Implementing Algorithms in Scratch

Before you get started

The program that you are about to use is a free, visual programming language called Scratch, from the folks at MIT.

You can download it from the Scratch web page:

http://scratch.mit.edu

You can upload your creations here, and view the many creations already there. It’s a great way to learn new tricks and see some cool stuff!

How does Scratch work?

It’s very simple! First, start Scratch. On the left-hand side you have lots of colourful tiles. These tiles can be dragged into the middle and snapped together to form scripts. Think of a script as a series of steps, where each tile represents one step. Each script as a whole represents an algorithm.

On the upper right-hand side you have the stage. This is where your scripts come to life! And below that, you have your sprite window. For example, when you start Scratch, you will see a picture of a cat sprite. Each sprite can have its own script. By making them each do something interesting, you can write a program!

How can I use these instructions?

These instructions are broken down into three levels (columns). The first level gives you a general description of the task. If you like a challenge, this is the place to start. The second step, just in case you get stuck, gives you a bit more information. The third step, will give you even more detailed instructions, just in case you get really, really stuck!

Complete this exercise at the beginning of your lab. Once you’re done, you can upload your miniature ocean to the Scratch website. That way you can download it at home.

Then, you’ll be writing your own algorithms and implementing them in Scratch! We’ve provided some suggestions for you at the end.
Step 1: Create a background

Click on the picture of the folder on the right-hand side and find an appropriate aquarium background.

Notice that it has appeared in the sprite window. Now you can write a script to go with it.

To start a new script, select the start tile from the Control menu.

Since we’re in the ocean, let’s add some bubbling noises. (Note: sounds won’t work on the lab computers)

The ocean never stops bubbling, so this sound should probably always play while your program is running.

Go ahead, and give it a try!

At the top you will see a drop-down menu. Select the arrow to its right and choose Media > Backgrounds.

Find the sounds tab and click Import. Select Effects and find the bubbles.

To have something play forever, you’ll need to exercise some control. Click the control menu on the left-hand side. Find the tile that says “forever” and drag it to your workspace.

Click the green flag.

Find the one that looks like the aquarium! (Media > Backgrounds > Nature)

Drag the flag-start tile to your workspace. Now, whenever you click the green flag, everything attached to this tile will run!

This will add the bubbles effect as a possible sound that can be played by the current sprite.

Notice how there is a gap in the middle of forever. Everything that is connected within this gap will play forever, or until you stop the program. Next click on the sound button. Find the tab that says “play sound … until done”. Drag it to the middle of “forever” and select Bubbles.

Click the stop sign to stop.
Step 2: Add some fish

Click on the picture of the folder on the right-hand side.

Like the background we just did, we need to write a script for our fish. This script will be connected to your fish sprite only (completely separate from the background script).

Now we need to make sure that the fish appears at the start of the program.

We're also going to make sure that the fish always appears in the same spot.

Let's get this fish moving. Like the bubbles, this fish will keep moving at all times.

Finally, we don't want the fish to disappear off the side of the screen, so we need to fix that!

Last, ensure the correct orientation when the fish “bounces” off the wall.

At the top you will see a drop-down menu. Select the arrow to its right and choose Media > Costumes.

Under “Looks” select “show”. This will set the sprite to visible at the start of our program (it’s important because later we’ll make the little guy disappear!)

Look under the Motion menu on the left.

Click the Control menu on the left-hand side.

Click the Motion menu on the left-hand side.

Check out the Motion menu.

Select the sprite and click the double arrow beside it in the workspace.

Find one that looks like a fish! You’ll want to resize the fish. Click on shrink, and then click on your fish (each time you click shrinks it a bit more).

Drag the flag-start tile to your workspace. Now, whenever you click the green flag, everything attached to this tile will run!

Drag “show” to the workspace and connect it to your program.

Select the “go to x:__ y:__” and connect it to “show”. Type in the x and y coordinates. Note that the top of the workspace shows the current x-y location of the mouse.

Find the tile that says “forever” and drag it to your workspace so that it connects to the flag-start tile.

Select “move 10 steps” and drag it to the middle of “forever”.

Select “if on edge, bounce” and add it to your program.

Hit and test it out!
Step 3: Add a shark

Now let’s add a predator to our tank!

Let’s copy our fish script into our shark script. Since their basic behaviour will be the same, this will save us some work.

This shark is hungry-- let’s make it eat the fish if it gets too close.

What happens when the shark is touching the fish? It should open its mouth and the fish should disappear.

Opening its mouth is as simple as switching to a different costume that has an open mouth. First we need to import the costume.

Check out the Looks menu and add this to your program.

Return to the sprite selection (see Step 2 for help)

Click on the fish sprite and drag its program over to the shark sprite and let go.

First we need to check to see if the shark and the fish sprites are touching. Use an if statement block to ask.

Find one that looks like a shark!

If you do it correctly the program will re-appear on the fish. And when you click the shark, a copy of the program will be there.

Under the Control menu drag the if-structure to your workspace.

Notice the shape of the missing part. Select the Sensing menu and drag over “touching ___?”.

Then use the drop down menu on this new piece to select the fish sprite.

Select the shark and then click the Costumes tab. Click Import and select the desired sprite.

Choose “switch to costume” from the Looks menu. Use the pull-down menu to select the new shark costume. Drag this into your if statement block. (Don’t forget to switch back to the original costume.)

Now add your if-statement to your script.
Step 3 continued

You need to add an if statement to your fish so that it disappears when the shark eats it.

Step 4: (optional) Some extra challenges

Here are some ideas to improve your fish tank:

1. Add some more fish

2. You might notice that the fish all disappear very quickly. Let’s modify the program so that our fish regenerate after some time has past. This will make our aquarium more interesting.

Check out the if statement you wrote for your shark – this one will be similar.

Check out step one if you forgot how.

After a certain amount of time has passed, change the fish back to visible. You’re going to need a way to tell the shark, however, whether or not the fish is currently visible or hidden. It wouldn’t look right to eat an invisible fish!

If your fish is touching the shark, it should wait for a small amount of time so that the shark has time to “eat” it, and then it should disappear.

You can ask for more than one condition to be true in an if-statement by using the “___ and ___” under the Numbers menu.

What might be another question you want to ask?

You can create your own variables under the Variables menu. Variables allow you to store information, such as whether or not something is true (you can think of 1 for true and 0 for false, for example).

Think about how you might use a variable to help you out here. Do some exploring to see what tools you have available to you, and don’t be afraid to ask for some help!
Step 5: Your Task

You will be writing your own Scratch program. Be creative!

1. Your program must use at least three sprites. Explore Scratch to figure out which sprites you want to use.

2. Decide what you’d like your sprites to do, and write an algorithm for each sprite.

   Your program must contain at least two loops (the repeat and forever blocks from the Control menu) and at least two if-statements.

3. Implement your algorithms in Scratch. Ask for help if you get stuck.

You can create whatever you want, but here are some suggestions in case you’re stuck

- a dog chasing a cat who’s chasing a mouse
- a dancing and/or singing sprite (you can draw your own sprites if you want)
- dodgeball (mouse x and mouse y from the Sensing menu can be used to detect the current position of the mouse)

Hit and test it out!