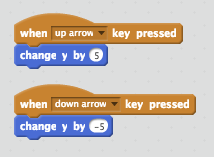
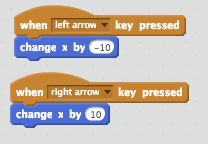
**CANADARM2 Project Instructions**

[Open Canadarm 2 – Final Scratch project](file:///E:\TLLP\Canada%20Learning%20Code%20Week\Canadarm2\Open%20Canadarm%202%20–%20Final%20Scratch%20project), [**https://scratch.mit.edu/projects/152052443/**](https://scratch.mit.edu/projects/152052443/)

Open Canadarm 2 – Starter/Initial Scratch project, [**https://scratch.mit.edu/projects/153481885/**](https://scratch.mit.edu/projects/153481885/)

* Review the project Sprites and backgrounds.
* Click on the ***Arm Sprite*** to begin. This sprite has **6 scripts** that allow it to extend the arm in four directions, grab the shape modules, and return to add them to the shape ports at the ISS (International Space Station). The first script we’ll make sets the start position for the arm on the ISS (location x = 40 and y = 50) and costume for the arm (closed). It starts with the ***Events Block***, when green flag clicked, sets the costume with the ***Looks Block***, switch costume block, and the position with a ***Motion Block***, go to x & y. Click on the green flag to test out the coding.   
  
* The next **two scripts** control the **arm moving up and down**. Y coordinates move the sprite vertically. Both scripts begin with an ***Events Block***, when key pressed and have a motion change y by block to move the arm either up (change y by 5) or down (change y by -5) depending on whether the up or the down arrow is pressed. Click on the arrow keys to test out the coding.



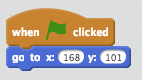
* Do the same thing for the x co-ordinates. The next **two scripts** control the **arm moving left and right**. X coordinates move the sprite horizontally. Both scripts begin with an ***Events Block***, when key pressed and have a motion change y by block to move the arm either right (change x by 10) or left (change x by -10) depending on whether the left or the right arrow is pressed.  Click on the arrow keys to test out the coding.
* 
* This **sixth script** controls the **arm grasping** (opening and closing). To do this, begin with an ***Events Block***, when key pressed block and click on the black arrow for the pop-up menu. Select space for the space bar.

Add to this event block a ***Looks Block***, switch costume to change the arm to **Canadarm- open** so that the arm looks as though it’s grabbing.

Then have an ***Events Block***, broadcast block that sends a message. Click on **New Message** and title it **Open**.

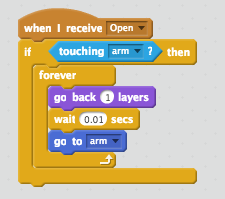
Then add another ***Looks Block***, switch costume to **Canadarm – closed**. If you test it now you probably will not be able to notice the change in the arm because the computer completes the action so fast. So put in a ***Control Block***, **wait 1 second** and you will be able to see the arm open and close. Click the space bar to text the coding.    


* Switch to the ‘**pentagon**’ sprite to add the next set of scripts. For all the shapes we need **3 scripts**. In the first script add the ***Event Block***, **when green flag clicked**. Then add a ***Motion Block***, go to, **x: 168 and y: 101**. This will move the pentagon to a position in space at the start of each game.

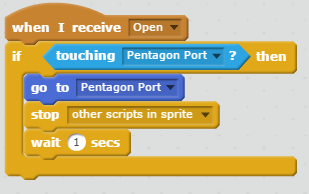


* The second of the 3 shape scripts, add an events block, **when I receive the OPEN broadcast**.

Add a ***Control Block***, **IF/THEN**. The IF/THEN block sets the condition that IF the arm is touching the shape, THEN the next instruction or set of instructions (blocks) are to be completed. In this case the instructions are with a ***Motion Block***, go to arm. The result will be IF the shape and arm are touching THEN the shape will go with the arm and both the arm and shape will move to the shape port.

In order for the shape and the arm to stay connected the condition of IF/THEN touching must be checked over and over again. That is why a ***Control Block***, **forever loop** is used. If you want to see what happens without the forever loop don’t include it. It looks like the shape is not really inside the arm. In Scratch we can change the layering of one sprite on another and in this case if we use a ***Looks Block***, go back 1 layer, it will look like the block IS inside the arm. Add a ***Control Block***, wait 1/100th of a second, and finally a ***Motion Block***, go to the arm. This sequence is what allows the shape module to attach and appear to follow the arm.   


* In the last sequence (3 of 3) begin with an ***Event Block***, when I receive broadcast block. Click on the black arrow for the pop-up menu and select OPEN. , next we have a control forever loop. Inside our loop this code repeats forever, we have another control if/then block. We are checking IF the shape is sensing touching the correct port. IF it the next sequence is run. In this sequence we have a motion go to block to have the shape go to the correct port, control stop scripts block, and finally a control wait block.



* Add the same code for each of the shapes. Click on the top block and drag it over to the sprite area. Hover it over the other shape and then release. Then click that sprite so that the Blue Box is on that Sprite. You will see that the blocks have copied into this sprite. You will need to change the ***Control Block***, IF/THEN, touching to the matching shape port and the ***Motion Block***, go to matching shape port.
* Test out your game out! How could you extend the game to make it more complex? Can you add a timer element to your game?