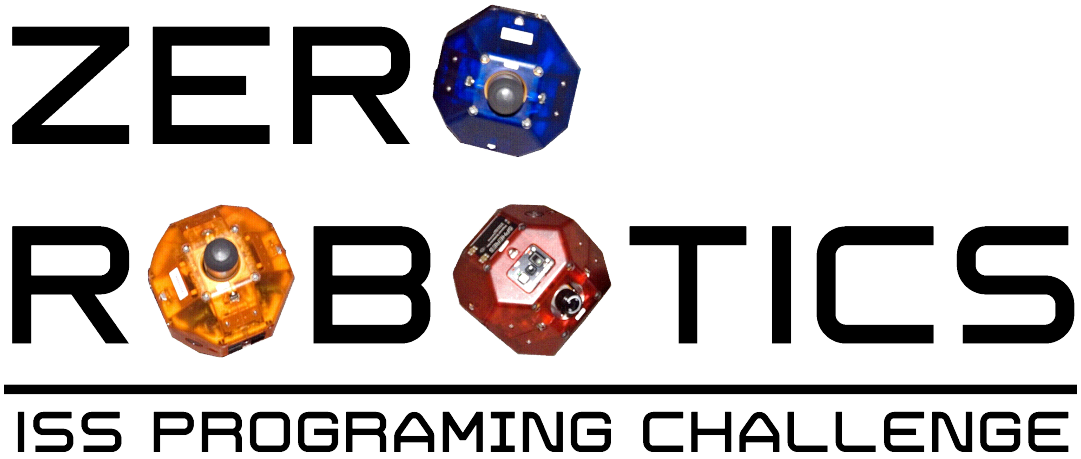
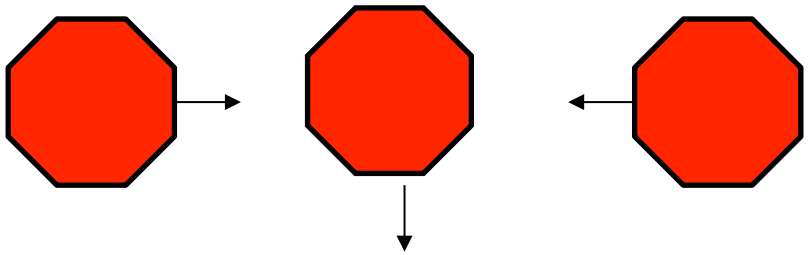


More Arrays and the setAttitudeTarget Function

Goals

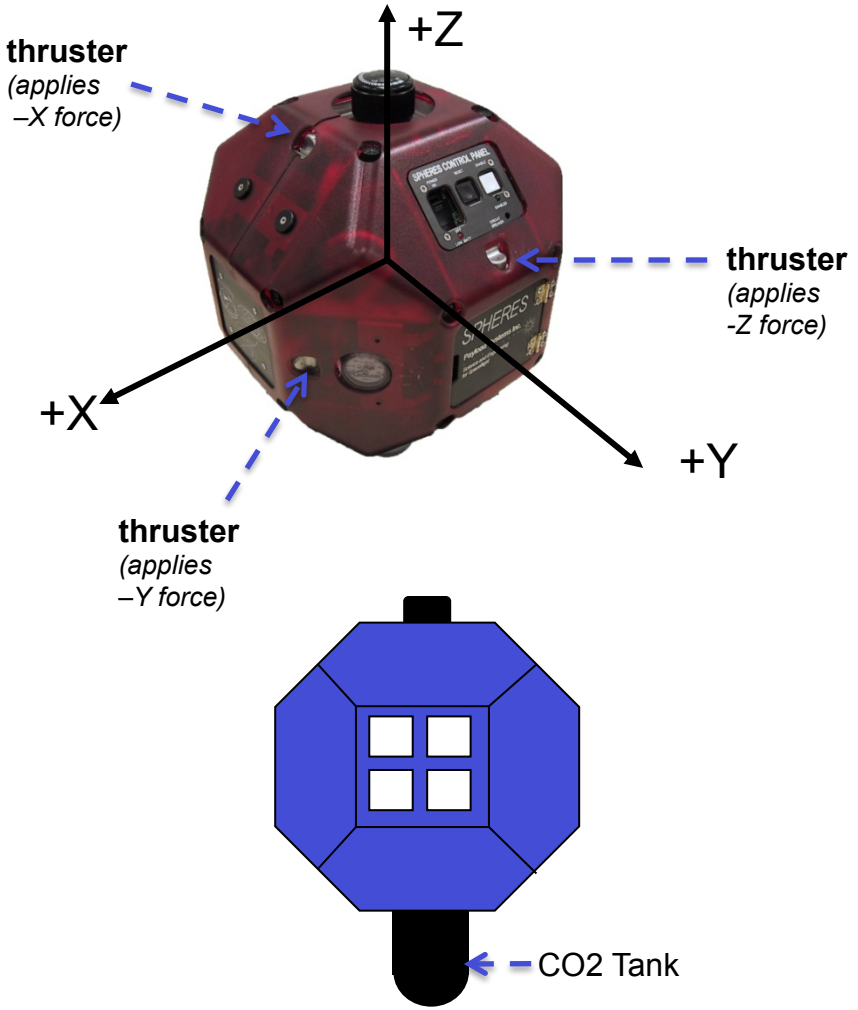
- In this tutorial you will:
  - Practice using arrays in programming
  - Learn about a new SPHERES control function:

    api.setAttitudeTarget—allows you to rotate the satellite to face in whatever direction you want.



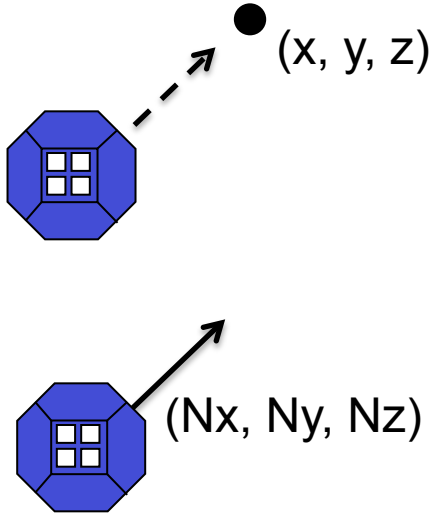
What makes a SPHERES move ?

- 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 (CO_2) gas is attached to the SPHERES satellite.
  - Each thruster releases 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**



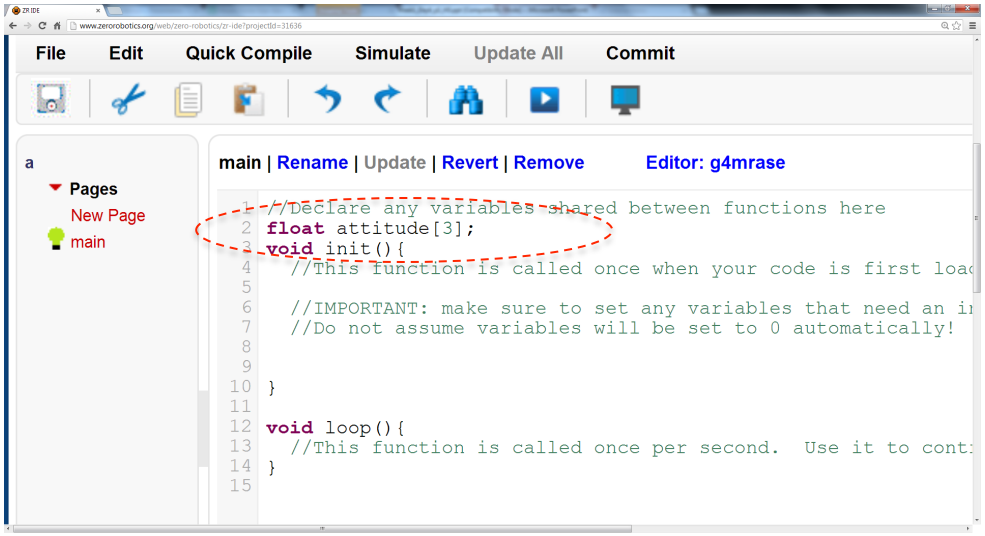
Ready to program?

- 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

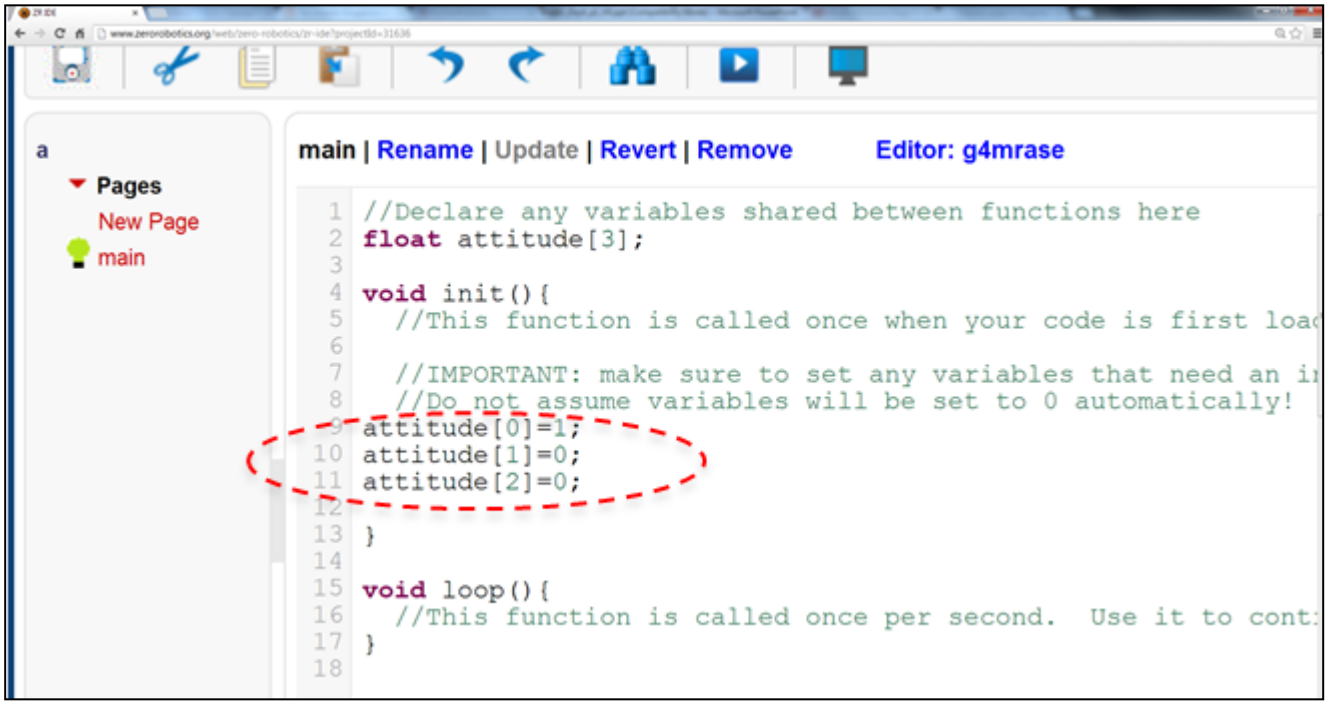
- Select light blue "ZR IDE" SPHERES icon on top ribbon
- Select "New Project"
  - Project Name: **Project 2**
  - Game: FreeMode
  - Editor: Text Editor
- Declare an array called "**attitude**" to store the attitude of the SPHERES satellite
  - Go to the area before void init() to declare the array.
  - Recall that the type will be float, and that the length will be 3 variables.



Assign Values to Your Array

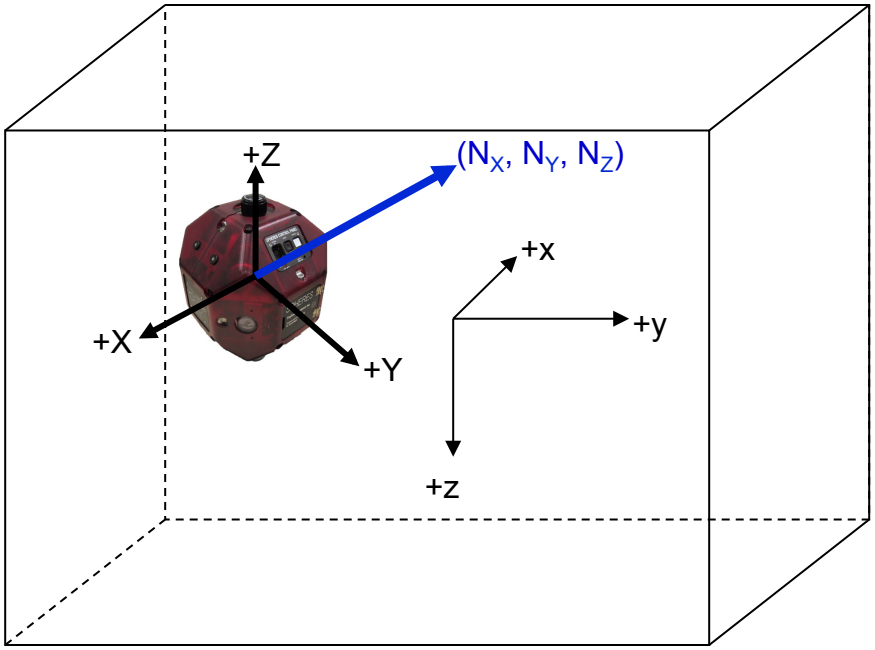
- Go to void init() and assign every element of the array a value corresponding to the coordinates (1,0,0) (Remember, the first element has the index 0, not 1).

- Don't forget the semicolons!



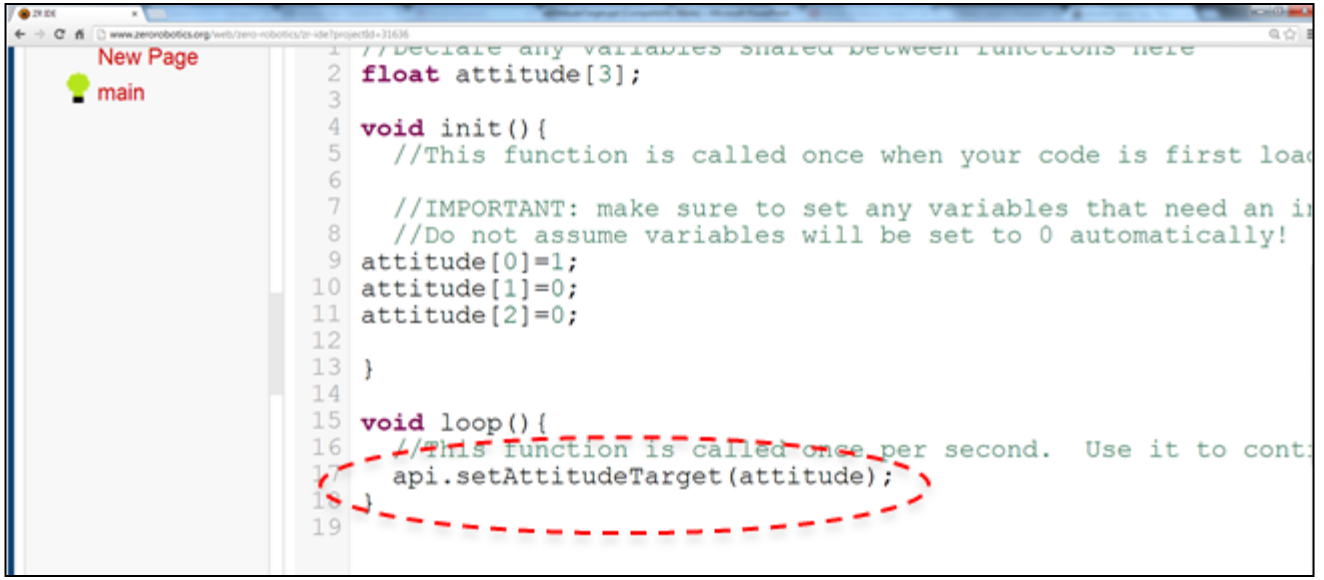
setAttitudeTarget

- The SPHERES Control Function **setAttitudeTarget** allows you to set the direction in which the satellite's Velcro (-X) face points.
- Attitude specifies a pointing **direction** (Nx, Ny, Nz) , not a pointing **location.**
- Commanding an attitude target makes the satellite fire thrusters to rotate to the target direction, then stop.



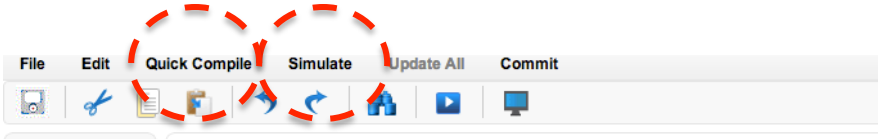
Add setAttitudeTarget Function

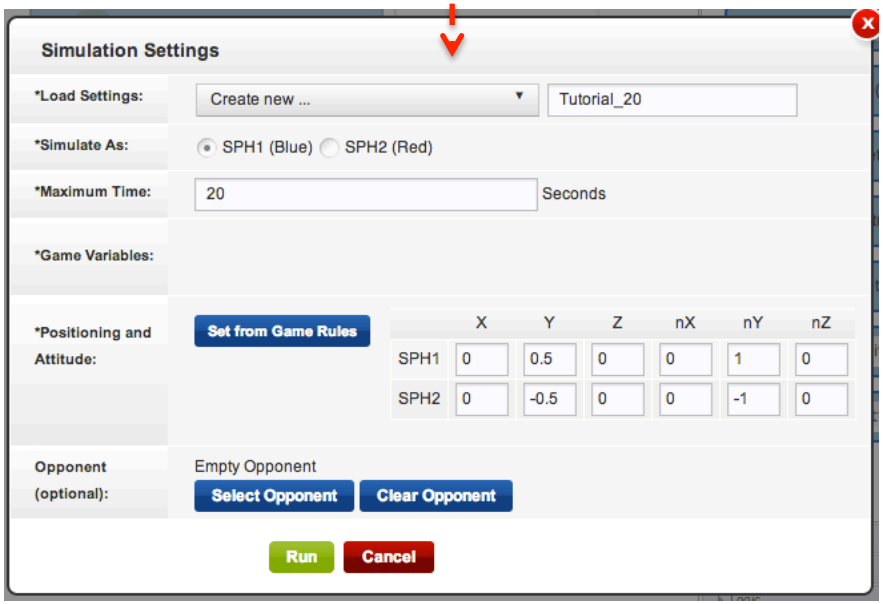
- The setAttitudeTarget control can be applied in the same manner as the setPositionTarget control.
- Go to void loop(), and put in **api.setAttitudeTarget**. Again, to designate which array the control will be applied to, put **attitude** within parenthesis and end with a semicolon.



Compile, Simulate

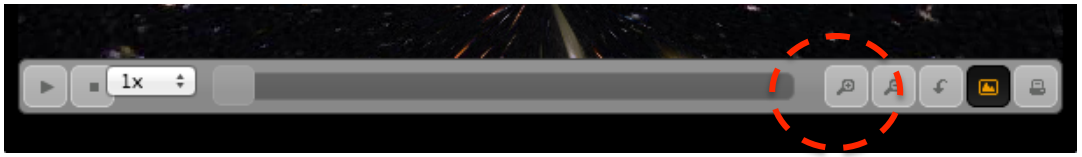
- Compile, Simulate
- In the Simulation Settings pop-up box:
  - *Load Settings:
    - Select "**Create new…**",
    - Type a settings name: "**Tutorials_20**"
    - (this simulation will run for 20 seconds)
  - *"Simulate As":
    - Select "SPH1 (Blue)"
  - * "Maximum Time":
    - **Change from 90 seconds to 20 seconds**
  - *Positioning and Attitude
    - Click "Set from Game Rules"
    - Leave the text fields alone
  - *Opponent:
    - Should be "Empty Opponent" (select "Clear Opponent" otherwise)
- Click on green "Run" button at the bottom

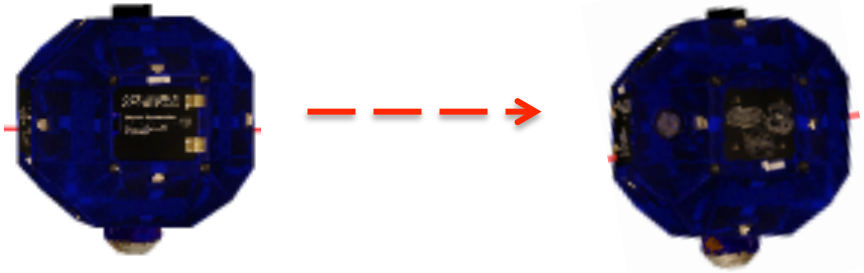


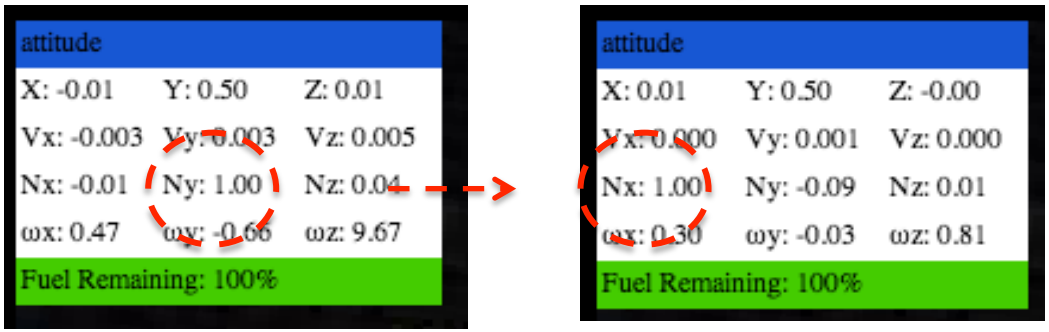


View Simulation

- Before playing the simulation
  - Click on the zoom-in tool at the bottom of the screen 10 times
- 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 Ny = 1.00 (pointing in positive y direction)
      Ended at Nx = 1.00 (pointing in positive x direction)







Modify Program, Compile & Simulate

- Close the Simulation Window
- Return to the Text Editor page
- 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.

- "Quick Compile" and "Simulate" as before
- "Run"

```
//Declare any variables shared between functions here
float attitude[3];

void init(){
  //This function is called once when your code is first loa

  //IMPORTANT: make sure to set any variables that need an i
  //Do not assume variables will be set to 0 automatically!
attitude[0]=-1;
attitude[1]=0;
attitude[2]=0;

}

void loop(){
  //This function is called once per second.  Use it to cont
  api.setAttitudeTarget(attitude);
}
```

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

Review

- 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!
- Note: the tutorial "setAttitudeTarget revisited" teaches rotation in 3 dimensions.