

# ZERO ROBOTICS

---

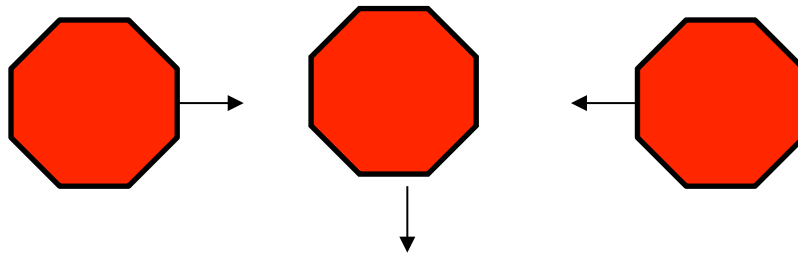
ISS PROGRAMING CHALLENGE

## More Simple Arrays and the setAttitudeTarget Function





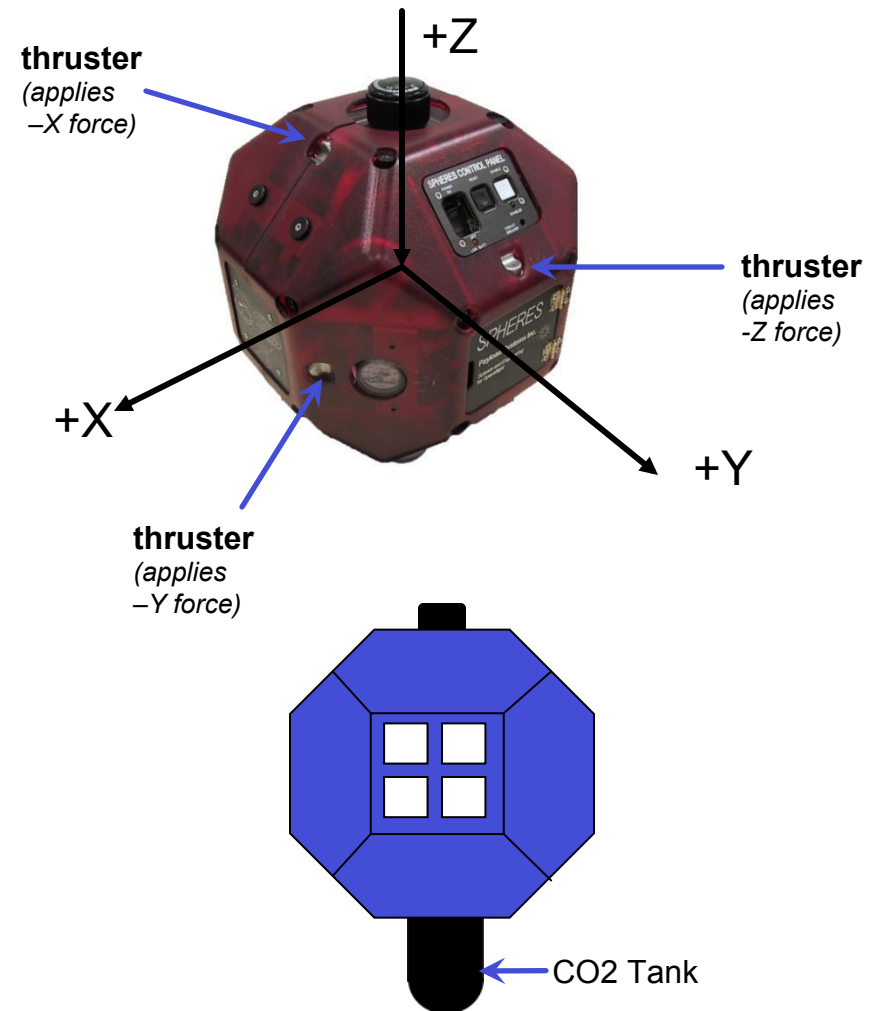
- In this tutorial you will:
  - Practice using arrays in programming
  - Learn about a new SPHERES control function:
    - `setAttitudeTarget`—allows you to rotate the satellite to face in whatever direction you want.



# What makes a SPHERE move ?



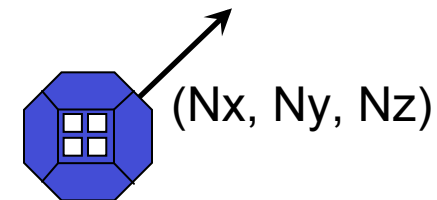
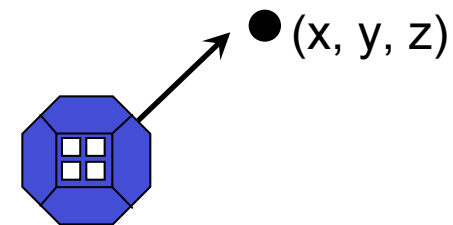
- A thruster is used to propel (move) the SPHERES satellite in a certain direction.
- There are 12 thrusters on each SPHERES satellite to help it move in 12 different directions.
  - 3 of the 12 thrusters are visible in the photo to the right.
- How does this work?
  - A tank of carbon dioxide ( $\text{CO}_2$ ) gas is attached to the SPHERES satellite.
  - Each thruster releases  $\text{CO}_2$  from the SPHERES satellite, creating a force on the satellite in the opposite direction.
- **Multiple thrusters on different sides are activated to rotate the satellite to a specified pointing direction**



## Ready to program?



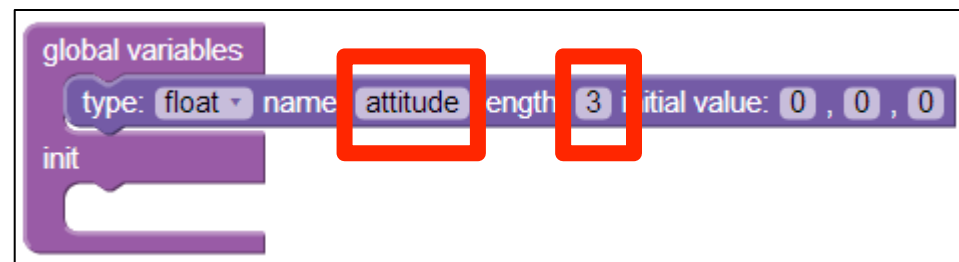
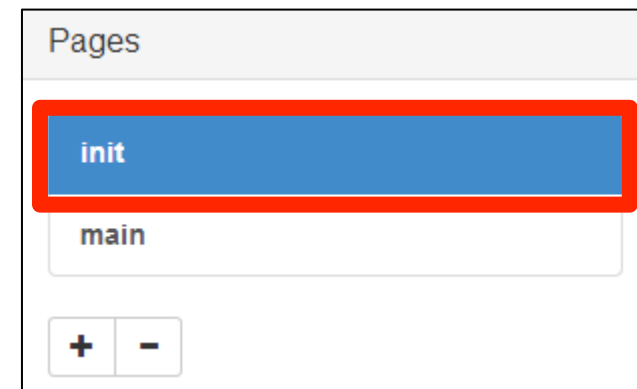
- Are you ready to write a program to rotate a satellite (control satellite attitude)?
- When you set the **position** of the SPHERES satellite, you created an array of 3 values  $\{x,y,z\}$ .
- To rotate (control the **attitude**) of the SPHERES satellite you will also need an array of 3 values  $\{Nx,Ny,Nz\}$ .
- Remember what you learned about arrays before?
- Okay, let's get started



# Create a New Project and a New Variable



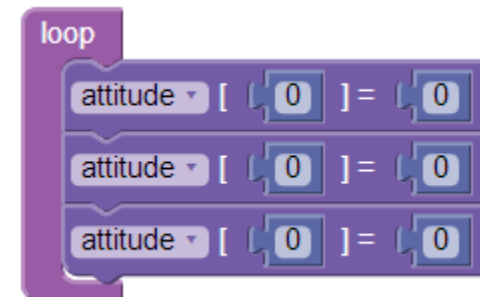
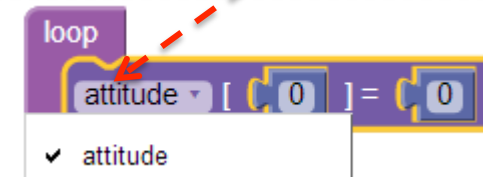
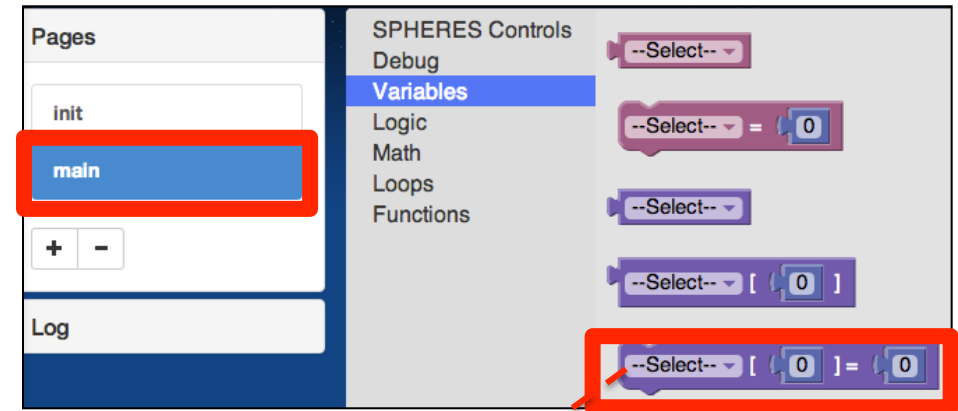
- Select “IDE” from the menu at the top of the page
- Select “New Project”
  - Project Name: **Project 3**
  - Editor: Graphical Editor
  - Game: FreeMode
- Click “Create Project”
- Go the the Init page
- Declare an array called “**attitude**” to set the attitude of the SPHERES satellite as follows:
  - Go to the **Init** page
  - Drag the array initialization block from the Variables accordion
  - Name: “attitude”
  - Length: 3
  - Leave initial values at 0, 0, 0



# Assign Values to Your Array



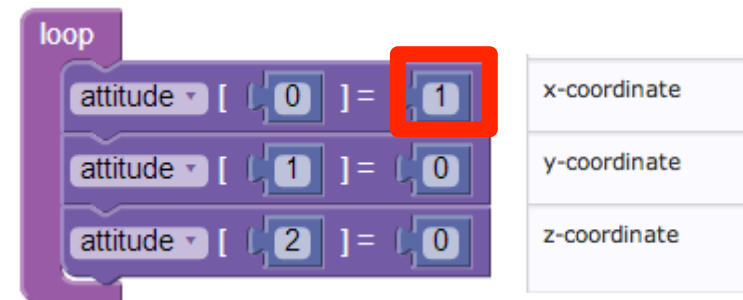
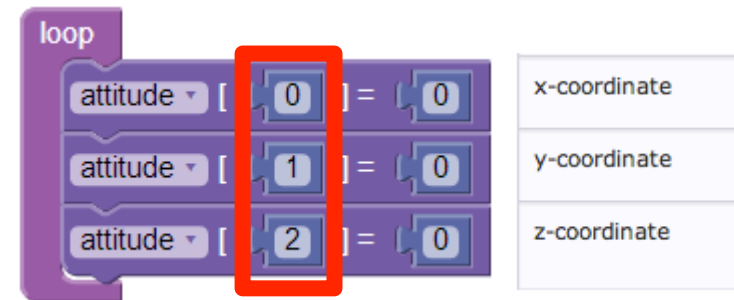
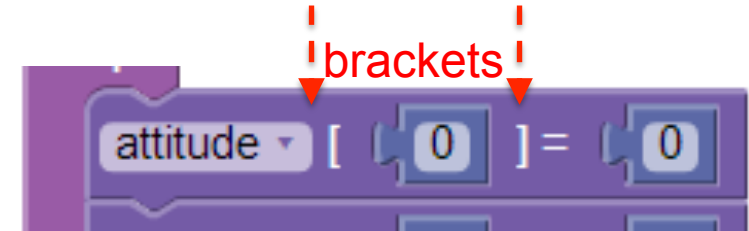
- Go to the main page to assign values to your array
  - Click on the Variables accordion
  - Drag the purple “--**Select**-- [ 0 ] = 0” (array assignment) block into the **loop**
  - Use the drop down menu to select “**attitude**”
  - Repeat 2 more times
- (You need 3 attitude [0] = 0 blocks since your array has 3 members)



## Assign Values to Your Array, cont.



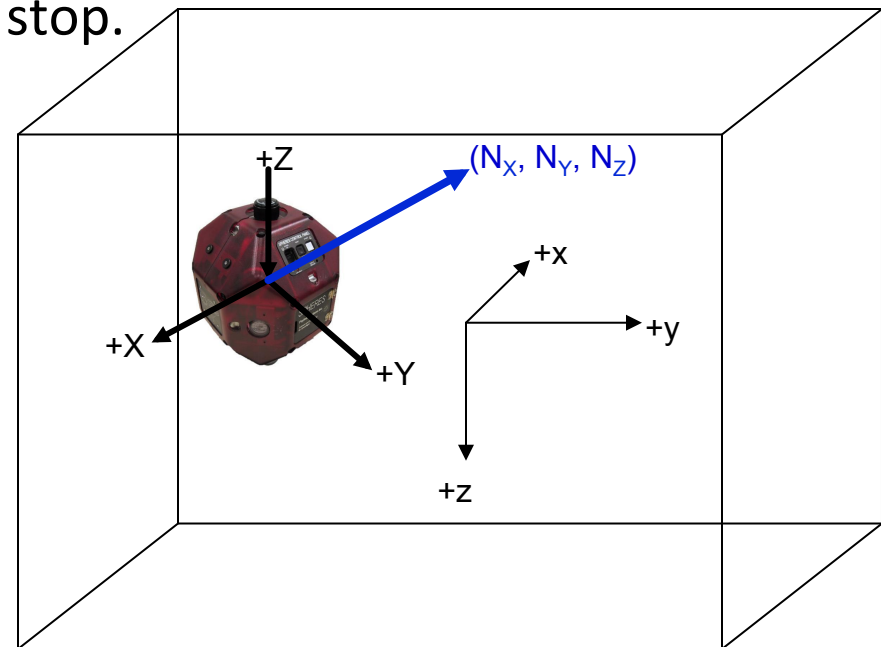
- Change the **[bracketed]** part of each “attitude[0] = 0” block as follows:
  - In the first (top) block, leave “0” selected (This will represent the “x” direction)
  - In the second block change to a “1” (This will represent the “y” direction)
  - In the third (bottom) block change to a “2” (This will represent the “z” direction)
- In the first (top) block, change the right-most “0” to “1.0”



## setAttitudeTarget



- The SPHERES Control Function “setAttitudeTarget” allows you to **set the direction for the satellite to point its Velcro (-X) face**
- Attitude specifies a pointing **direction** ( $N_x, N_y, N_z$ ) , not a pointing **location**.
- Commanding an attitude target makes the satellite fire thrusters to rotate to the target direction, then stop.

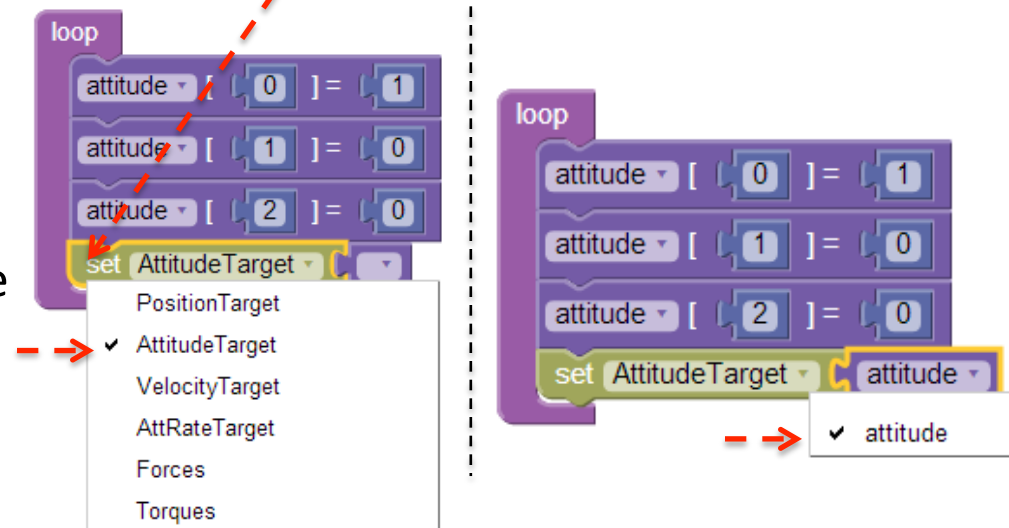




# Add setAttitudeTarget Function



- Create a statement to set the attitude of the SPHERES satellite
  - Click on the SPHERES Controls accordion
  - Drag the setPositionTarget block into your loop
  - Using the first dropdown menu, change **PositionTarget** to **AttitudeTarget**
  - Use the second dropdown to select **attitude**
- This program you've created tells the SPHERE to move to the target attitude defined by the array "attitude[3]"
  - The Velcro face of the SPHERES satellite will rotate to point in the positive x direction



# Compile, Simulate



- Compile, Simulate
- In the Simulation Settings pop-up box:
  - “Maximum Time”:
    - **Change from 90 seconds to 20 seconds**
- Click “Simulate” button
- NOTE: Before playing the simulation
  - Click on the zoom-in tool at the bottom of the screen 10 times

Zero Robotics About ZR Tourna

File Edit Simulate Help

Pages

Log

Type here and pre

Simulate

Simulate As ☒ Satellite 1 (Blue) ☐ Satellite 2

Opponent No Opponent Select

Maximum Time (s) 20

Initial Position	X	Y	Z	AttX	AttY	AttZ
Satellite 1	0	0.5	0	0	1	0
Satellite 2	0	-0.5	0	0	-1	0

Reset All

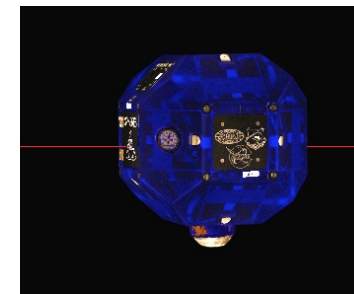
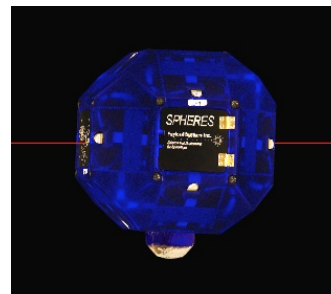
Cancel Simulate

Play Stop 1x 2x 4x 10x Zoom In Zoom Out Refresh Reset Back to Project

## View Simulation



- Start the simulation
  - The visible face on the SPHERES satellite will change as the satellite rotates to point in the positive x direction.
  - Look at the scoring box (top-left corner of the screen with blue label) which provides information about the blue SPHERES satellite:
    - Started at  $N_y = 1.00$
    - Ended at  $N_x = 1.00$



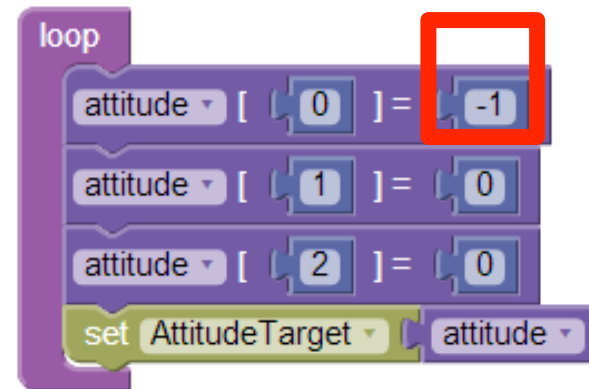
attitude								
X: -0.01	Y: 0.50	Z: 0.01						
Vx: -0.003	Vy: 0.003	Vz: 0.005						
Nx: -0.01	Ny: 1.00	Nz: 0.04						
$\omega_x$ : 0.47	$\omega_y$ : -0.66	$\omega_z$ : 9.67						
Fuel Remaining: 100%								

attitude								
X: 0.01	Y: 0.50	Z: -0.00						
Vx: 0.000	Vy: 0.001	Vz: 0.000						
Nx: 1.00	Ny: -0.09	Nz: 0.01						
$\omega_x$ : 0.30	$\omega_y$ : -0.03	$\omega_z$ : 0.81						
Fuel Remaining: 100%								

(pointing in positive y direction)  
(pointing in positive x direction)

- Select “Back to Project”
- Next try pointing in the negative x direction
- Change: “attitude[0] = 1” to:  
“attitude[0] = -1”
- Important Notes:

For these exercises, point the satellite by setting only one of the values [0], [1], [2] to +/-1 and leave the rest set to 0 as shown in the table.



## To point the satellite in the following directions:

	+/- x direction	+/- y direction	+/- z direction
set [0] =	+/-1	0	0
set [1] =	0	+/-1	0
set [2] =	0	0	+/-1

- “Compile” and “Simulate” as before
- “View Results”

## Text Version of Code

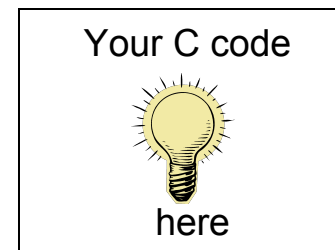


- Remember the program you wrote with “setPositionTarget” (from last time) and its C code ?



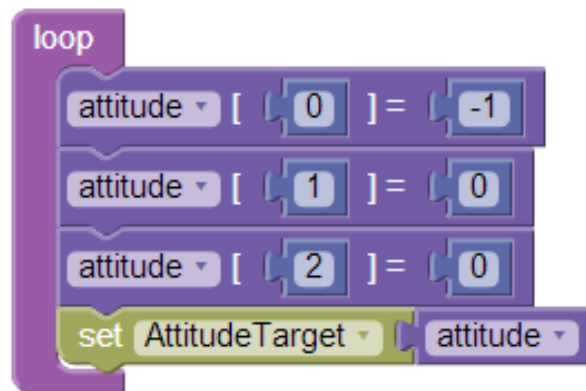
```
1 void loop() {
2   position[0] = 2;
3   position[1] = 2;
4   position[2] = 0;
5   api.setPositionTarget(position);
6 }
```

- Can you figure out the C code for the “setAttitudeTarget” program you just wrote?





- If the C code shown below matches what you thought- You are right!



```
1 void loop() {  
2   attitude[0] = -1;  
3   attitude[1] = 0;  
4   attitude[2] = 0;  
5   api.setAttitudeTarget(attitude);  
6 }
```



- Congratulations!
- You are getting good at programming with arrays!
- You know how to program a SPHERES satellite to rotate and point in a specific direction!

