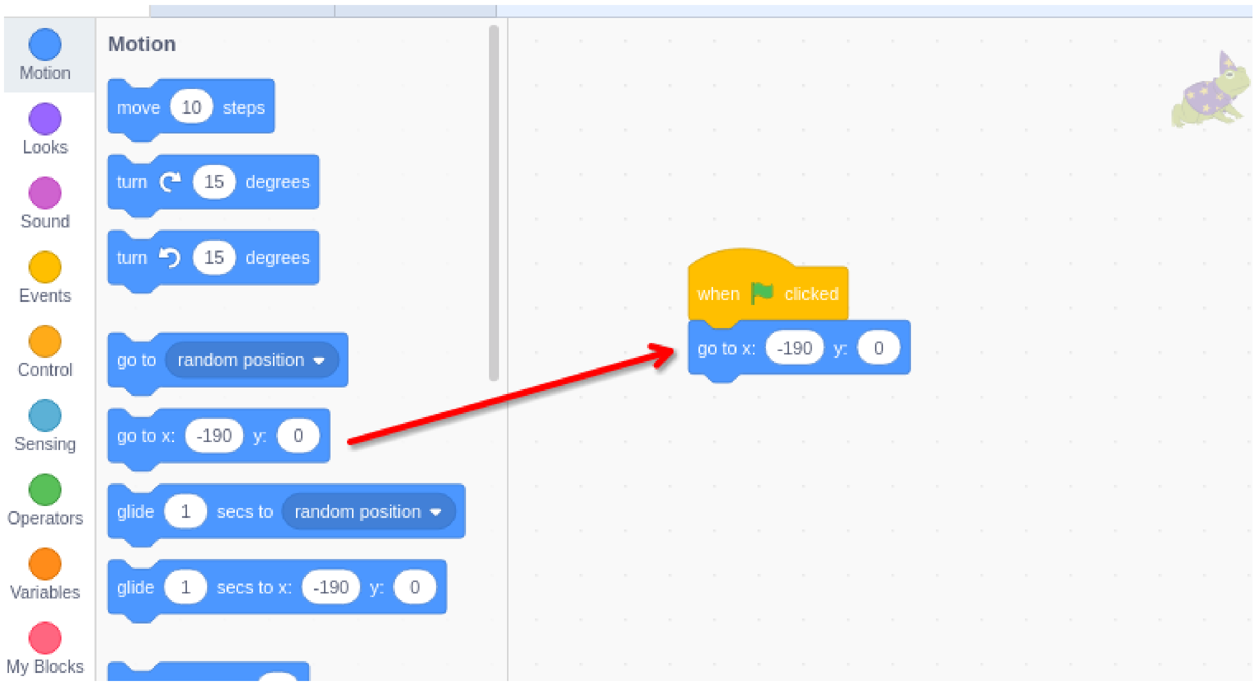


Figure 6: Set Location

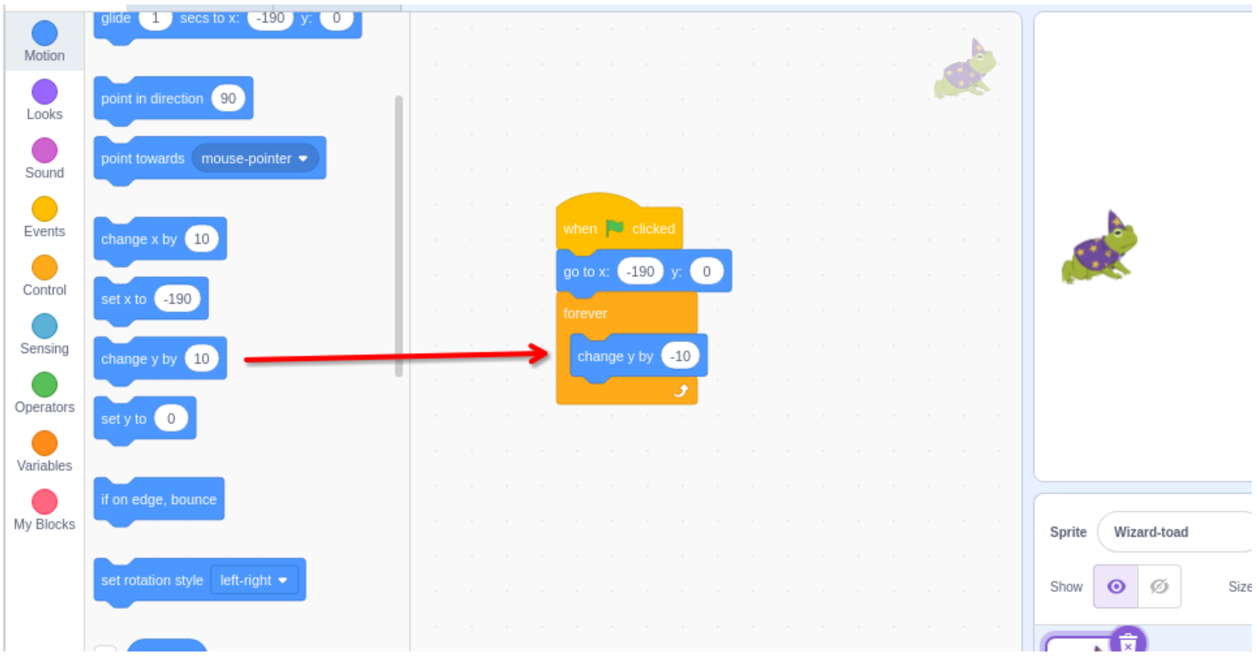


Figure 7: Constanly Falling

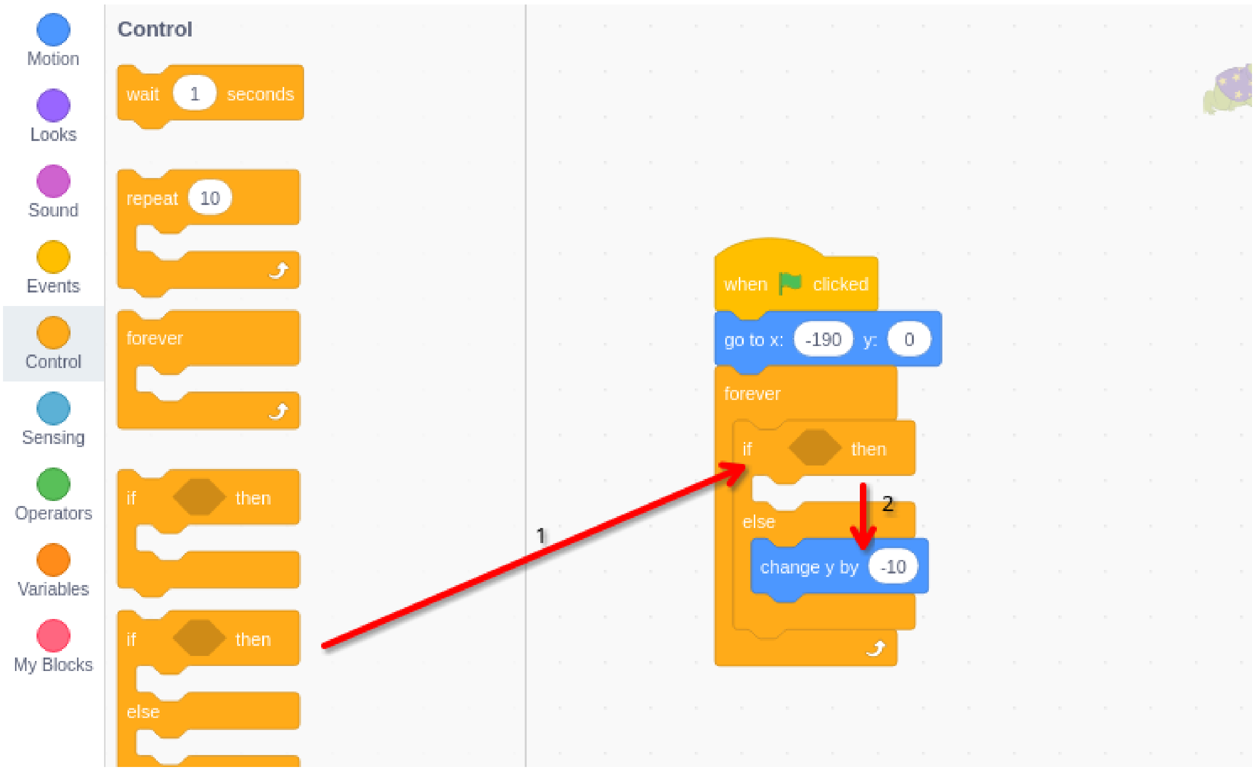
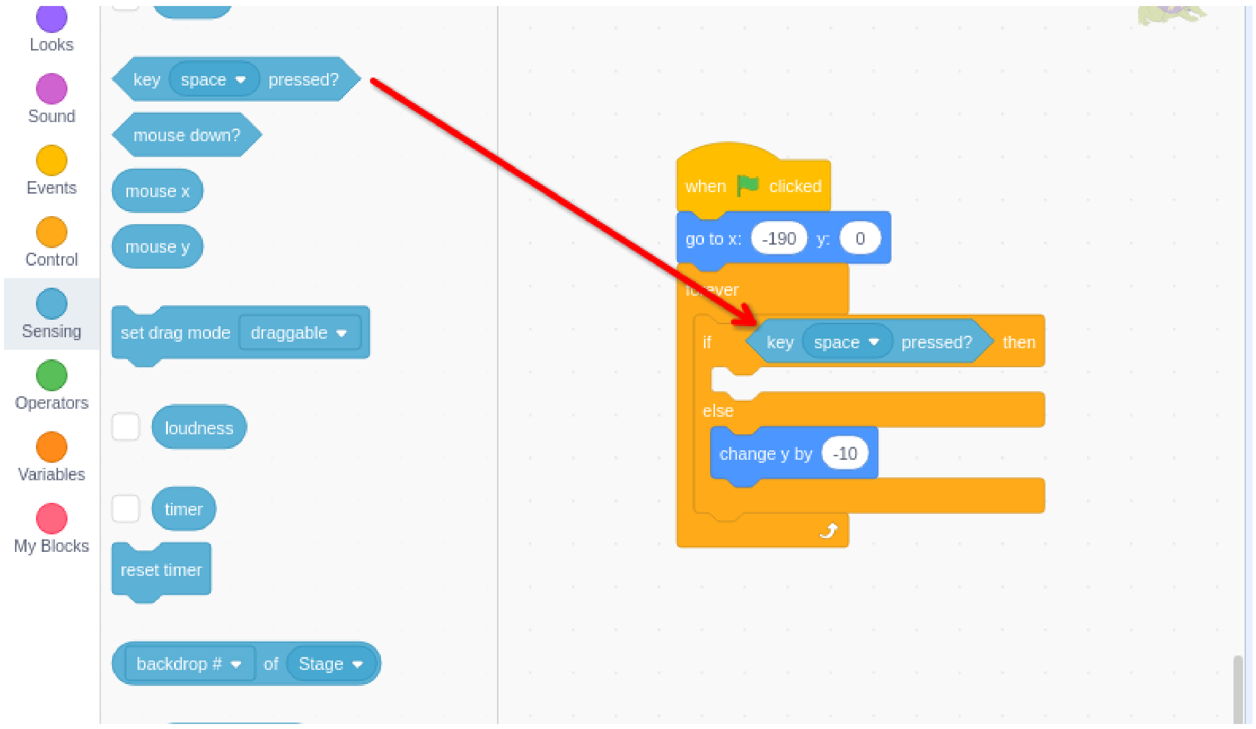
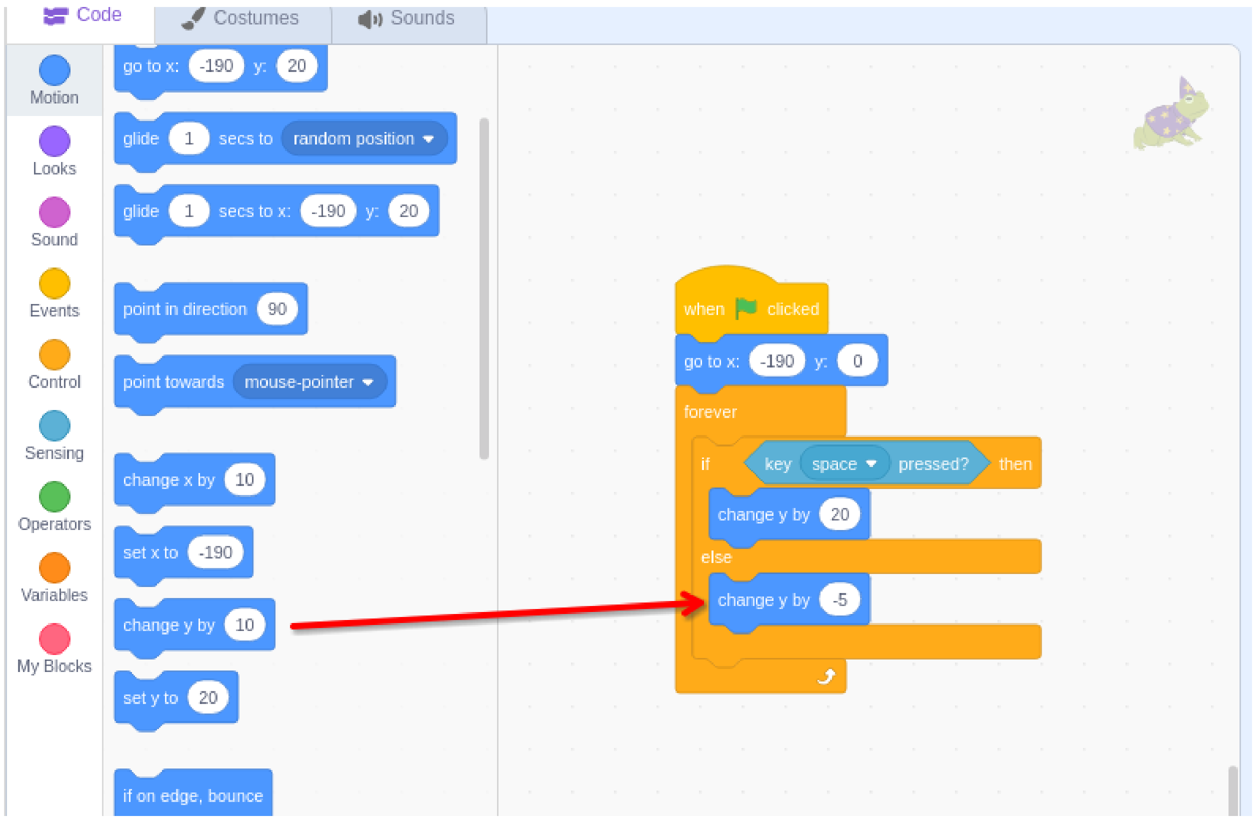


Figure 8: If Then Else



```
when 🏴 clicked
go to x: -190 y: 0
forever
    if key space pressed? then
    else
        change y by -10
```

Figure 9: Key Pressed



```
when 🏴 clicked
go to x: -190 y: 0
forever
    if key space pressed? then
        change y by 20
    else
        change y by -5
```

Figure 10: Jump

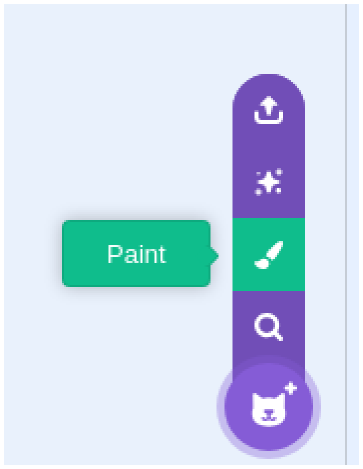


Figure 11: Paint

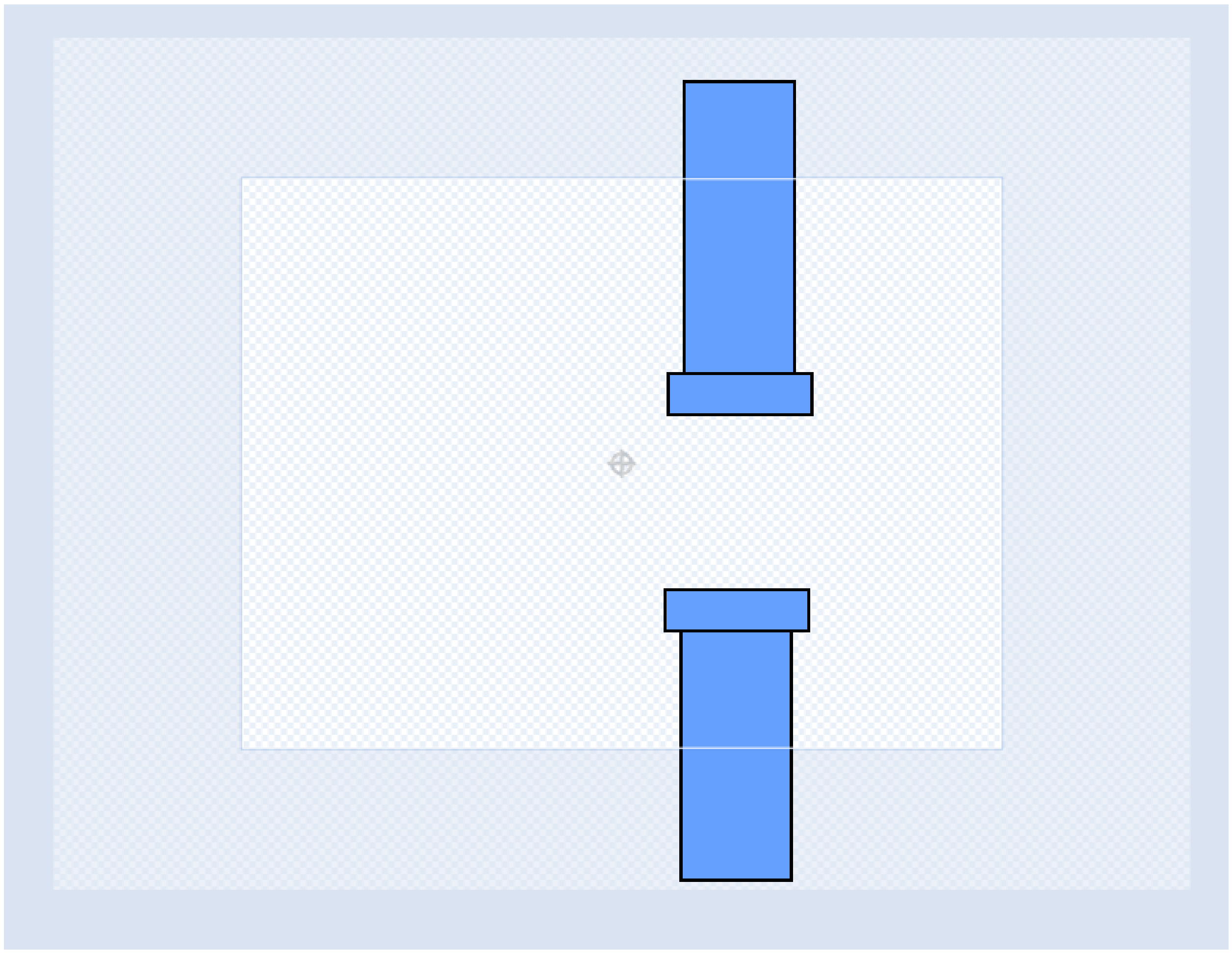


Figure 12: Tubes

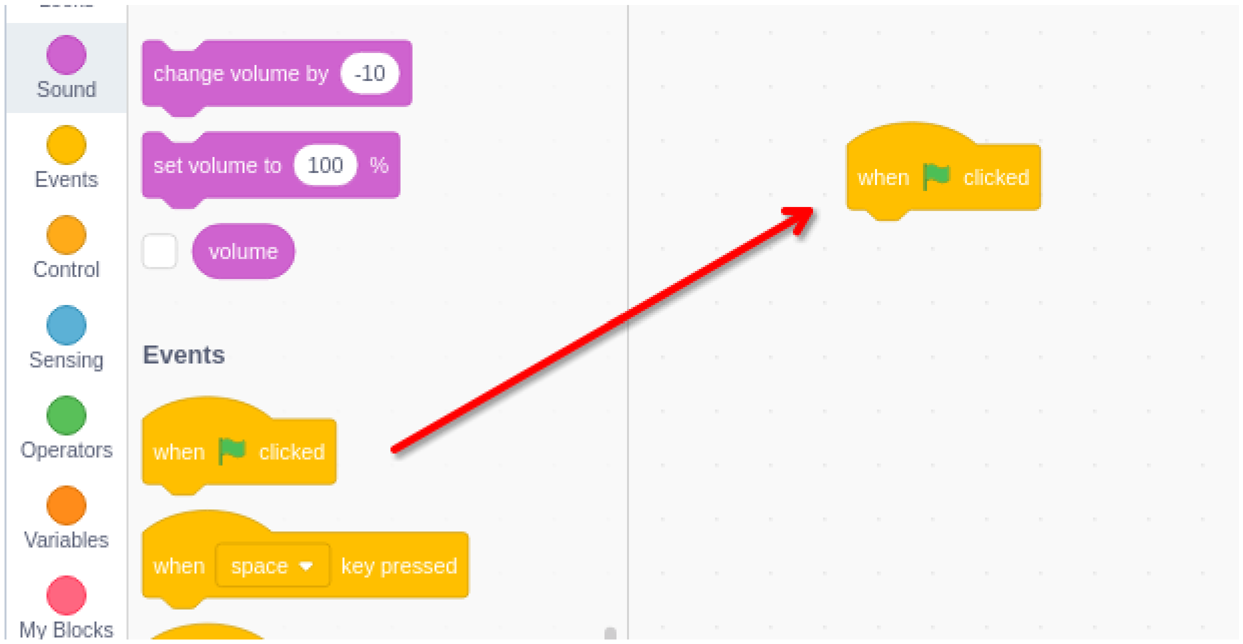


Figure 13: Event Start

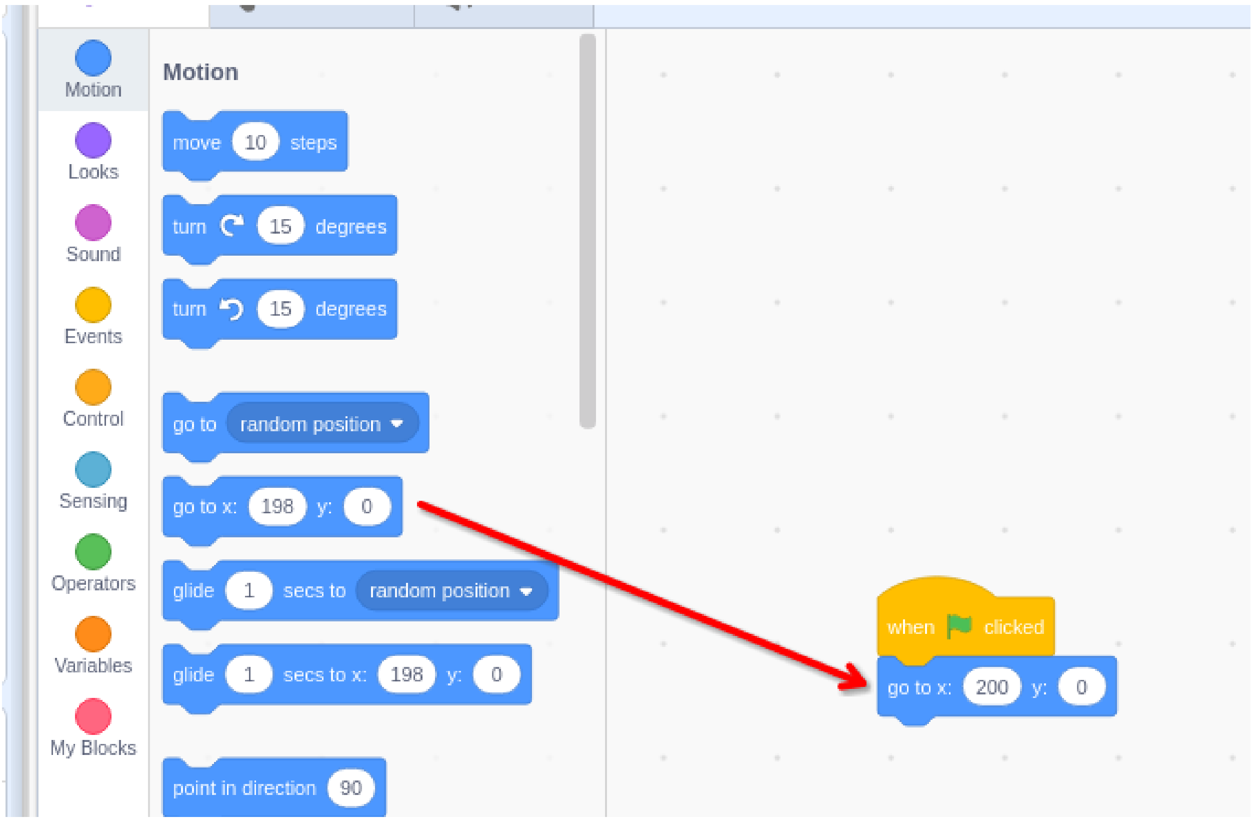


Figure 14: Set Pipes Location

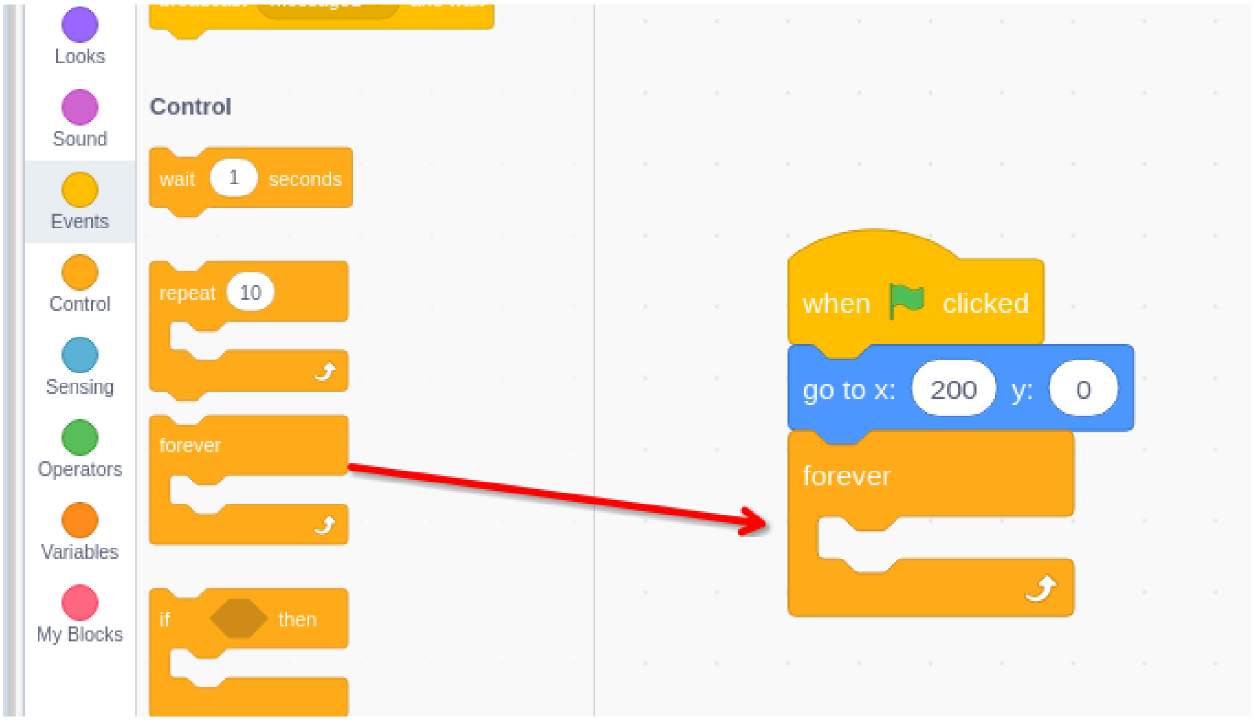


Figure 15: Pipes Loop

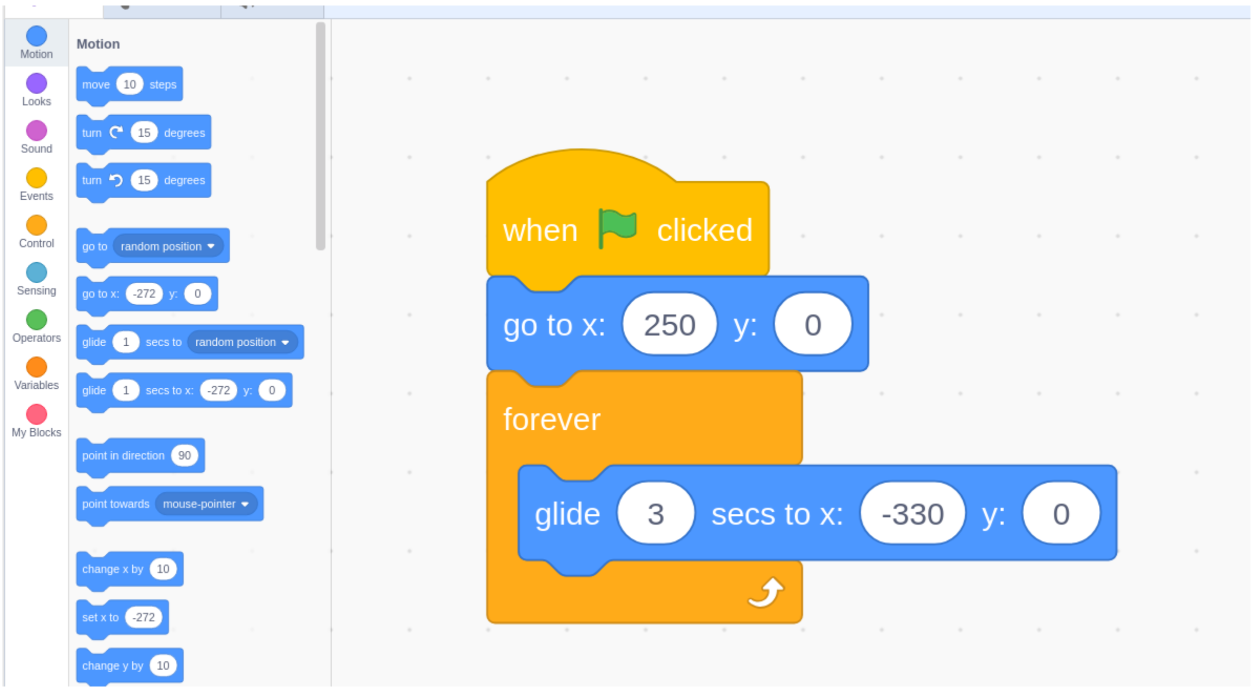
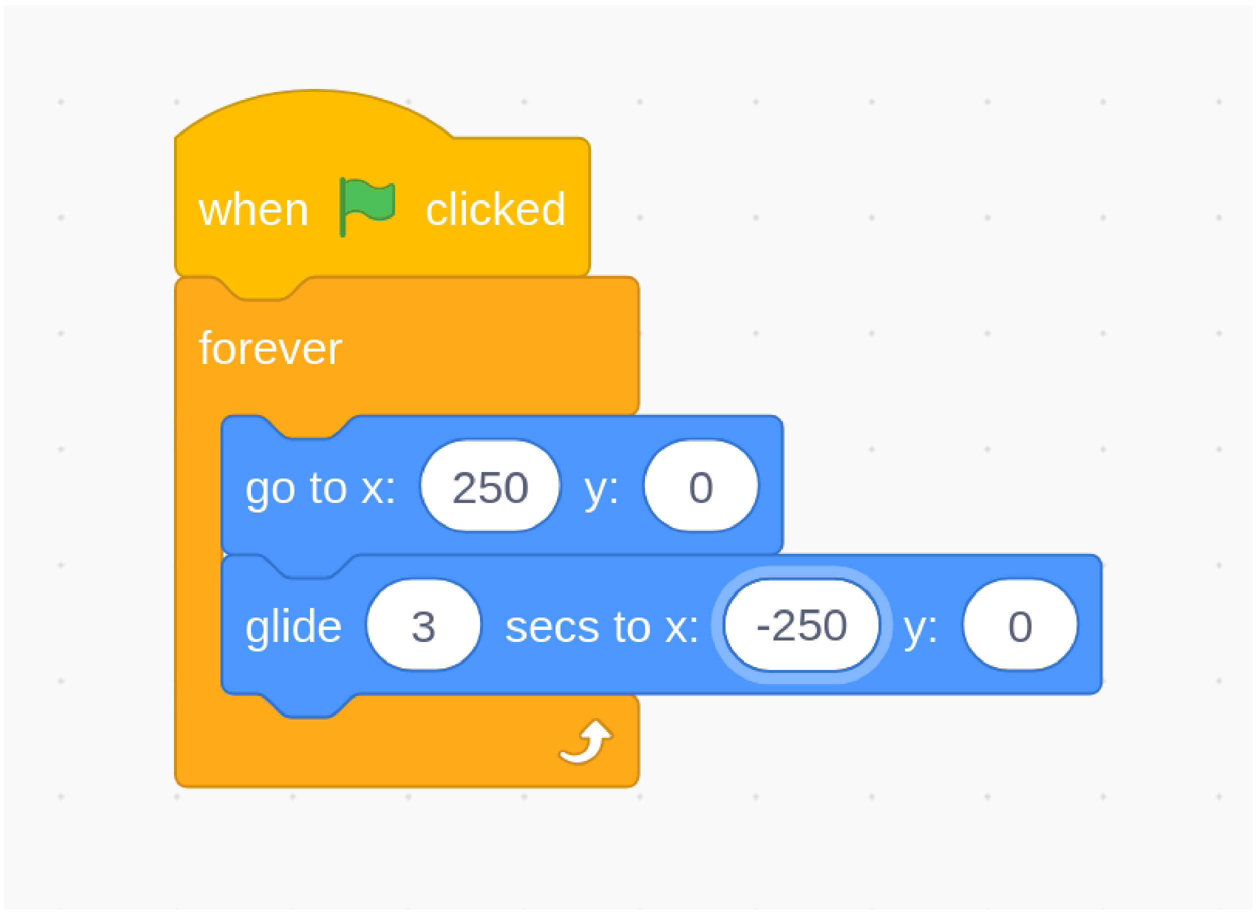


Figure 16: Pipes Moving Loop



```
when 🏳 clicked
forever
    go to x: 250 y: 0
    glide 3 secs to x: -250 y: 0
```

Figure 17: Pipes Moving Loop Reset

Now when we run the game we can see the pipes constantly looping

Detect collision between flappy and tube

So now we have our sprite and pipes moving properly, we now need to detect when they touch.

As this check will be running all the time we need another `forever` Control block that starts from the `When flag clicked` Event block.

This can be on either the sprite or pipes page (I will leave it on the pipes page)

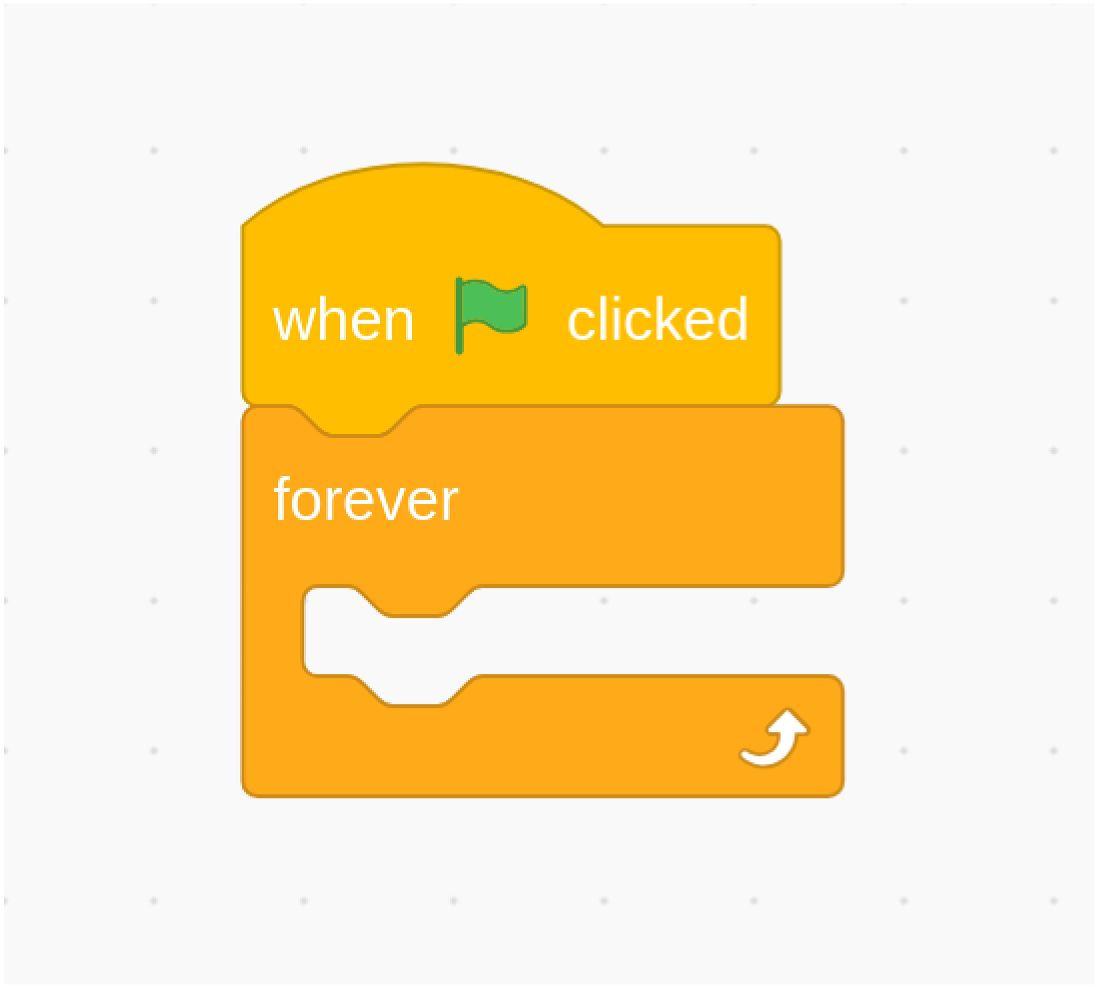


Figure 18: Start Forvever

We can use the `touching` Sesning block to detect when 2 sprites touch. So lets put this into an `if then` Control block;

To finish the game we can use the `Stop` Control block

Increament the score when going through the pipes

Finally we need to show the most important thing, the score. To hold this we will create a `varible` called `Score`.

Click `Make a varible` in the `Varibles` block section and call it score.

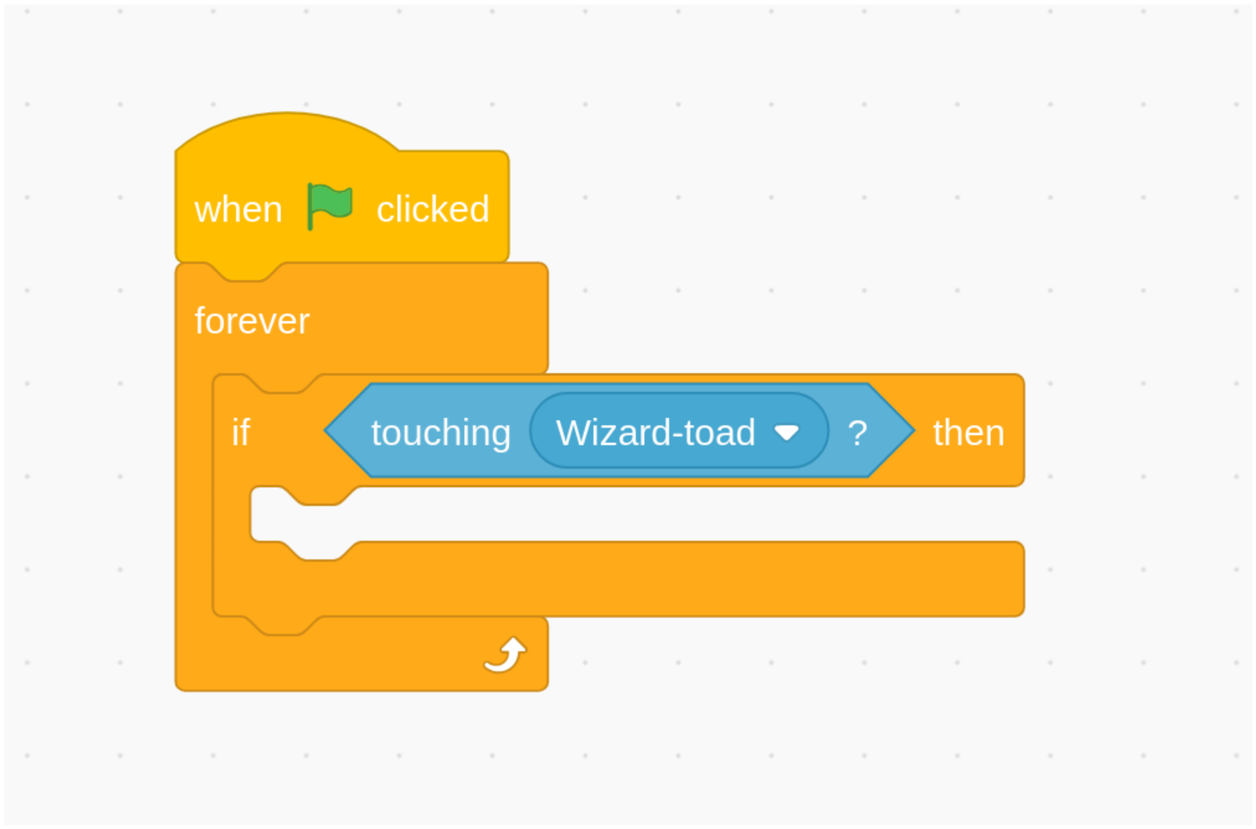


Figure 19: Touching

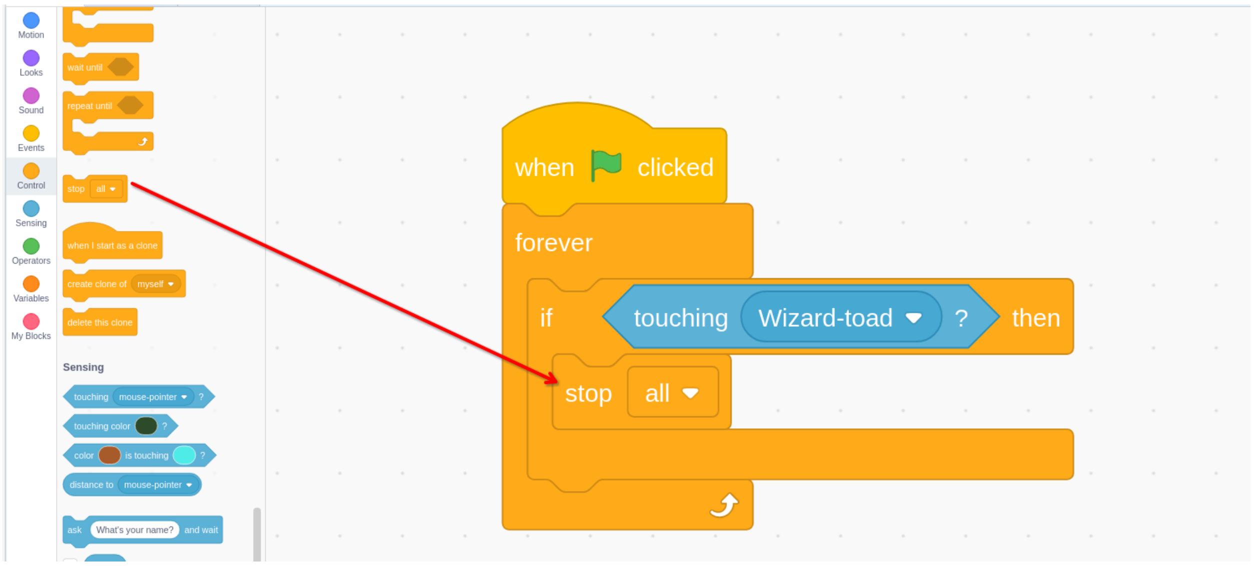
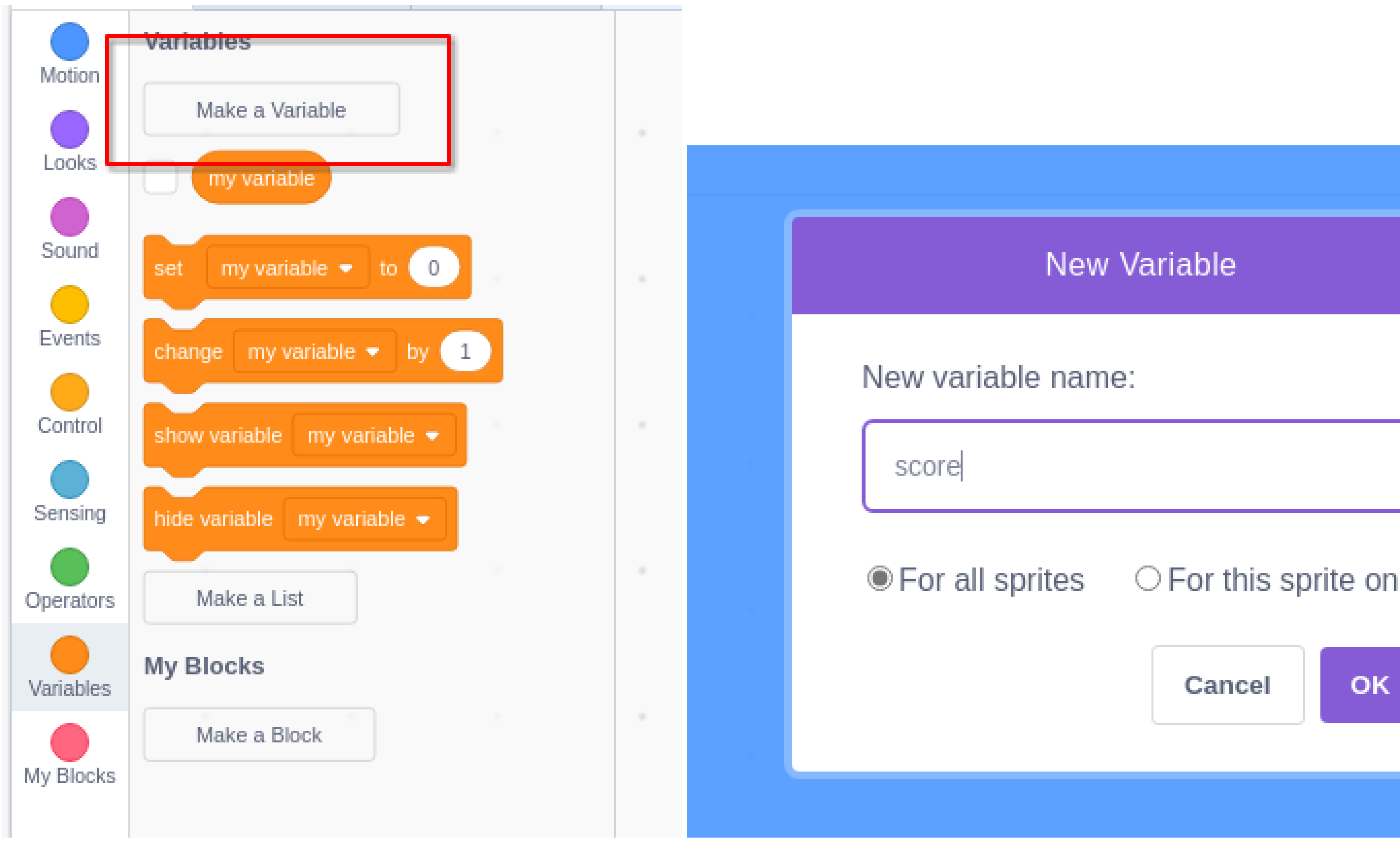


Figure 20: Stop



This is automatically shown on screen when the check box next to the varilbe is ticked

We can add this to the moving pipes loops to add 1 to the score every loop

The score is now cumlative through games, so each time the game starts lets reset the score.

We can technically add this to after any of the `When flag clicked` Event block, but for neatness we will add to it of the moving loop.

That is all of the basic features of the game complete, but we can still add loads of polish.

Improvements

- Play the game to adjust the gap between the pipes.
- Set a nice backdrop

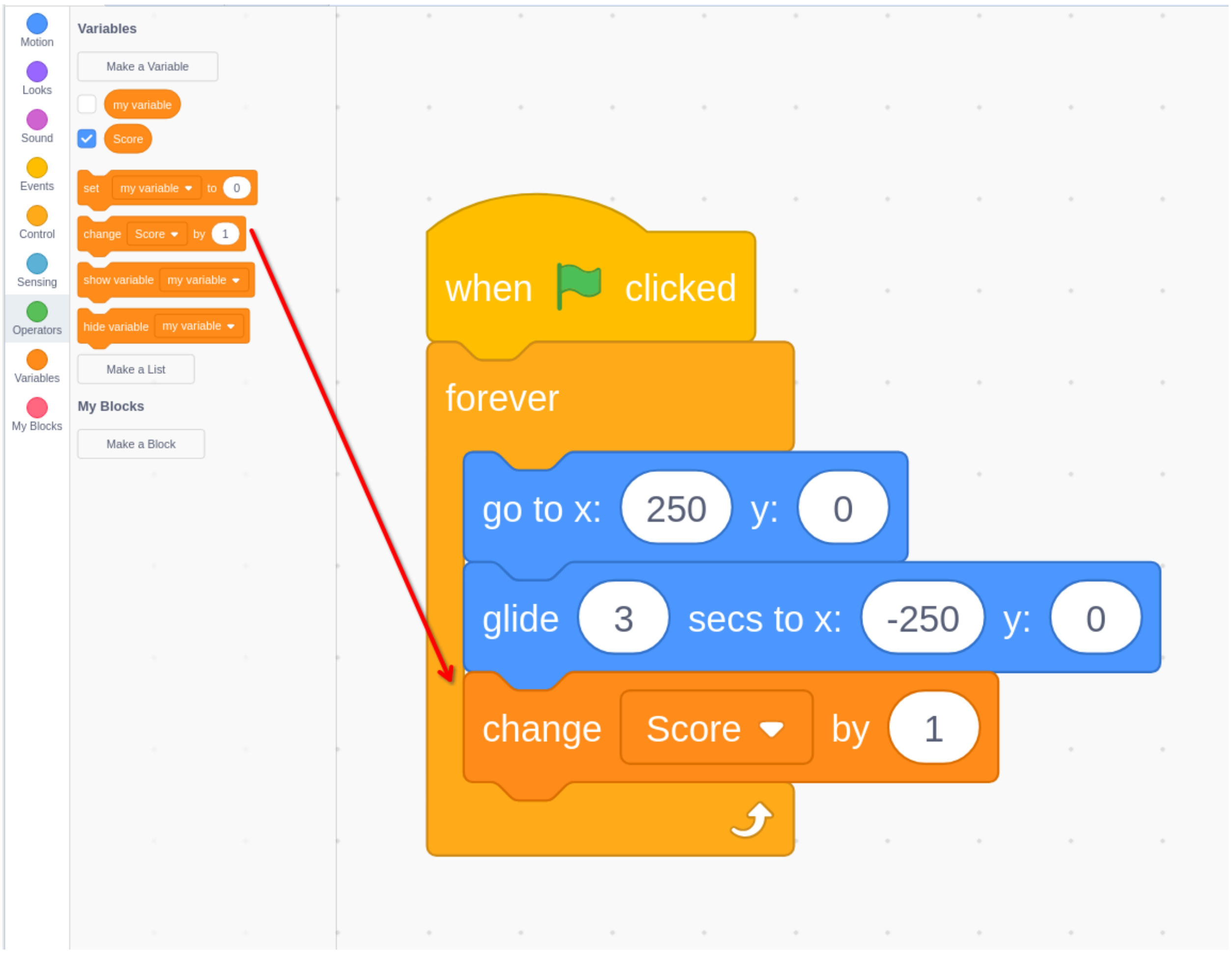


Figure 21: Change Score

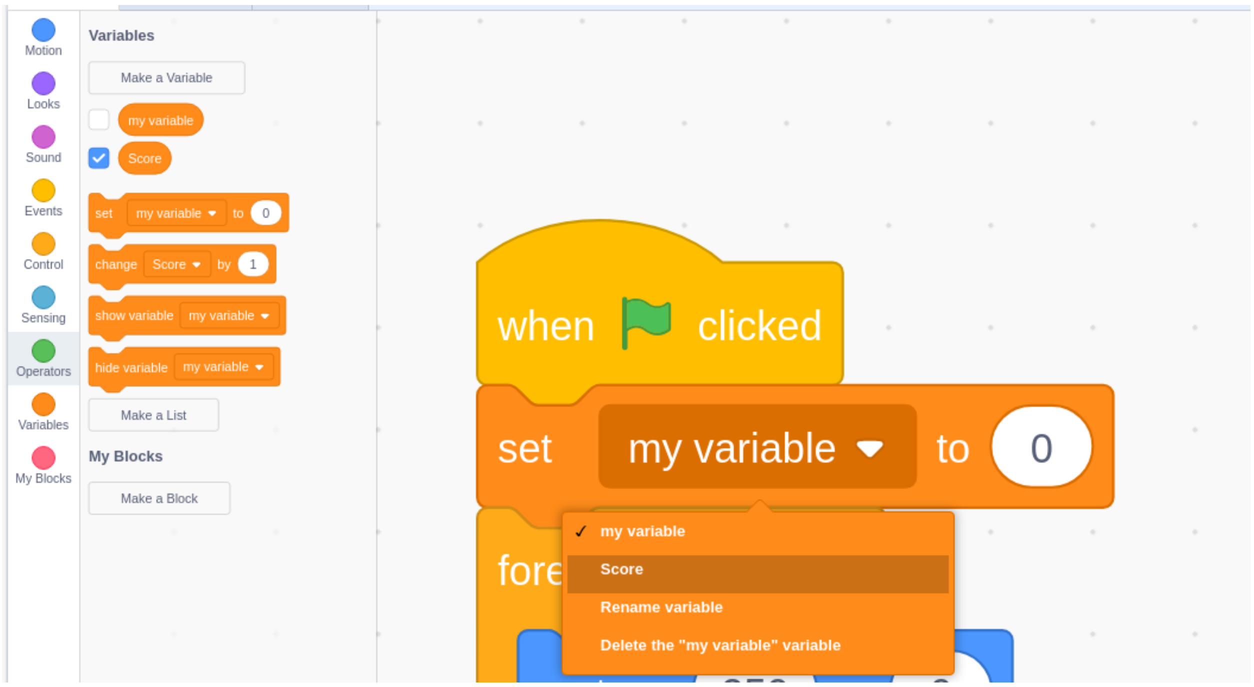
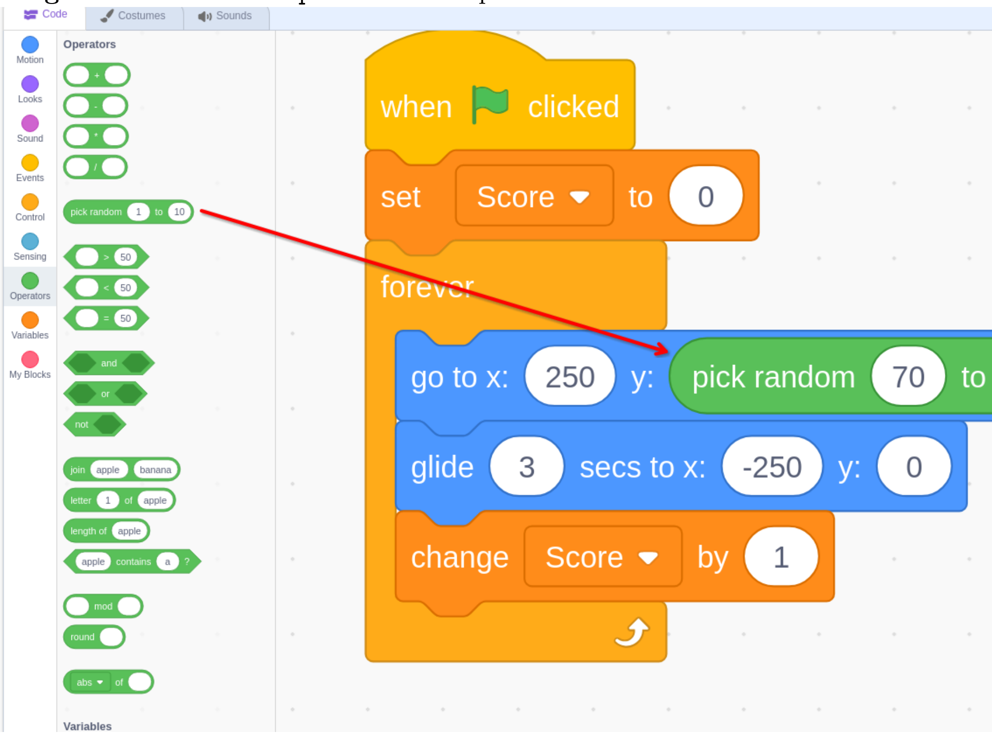


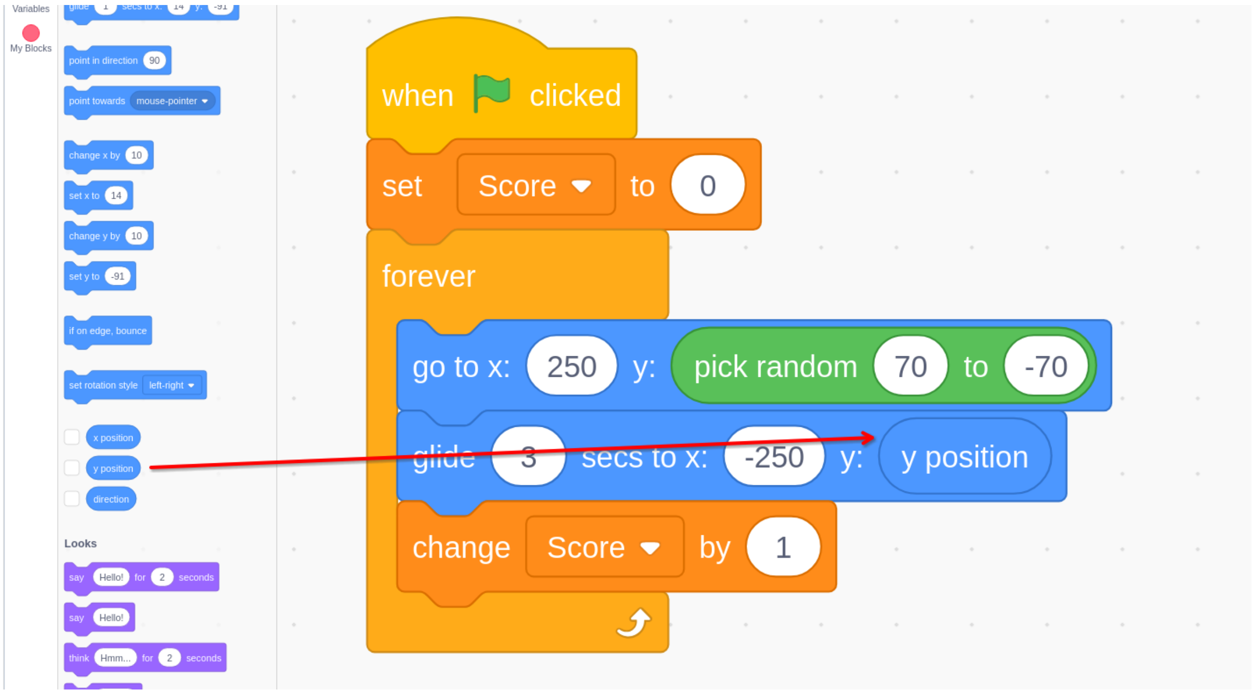
Figure 22: Rest Score

Have the pipes come in at different heights We can use the `pick random` Operator block to create a random height for the blocks to come in at.



I dragged the pipes sprite on the game canvas to work out the best y co-ordinates to come go between.

We need to insert the `y position` Motion block into the `glide` Motion block to keep the level of the pipes constant.



```
when 🏴 clicked
set Score to 0
forever
    go to x: 250 y: pick random 70 to -70
    glide 3 secs to x: -250 y: y position
    change Score by 1
```

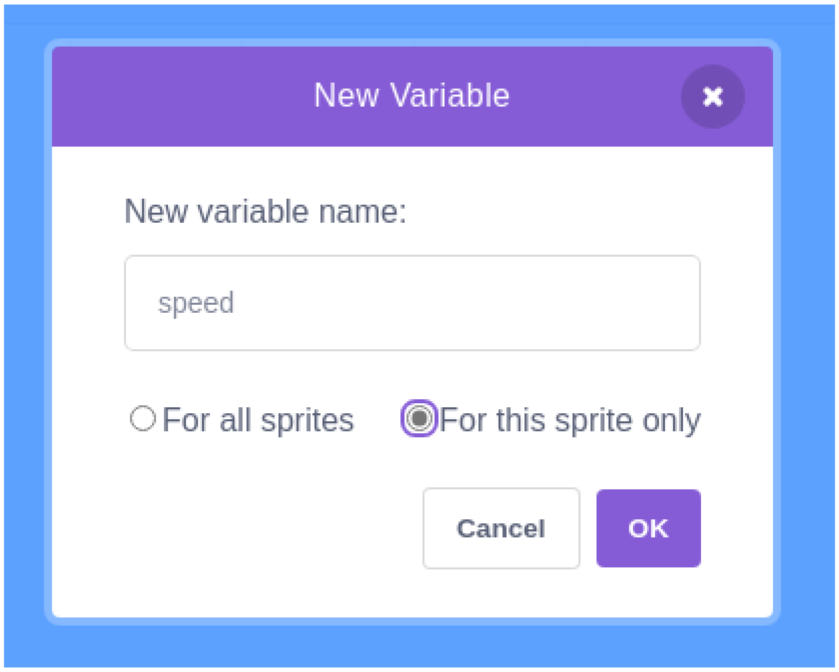
Figure 23: Different Heights Y

Speed up the game To make the game speed up and be harder as it continues. We can change the `glide` Motion blocks speed to be a varible. So lets create one called `speed`

Be sure to untick it so it doesn't show on screen. Now lets set its inital speed when we start the game;

Every loops we want to remove a small amount so the process is gradual.

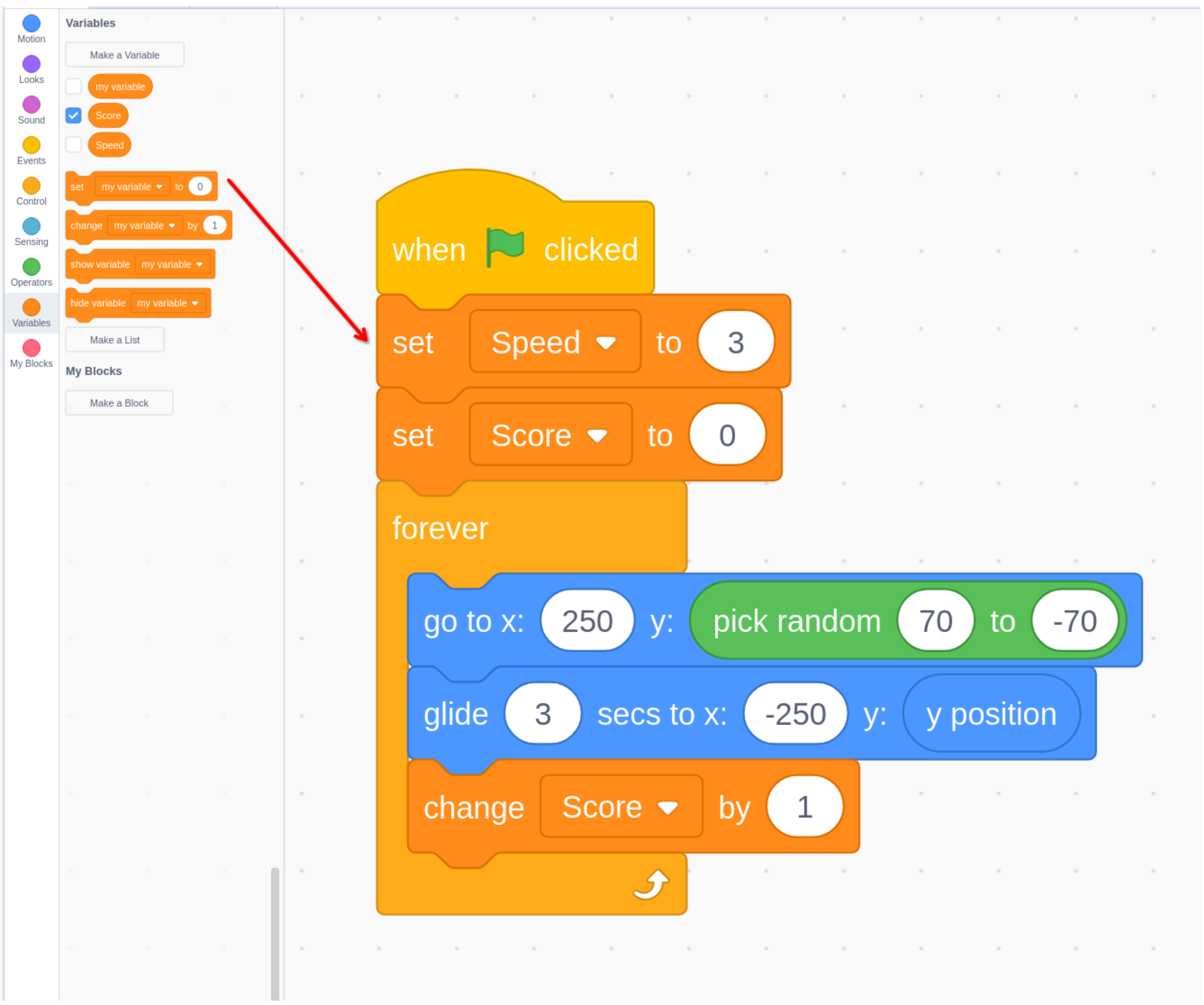
Now lets actually use the `Speed` variable in the `glide` Motion block



speed

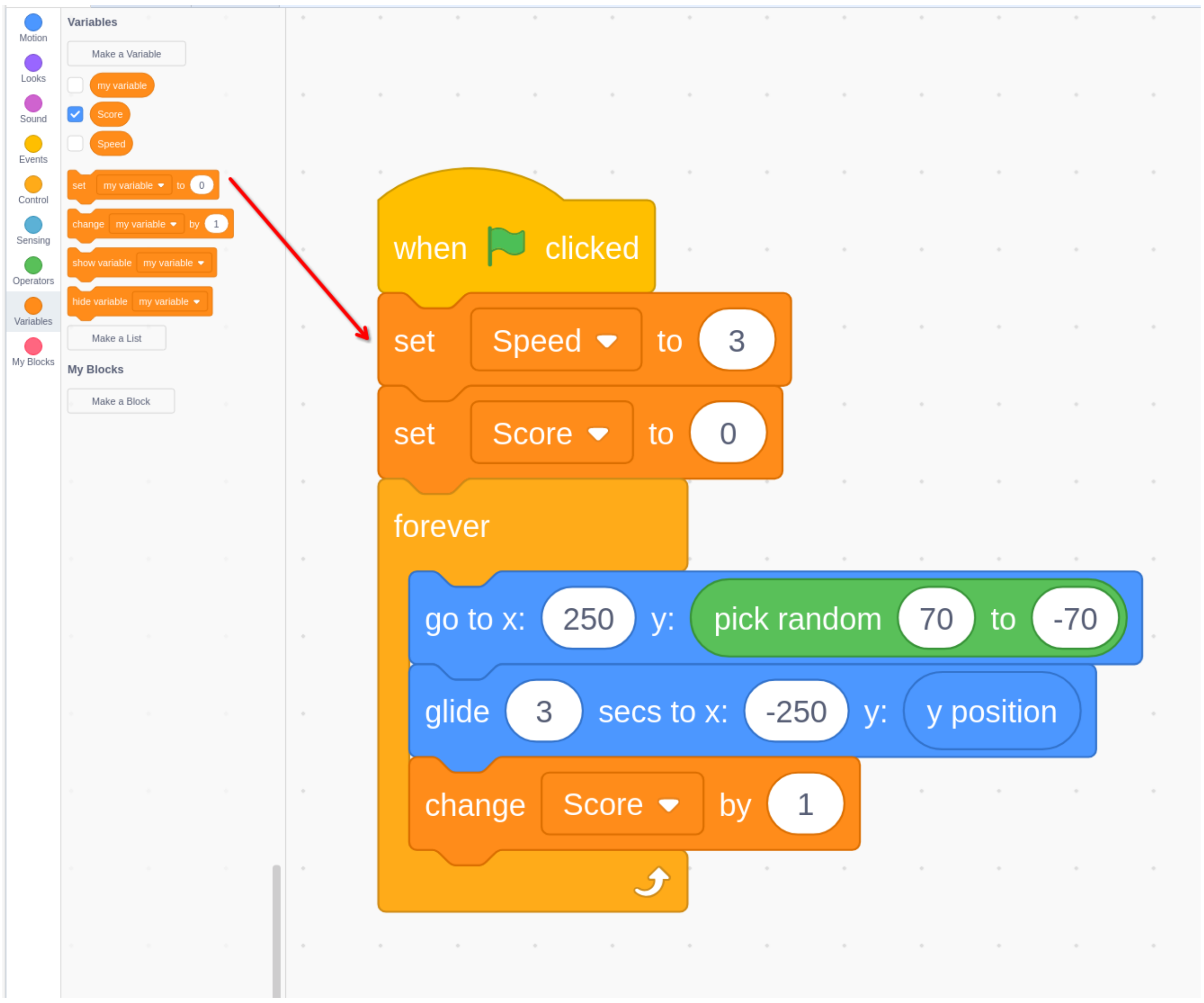
For all sprites [✓ For this sprite only]

Figure 24: Variable Speed



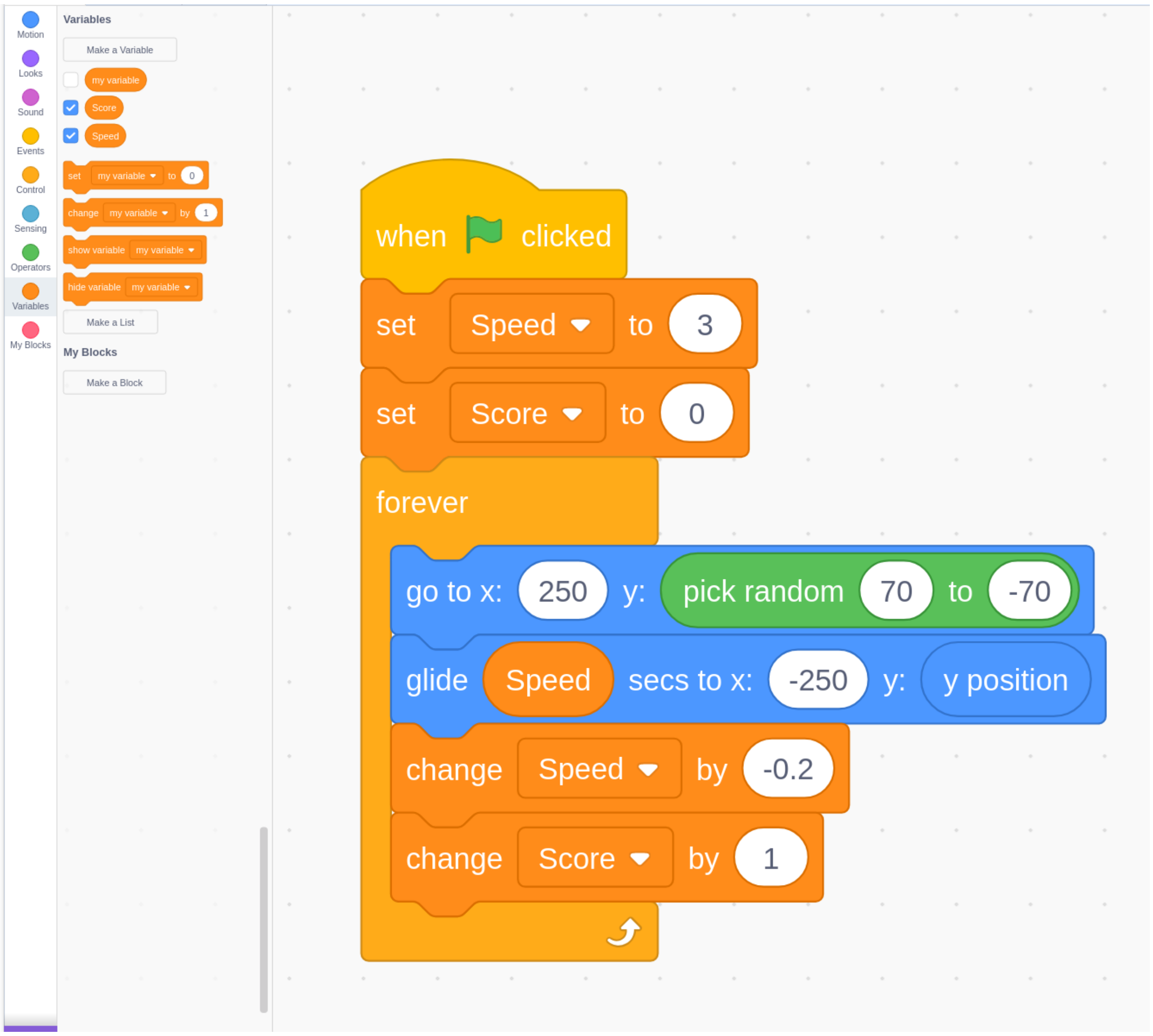
```
when 🏁 clicked
set Speed to 3
set Score to 0
forever
    go to x: 250 y: pick random 70 to -70
    glide 3 secs to x: -250 y: y position
    change Score by 1
```

Figure 25: Variable Set Speed



```
when [flag] clicked
set Speed to 3
set Score to 0
forever
    go to x: 250 y: pick random 70 to -70
    glide 3 secs to x: -250 y: y position
    change Score by 1
```

Figure 26: Variable Set Speed



```
when ⚑ clicked
set Speed to 3
set Score to 0
forever
    go to x: 250 y: pick random 70 to -70
    glide Speed secs to x: -250 y: y position
    change Speed by -0.2
    change Score by 1
```

Figure 27: Use Variable Speed