

ZERO ROBOTICS

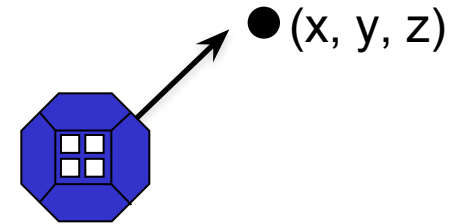
ISS PROGRAMING CHALLENGE

Introduction to Arrays and the setPositionTarget Function (Project 2)





- In this tutorial you will:
 - Practice using the ZR IDE
 - Learn about and use arrays in programming
 - Learn about a SPHERES control function:
 - setPositionTarget – sets a target position (point in 3 dimensions) for the satellite to move to and stop





- An array is a list of data of the same type.

- Examples:

`{1.2, 3.0, -2.5}`

This is an array of 3 floats (could be an x, y, z coordinate point)

`{99, 95, 82, 90, 76, 91, 93, 85, 100, 65}`

This is an array of 10 integers (could be a set of test scores)

Declaring Arrays in C



- When you declare an array, you are actually declaring a lot of variables at once.
- An array is declared by assigning:
 - variable **type** (e.g. float)
 - the array's **name** (e.g. position)
 - **number of variables** (e.g. 3)
- The variables in the array are named with the name of the array plus a number in square brackets. **The numbering starts with 0.**
- For example, in the array at the right the members are three float variables called position[0], position[1], and position[2]. The value of each variable is assigned individually.

Array declaration:

```
float position[3]
```

Member variables:

```
position[0] = 0.5  
position[1] = 0.7  
position[2] = 0.8
```

Array Declaration Question



Suppose the numbers below represent the test scores in a class.

{99, 95, 82, 90, 76, 91, 93, 85, 100, 65}

How would you declare an array to hold the scores? (What are the three things you must include in the declaration?)

What would be the names and values of the variables in the array?



The data consists of **ten integers**, so we want ten int variables.

Declaration: **int classScores****[10]**

Members: classScores[0] = 99

 classScores[1] = 95

 classScores[2] = 82

 classScores[3] = 90

 classScores[4] = 76

 classScores[5] = 91

 classScores[6] = 93

 classScores[7] = 85

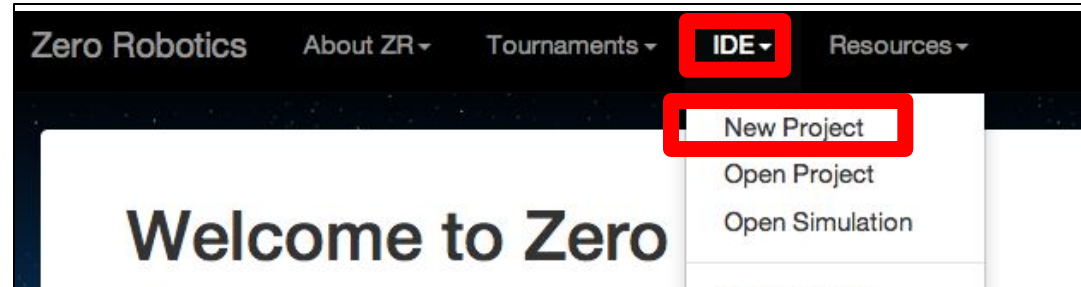
 classScores[8] = 100

 classScores[9] = 65

Create a New Project



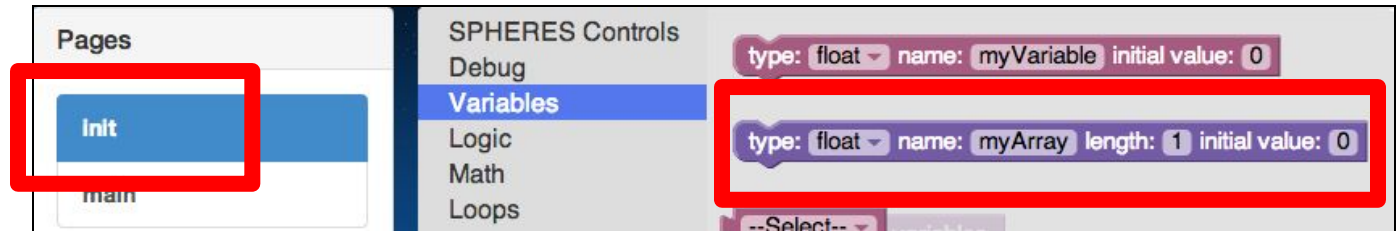
- Create a project to move the SPHERES satellite to a point with x, y, z coordinates
- Select “IDE” menu on top ribbon
- Select “New Project”
- Enter
 - *Project Name
 - **Project 2**
 - *Editor
 - Select “Graphical Editor”
 - *Game
 - Select “FreeMode” from drop down menu
- Click “Create Project”

A screenshot of the "New Project" dialog box. It contains a "Project Name" field with the text "Project 2". Below this are two radio buttons: "Text Editor" and "Graphical Editor", with "Graphical Editor" selected. Under the "Game" section, there is a dropdown menu currently showing "Free Mode", which is pointed to by a red arrow. At the bottom right of the dialog are "Cancel" and "Create Project" buttons. A red arrow from the "IDE" menu in the previous image points down to this dialog box.

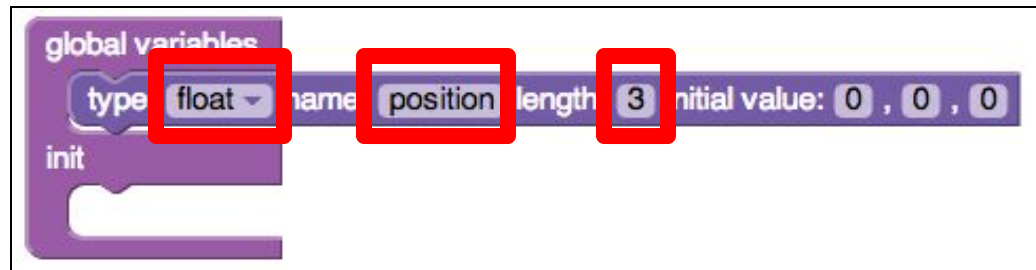
Declare a New Array



- Start your program by declaring an array to hold x, y, z position coordinates for the SPHERES satellite
 - Go to the Init page
 - Click on the Variables accordion
 - Purple blocks are used for arrays (pink are used for variables)
 - Click on the top **purple** block and drag it into the global variables loop



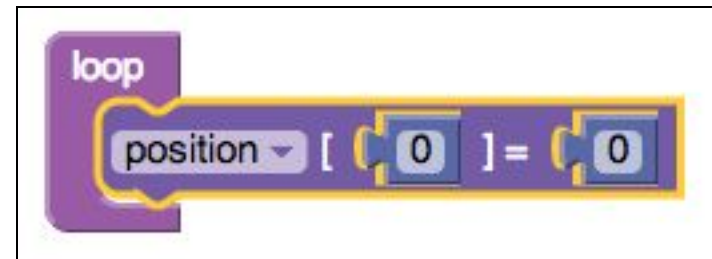
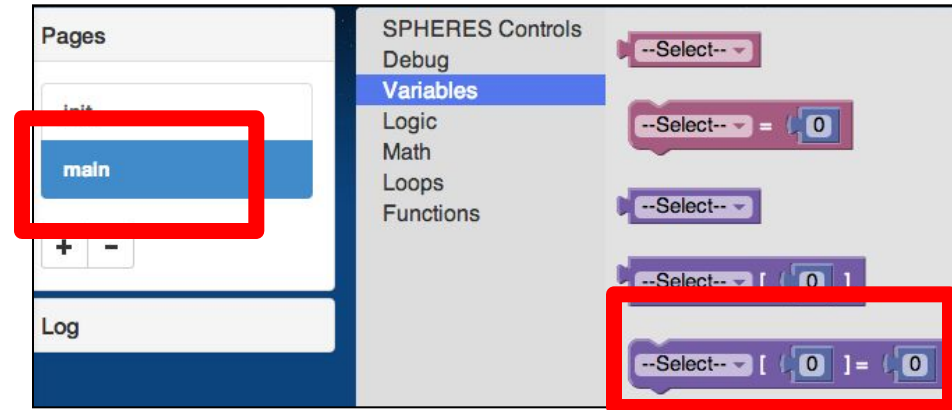
- Declare
 - type: select “float”
 - name: enter “position”
 - length: enter: 3
 - initial Value: Leave initial values=0



Assign Values to Your Array



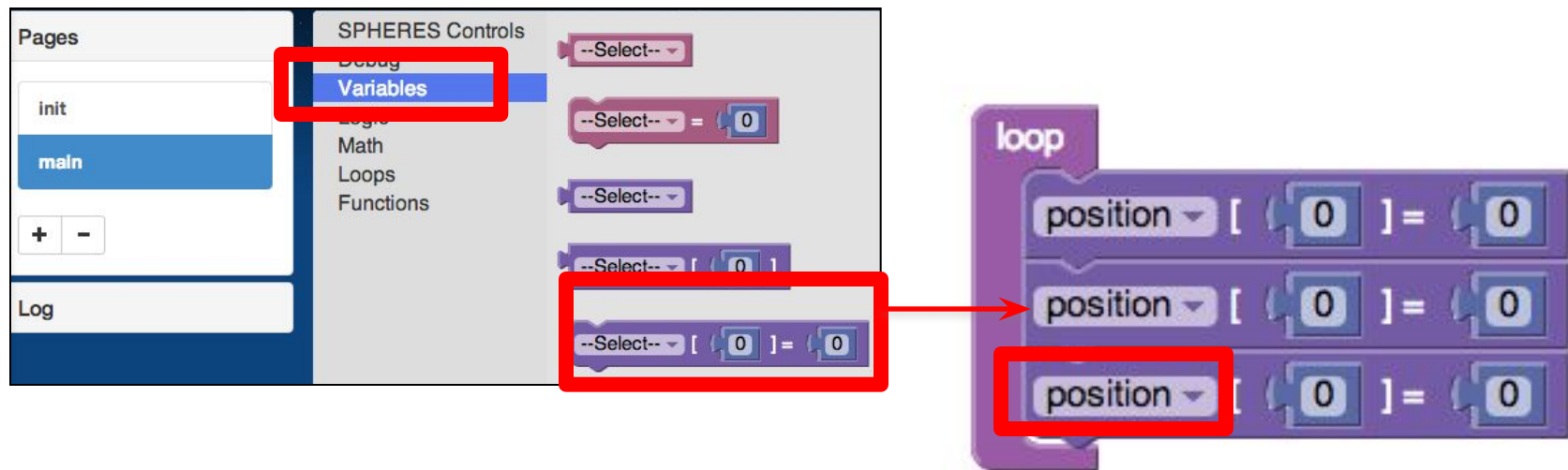
- Now you need to set the variables in your array to the coordinates you want the satellite to move to.
- We will do this on the main page
 1. Click on “**main**” page
 2. Click on Variables accordion
 3. Click on the purple “--Select--[0] = 0” block
 4. Drag and Drop the block into the **loop**
 5. Use the drop down menu to select “**position**”



Assign Values to Your Array, cont.



- From the Variables accordion, repeat drag and drop of the purple “select [0] = 0” block into the loop **two** more times
- Be sure to toggle all to “**position**”



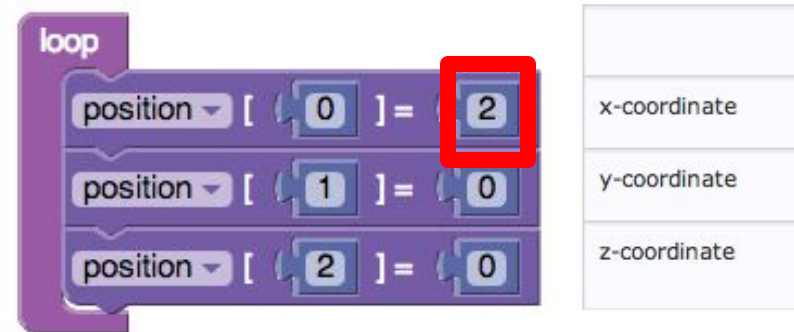
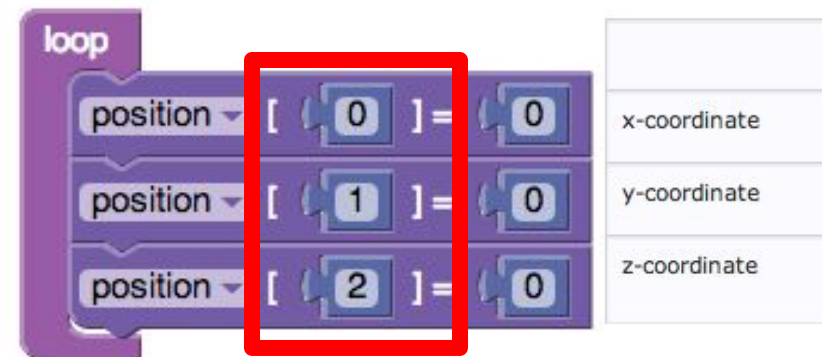
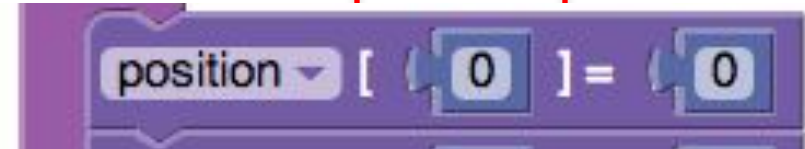
- You need 3 blocks because your array has 3 members that need to be set.

Assign Values to Your Array, cont.



- Change the bracketed part of each “position[0] = 0” block as follows:
 - In the first (top) block, leave “0” selected (This line will set the variable that represents the “x” coordinate)
 - In the second block change to a “1” (This will set the “y” coordinate)
 - In the third (bottom) block change to a “2” (This will set the “z” coordinate)
- In the 1st (top) block change the right-most “0” to “2.0”
 - This sets the target x coordinate to 2.0.
 - Leave the other coordinates at 0 for now.

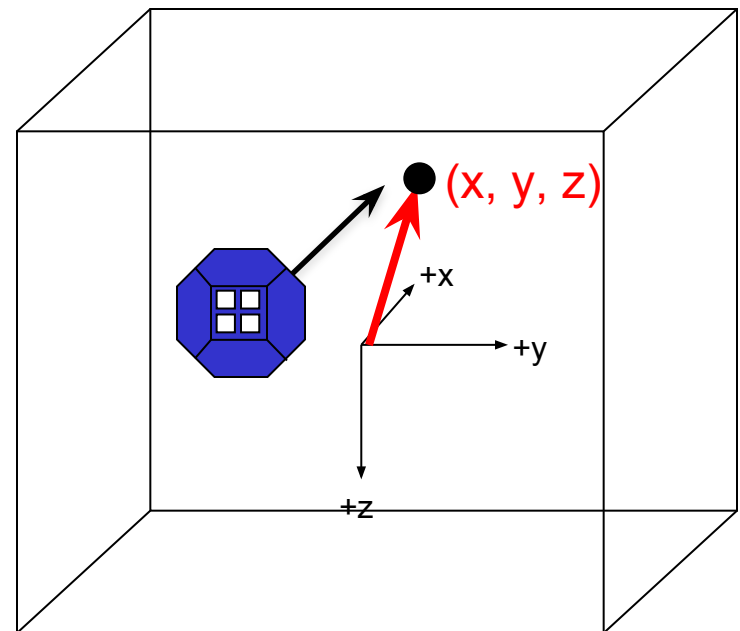
|brackets|



setPositionTarget



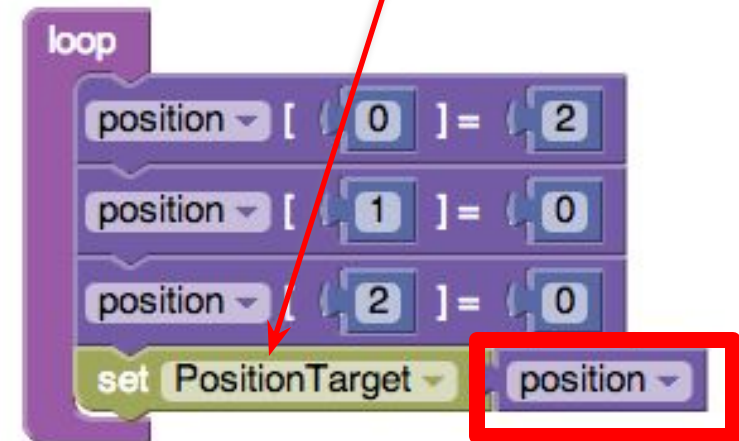
- The SPHERES control function “**setPositionTarget**” allows you to move the satellite to a target position
- The target point is input as an array of three floats that represent its x, y, z coordinates in meters.
- When a position is commanded, the satellite will fire thrusters to move to the target point, then stop



Add setPositionTarget Function



- Create a statement to set the position of the SPHERES satellite
 - Click on the “SPHERES Controls” accordion
 - Select the “setPositionTarget (--Select--)” block and **drag** the block **and drop** it inside the “loop” below the other blocks.
 - Toggle drop down menu to “position”
- This program you’ve created tells the SPHERES satellite to move to the target position defined by the array “position”
 - The satellite will move to the position $x=2, y=0, z=0$ on the x axis



Compile, Simulate



- Click on Compile (top menu)
- When complete Click on Simulate (top menu)
- The Simulation window will open
- Change “Maximum Time” setting to 60
- Click “Simulate”
- a “Running” window will pop up while the simulation is being constructed

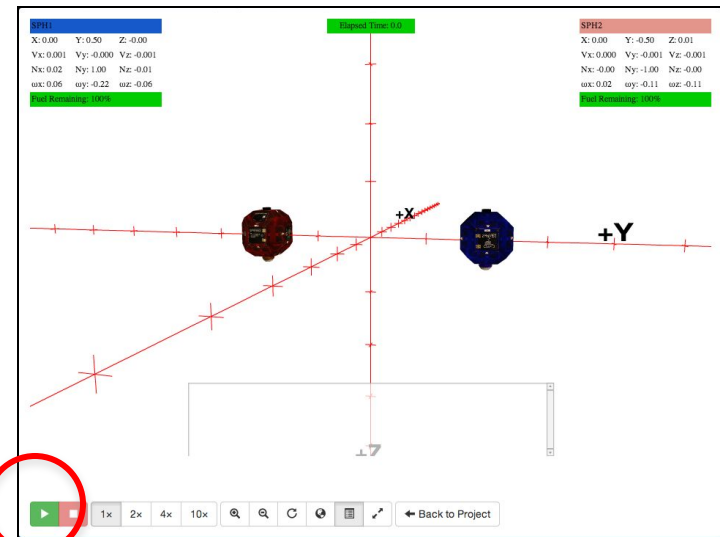
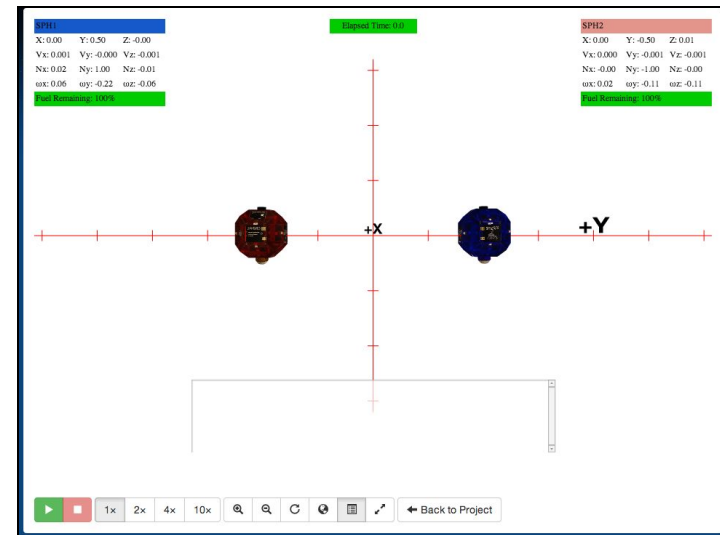
The screenshot shows the Zero Robotics web interface. At the top, there is a navigation bar with 'Zero Robotics', 'About ZR', and 'Tournament'. Below this is a menu bar with 'File', 'Edit', 'Simulate', and 'Help'. The 'Simulate' menu is open, showing options: 'Compile', 'Codesize Estimate', 'Simulate', 'Profile', and 'Submit'. The 'Simulate' option is highlighted with a red box. Below the menu, the 'Simulate' window is open. It has a 'Simulate As' section with 'Satellite 1 (Blue)' selected. An 'Opponent' section shows 'No Opponent' and a 'Select' button. A 'Maximum Time (s)' input field is set to '60' and is highlighted with a red box. Below this is a table for 'Initial Position' with columns for X, Y, Z, AttX, AttY, and AttZ. The table has two rows: 'Satellite 1' and 'Satellite 2'. At the bottom of the 'Simulate' window, there is a 'Reset All' button and a 'Simulate' button, which is highlighted with a red box. To the right of the 'Simulate' window, a 'Running...' status box is visible.

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

View Simulation



- Click “View Results” in the log window
- Remember, the initial view shows y and z axis
 - horizontal line (the y-axis)
 - vertical line (the z-axis)
- To see the x axis:
 - Click and hold the left mouse button anywhere on the background and drag the mouse until x, y and z axis are visible
- Click the “Play” arrow at the bottom left of the screen and wait a few seconds.
 - The blue satellite will start moving to coordinates (2.0,0.0,0.0), just as you told it to!





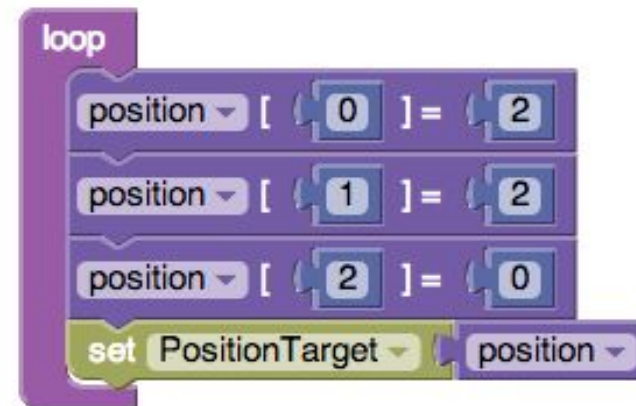
- Close the simulation window to return to the Graphical Editor page
- Make the following changes to program the satellite to move to the x,y,z coordinate point = (2,2,0)

- Change

“position[1] = 0” to:

“position[1] = 2.0”

- “Compile” and “Simulate”
- “View Results”
 - This time the blue SPHERES satellite should move to the x,y,z coordinate point (2,2,0)
- Return to the Graphical Editor page

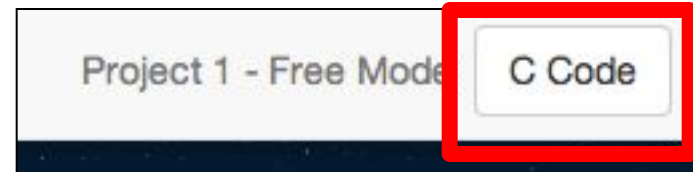


x-coordinate
y-coordinate
z-coordinate

View Program in C Code

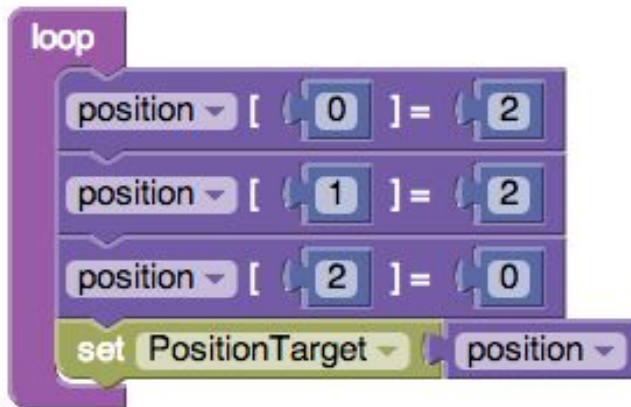


- On the right side of the top menu bar select “**C Code**”



- Compare:

Your program - versus - C Code

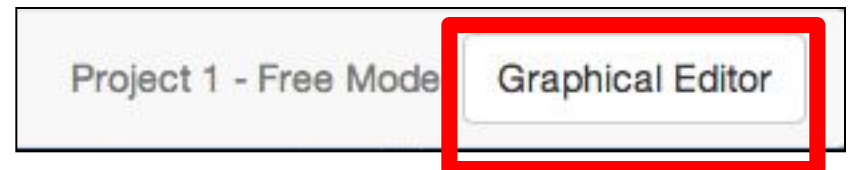


```

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

(Note: “api” stands for ZR application programming interface)

- Select Graphical Editor to return to the graphical editor





- Congratulations!
 - You have used an array to program a SPHERES satellite.
 - You programmed the satellite to move to a point in 3 dimensions!

