SUMMIT
ACADEMY
NEXT LEVEL
CODING CLUB

Lesson Plan #2

STARTER PROJECT

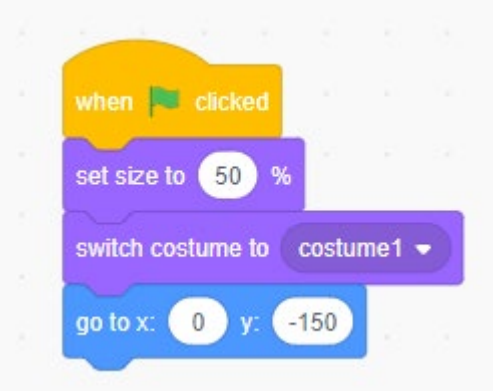
Connor Kalvar | 9/29/2021

Second Project – Space invader Clone

Hopefully having completed the first project has given the students a basic understanding of how Scratch works. This next project will focus on using two of Scratch's more important features: clones and broadcast messages, which are both very useful in creating games.

1| We will start off by creating a new project, which we do by clicking the create button on the home page at Scratch.mit.edu. Hopefully you have set up an account at this point, if you haven't, refer to the first lesson plan to find out how to set up a Scratch account.

2| We'll start off with this script:



We set size to 50% so that Scratch cat doesn't take up as much room. Then we set the costume to costume 1, this is to avoid any future problems of Scratch cat being stuck on the wrong costume. Future proofing code is always a good idea. Finally, we go to x:0 and y; -150 to put our character at the bottom of the screen.

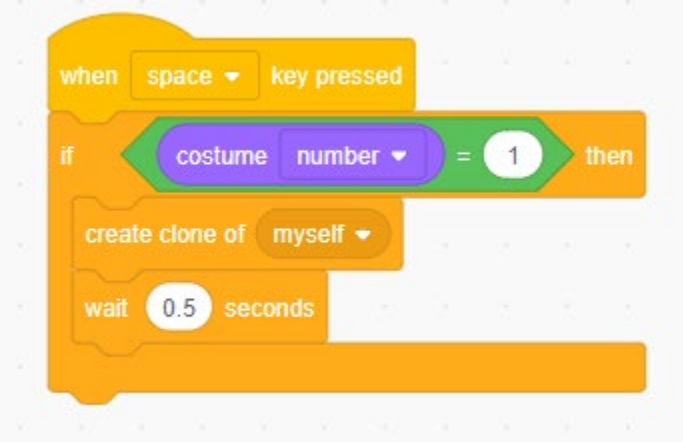
3| Now, let's add our basic movement. Hopefully these scripts look familiar:

4| So we can only move left and right because we don't need up or down movement for this game. But what's with the if then statements? This is for later on when we add the lasers that Scratch cat will shoot, so that the lasers don't move when we press the arrow keys. The costume number block can be found in looks. The costume number is the number on the top left of every costume. See it? It's basically just the order they're listed in.

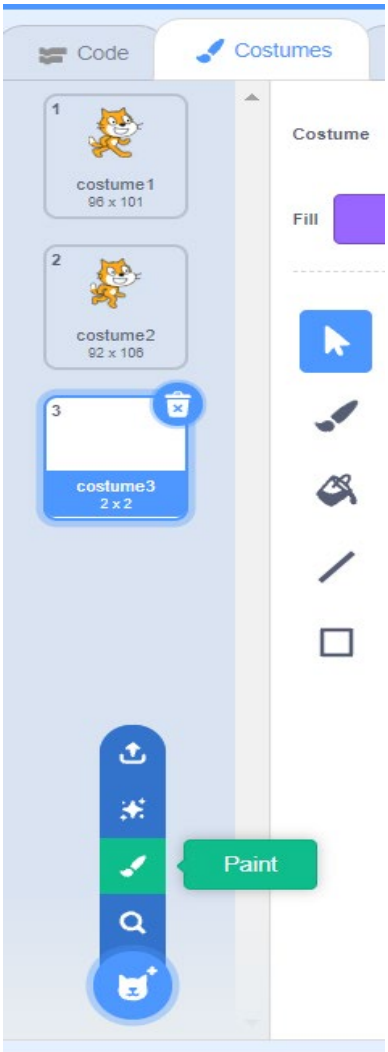
The green block there is an equal statement which can be found under operators, and as the name suggests it sees if two things are equal.

So we're seeing if when we press the arrows, if what is trying to move has a costume number of 1, and if it doesn't, then it doesn't move.

5| Now let's add the shooting functionality:

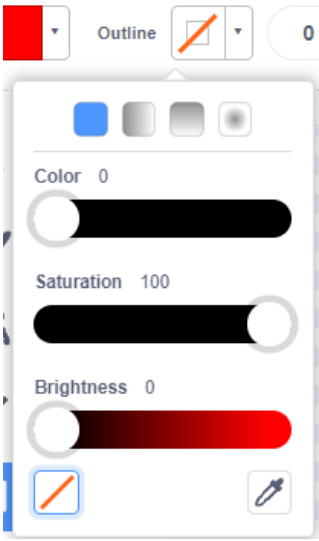


6| Similar to the movement scripts we have an "if then" statement checking to see if the costume number is 1 so that the lasers don't shoot more lasers. The "wait 0.5 second" block makes it so that we can only shoot ever half second. The "create clone of myself" block is what is going to allow us to shoot lasers. The create clone of myself block can be found under the control tab.



7| Let's go create a laser costume for us to us. Once again, we click into the costume tab and go to the bottom left and select paint from the popup menu. Select the square tool and then go to the colored box that says "fill". Click it and change the color to red (or blue) Like this:

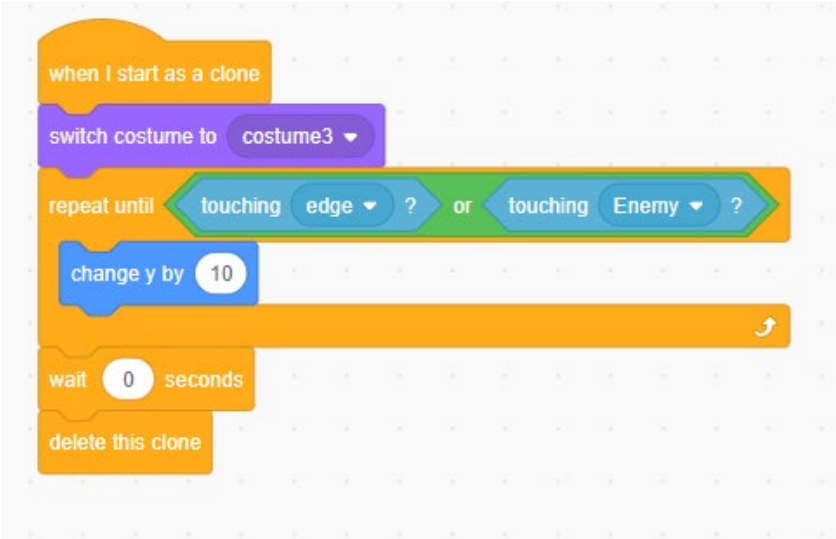
8| Then, click the box next to it (outline) And click the diagonal red line on the bottom left corner, this will make the outline transparent:



9| Now create a tall and narrow rectangle, this is our laser costume.

Click back into the code, so we can get to shooting with the laser.

10| This is the script we will use to shoot:



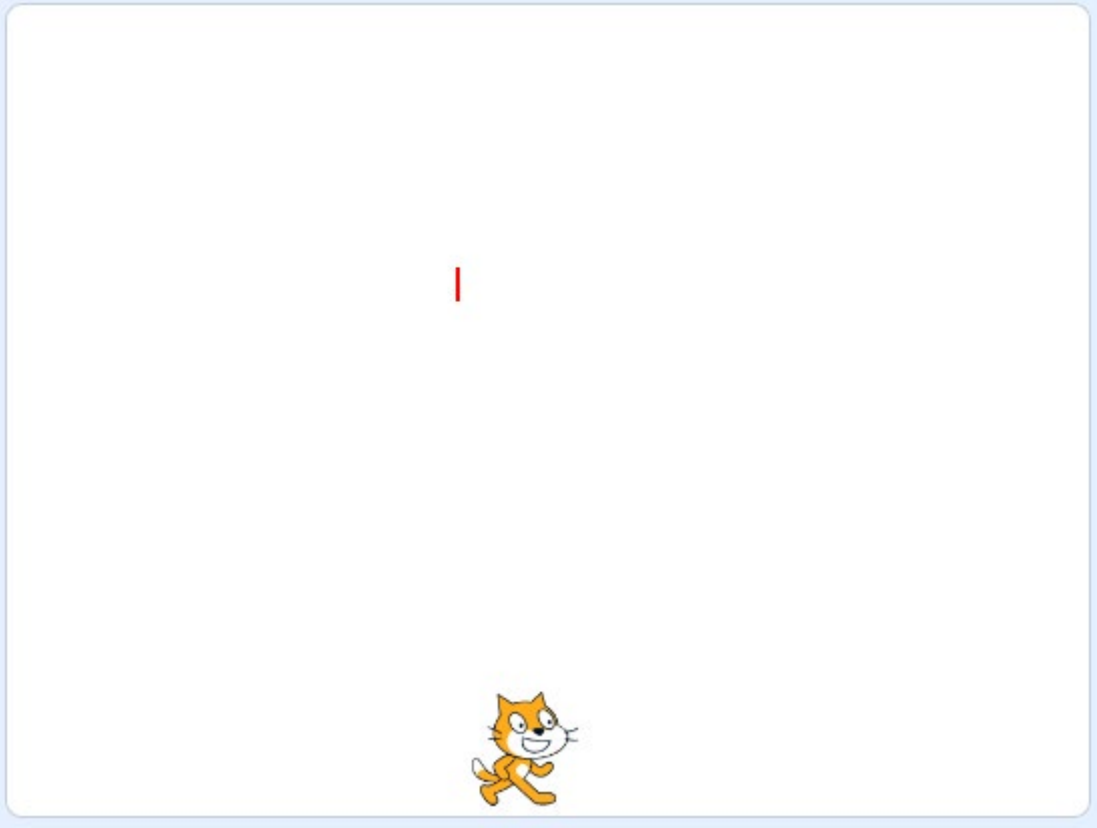
Remember that "create clone of myself" block from earlier, the "when I start as a clone" block gives them functionality separate from the rest of the sprite. *Clones inherit the position, costume, and every script that starts with a control block* (when key pressed, when I receive, when this sprite clicked, etc) except when green flag is clicked. So that's why we put in those checks for costume number earlier. Because the clones get those scripts, but we change the clone to our laser costume, changing its costume number to 3, which isn't 1, meaning that the clones will not do those actions.

The "repeat until" does what its name implies, it repeats until some condition is met. In this case we are repeating until we touch the edge or" enemy" (we haven't created that sprite yet) and only

one of these conditions has to be met for the “repeat until” loop to stop. The “or” block can be found in operators.

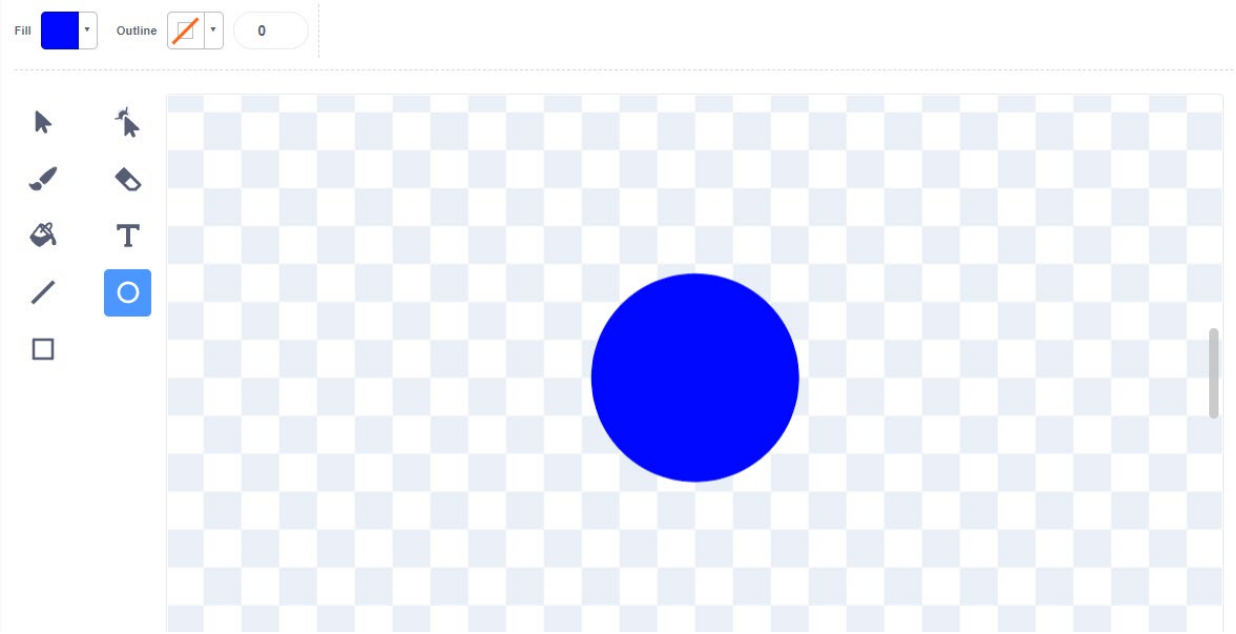
The action inside the “repeat until” loop is to change the y by 10 so that the laser flies forward. The “wait 0 seconds” basically waits a frame. This is so that when the laser collides with an enemy, it doesn’t delete itself before the enemy can check to see if it’s touching a laser.

11 Now test out all these scripts. You should be able to move left and right and shoot out lasers.

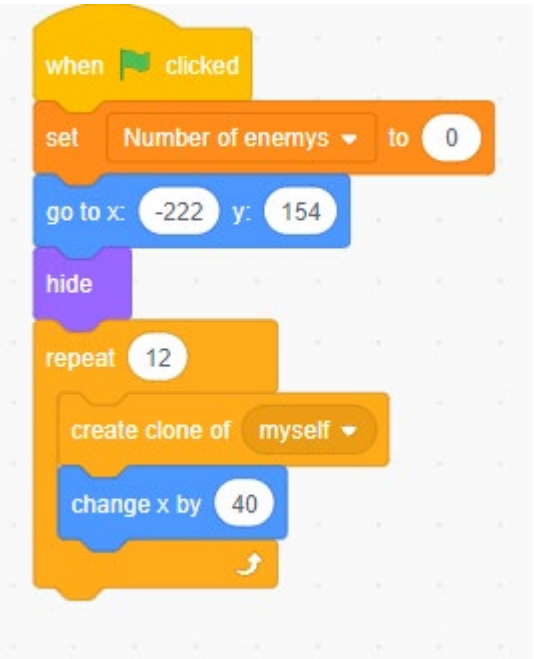


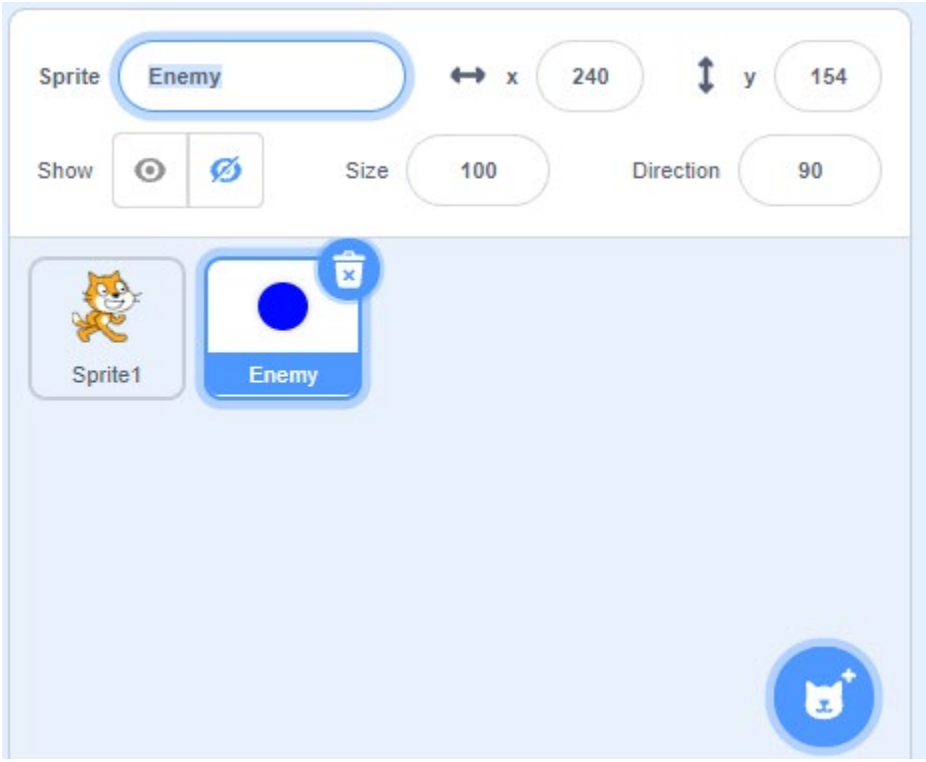
Now, let’s make something to shoot at.

12 Like last time, go to the bottom right corner and select paint to create a new sprite. Select the circle tool (above and to the right of the square tool) and make a circle (it can be any color).



12 | We can rename sprite by going to the sprite list under the stage, clicking on the sprite, and then clicking the sprite box on the top right.



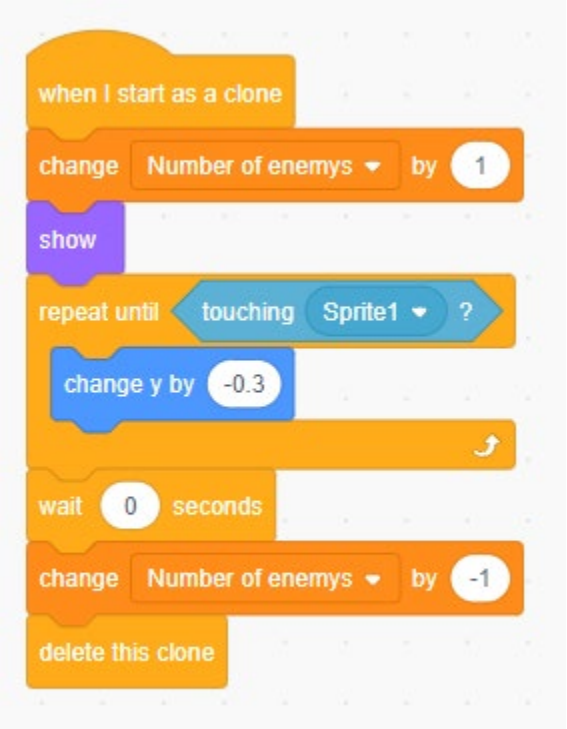


13 | Now, let's give this enemy some functionality. Go back into the code tab and start off by creating a new variable called "number of enemies". This will allow us to know when we have destroyed all the enemies. Our first script will be this. We start off by resetting the variable number of enemies, then we go to the top left of the screen.

Then we hide because we are only going to see the clones of the enemy and not the sprite itself. Then, we create a clone and move to the right 40 units. Remember how I said one of the things a clone inherited from the sprite it is cloned from is position. This loop allows us to very easily create a line of clones, as the clones simply start in the right position because they go to where the sprite is.

15| Make sure to go back to Scratch cat and add in that "if touching enemy" to the "when I start as clone" script.

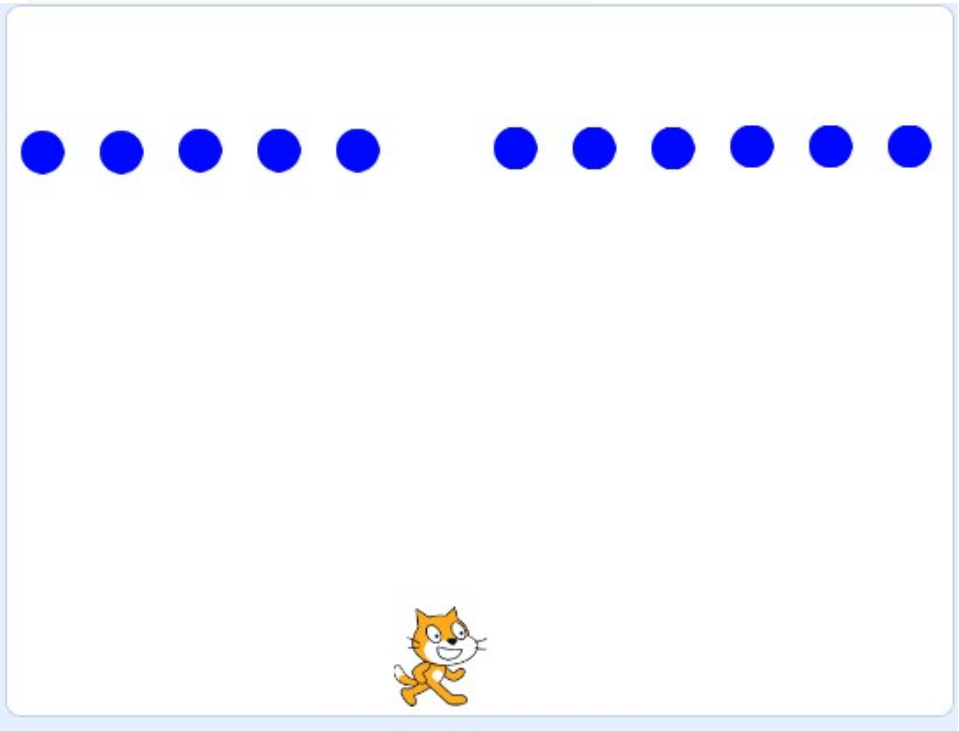
16| Now we need to give these clones their functionality. There are two "when I start as clone" script we will need.



This first one controls movement and keeps track of all the clones. We start off by changing the number of enemies by one for each clone that is created. Then we show the clone, as one of the other things a clone inherits is where its being shown or not. Then we use a "repeat until" loop to move the clone downwards towards the player until we touch sprite1 or a clone of sprite1. That's right, the touching block can also tell us when we're touching a clone of a sprite.

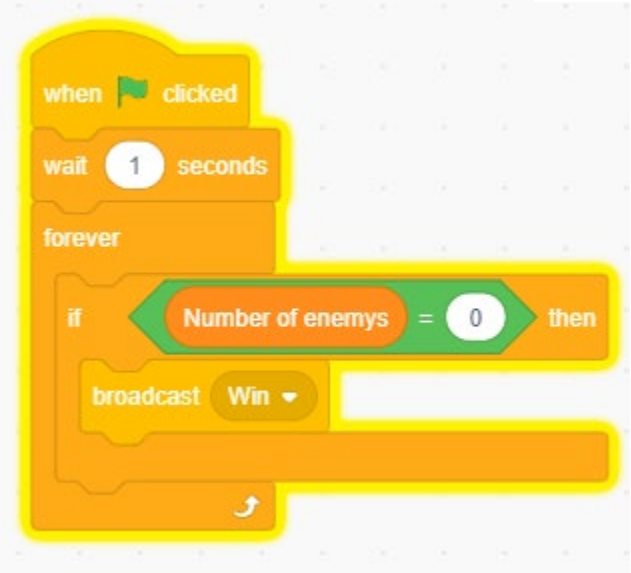
So when we touch a clone of sprite1 (the laser) we stop moving, wait a frame, so that the clone doesn't delete itself before the laser knows it's touching an enemy. We then remove one from the enemy count and delete the clone.

17| Go ahead and test these scripts out. When you click the green flag, a line of enemies should show up, and when you press "space", and a laser hits the enemy, they should disappear.

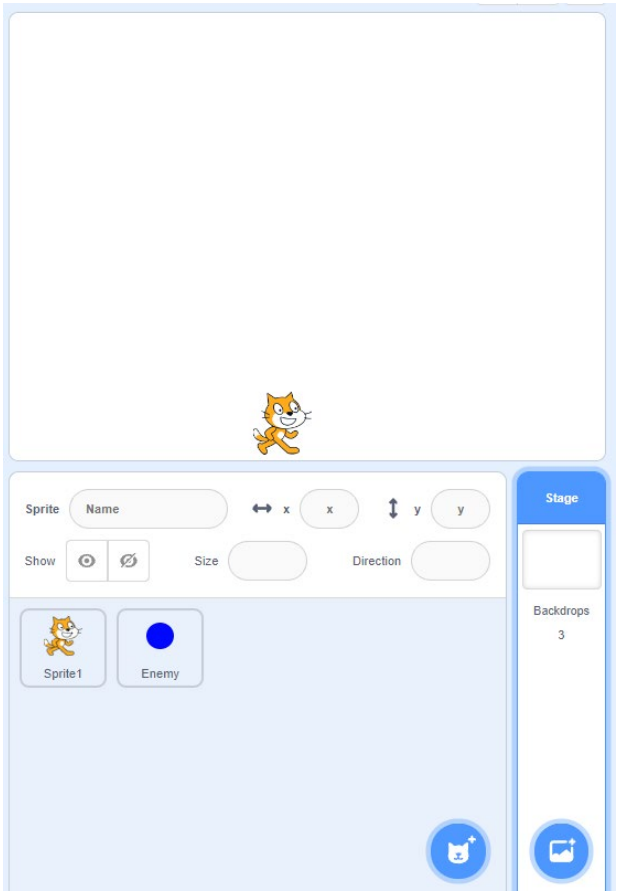


18 | Now we can add in our ”win” conditions with two more scripts in the enemy sprite.

This is the second “when I start as a clone” script I was talking about. This script is constantly checking to see if a clone has touched the edge and if it has, we broadcast a message and stop all the other scripts in the project, because the game is over. Let’s talk about what a broadcast message is. It’s a way to cause one sprite to trigger another sprite to do something. We can use this to chain multiple things together and control the order in which they happen. We can create a broadcast message by clicking on the drop-down menu and selecting new message. Create two new messages, one called “win” and the other called “lose”



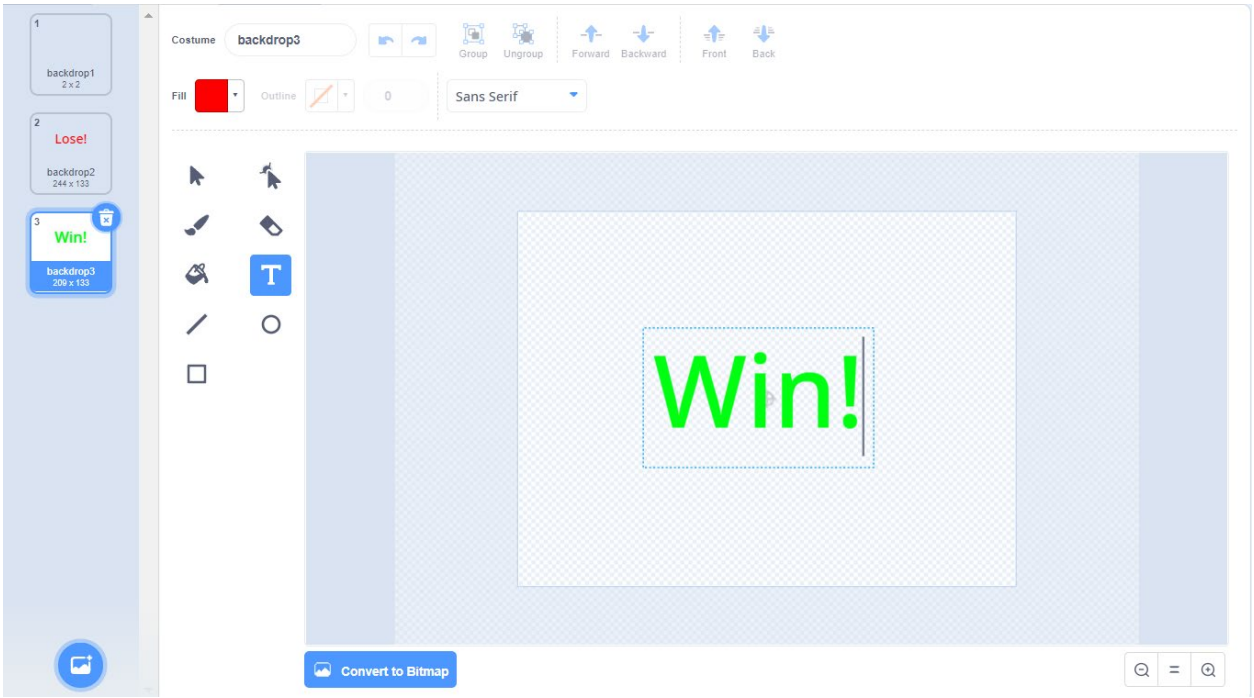
19 | This other script does something similar to the one above, except we aren’t running it as a function of a clone because we don’t need to. This script waits for the number of enemies to equal 0 and then broadcasts a message. We have the “wait 1 second” block because we start off with “number of enemies” set to 0 and we don’t want to win before the enemies are spawned in .



20 Now let's go add some functionality to those broadcast blocks. Go the stage on the right side and click on it.

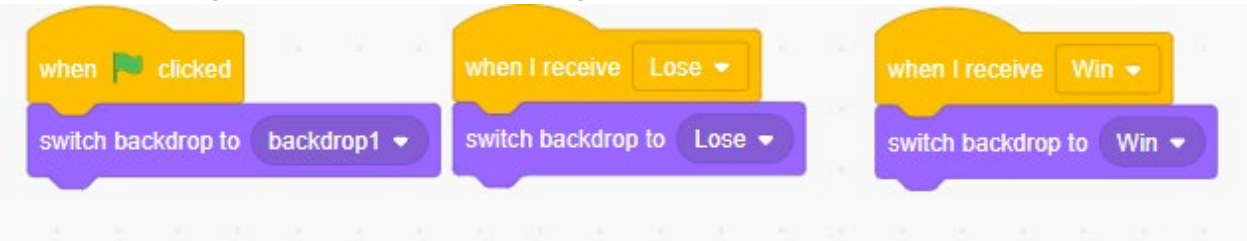
21 Now we can add some code into the stage. First we need to create some new backgrounds that tell us whether we lost or won. Go to the backdrops tab, and paint two new backgrounds. Using the text tool (It's the letter T), create one background that says "win" and one that says "lose".

Like this:



22 Now click back into the code tab and add these three simple scripts.

When the green flag is clicked, we switch back to the blank background. When we receive the "lose" broadcast message, we switch to the lose background and when we receive the "win" broadcast message we switch to the win background.



And that's it, we now have a functioning version of space invaders.