ZERO ROBOTICS

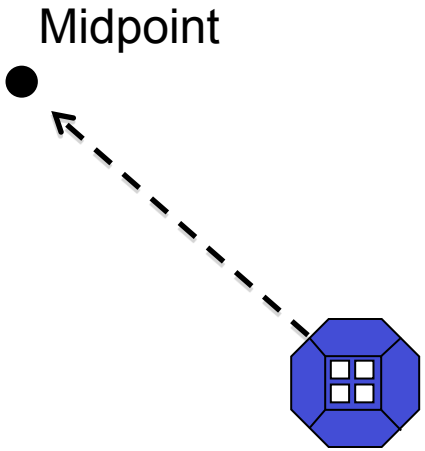
ISS PROGRAMING CHALLENGE

For Loops

Goals

In this tutorial you will:

- Use a **for loop** to repeat an action a set number of times
- Find the position of the other satellite
- Program your satellite to move toward the other satellite, but stop halfway



Create a new project

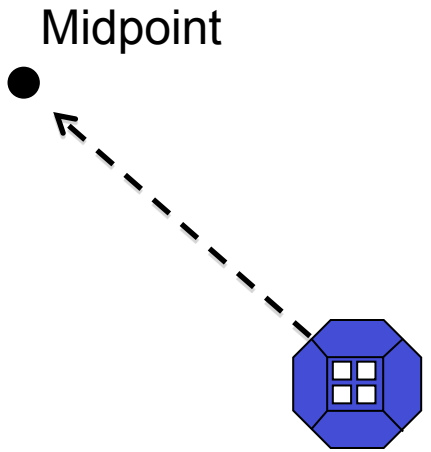
- Create a new project
- Name it “Project8” and choose “FreeMode” and “Text Editor”
- Create the following variables and arrays:
  - **`int counter`**
    - Set initial value to 0
  - **`float my_state[12]`**
  - **`float other_state[12]`**
  - **`float target[3]`**

do not set initial values

Program outline

In this tutorial, you will move your blue satellite half the distance toward the red satellite.

- First, you will use two API functions, **`getMyZRState`** and **`getOtherZRState`**, to find the starting positions of the two satellites.
- You will find the coordinates of the midpoint between the satellites.
- You will move to that position using **`setPositionTarget`**.



`getMyZRState` and `getOtherZRState`

- **`getMyZRState`** finds the position of your satellite (blue) and writes it to an array.
- The array must consist of 12 floats. The first three members (index numbers 0 to 2) contain the x, y, and z coordinates of your current position.
- The other numbers in the 12-member array contain other information about your current state (for example, your current velocity) that you will not use in this tutorial.
- **`getOtherZRState`** does the same thing, but it sets the array to the state of the other satellite (red.)

```
void loop(){
  //This function is called once per second.
  //Use it to control the satellite.

  api.getMyZRState(xxx)

  api.getOtherZRState(xxx)


}
```

Array members:

xxx[0] : x coordinate
xxx[1] : y coordinate
xxx[2] : z coordinate
xxx[3] to xxx[11]: other things

Set up counter

- Go to void loop() section and create an if-then statement using the condition that the counter is set to zero ("counter == 0").
- Putting the calculations in this statement means they will happen only once, at the start when **counter** is 0. Otherwise your target will keep changing as your position changes.
- Finally, increment counter. Add **counter++;** after the if statement as you have done previously.

```
void loop(){
  //This function is cal
  //Use it to control the
  if (counter == 0 ){
  }
  counter++;
            }
```

Set *my_state* and *other_state*

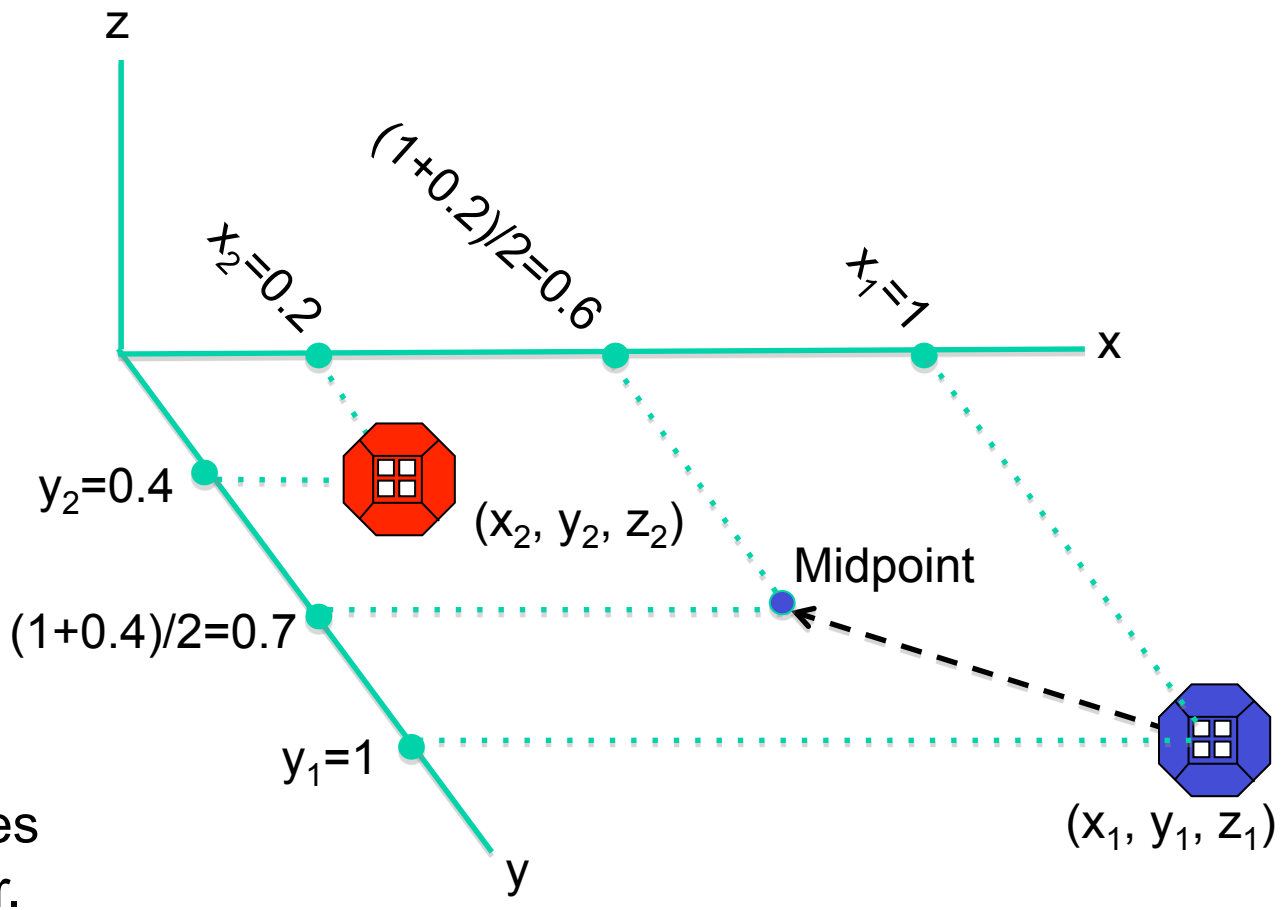
- Now you will find the positions of the two satellites so you can calculate your target.
- In the if-then statement, call the api functions **getMyZRState** and **getOtherZRState**, and write this information to the previously created arrays **my_state** and **other_state** respectively.
- The arrays **`my_state`** and **`other_state`** have now been set to the states of the two satellites.

```
void loop(){
  //This function is called once per se
  //Use it to control the satellite.
  if (counter == 0 ){
     api.getMyZRState(my_state);
     api.getOtherZRState(other_state);
  }
    counter++;
          }
```

Calculating the target coordinates

- The target is the midpoint between the two spheres.
- We can find the coordinates of the midpoint by taking the average of each coordinate as shown.
- For example, the x coordinate is

  $(x_1 + x_2) / 2$
- Using a **for loop** makes this calculation simpler.



Using for loops

- The for loop has 3 statements:

  The first is the starting index, the second is the condition for termination of the loop, and the last is how the loop will be incremented. **(index1 = x; index1 <= y; index1++){ }**

- The index variable is a variable to keep track of the for loop, and will need to be initiated within void loop ().
- To create a for loop for calculating the target array:
  - –Declare the variable **int index1** in void loop above the if statement.
    - Unlike the *global* variables declared above void init(), variables declared inside the loop are *local*; they will be "destroyed" when the loop ends and will not retain their values when the loop is called again a second later.
  - –After **api.getOtherZRState(other_state);** enter the for loop
    - Set the starting index to 0 **index1=0**
    - Set ending condition to be **index1 <= 2**
    - The index will be incremented by 1 after each iteration, so set the last argument of the for loop to **index1++**.
    - The target calculation will be added between the brackets.
- The code inside the loop will be executed until the termination condition is false.
  - In this case, it will execute 3 times as index1 goes from 0 to 2.

Using for loops (cont.)


```
void loop(){
  //This function is called once per second.
  //Use it to control the satellite.
  int index1;

  if (counter == 0 ){
     api.getMyZRState(my_state);
     api.getOtherZRState(other_state);
    for (index1 = 0; index1 <= 2; index1++){
    }
  }
    counter++;
          }
```

Syntax

- Instead of declaring the index variable outside of the for loop in a separate line, you can declare it and assign it an initial value inside the for loop like this:

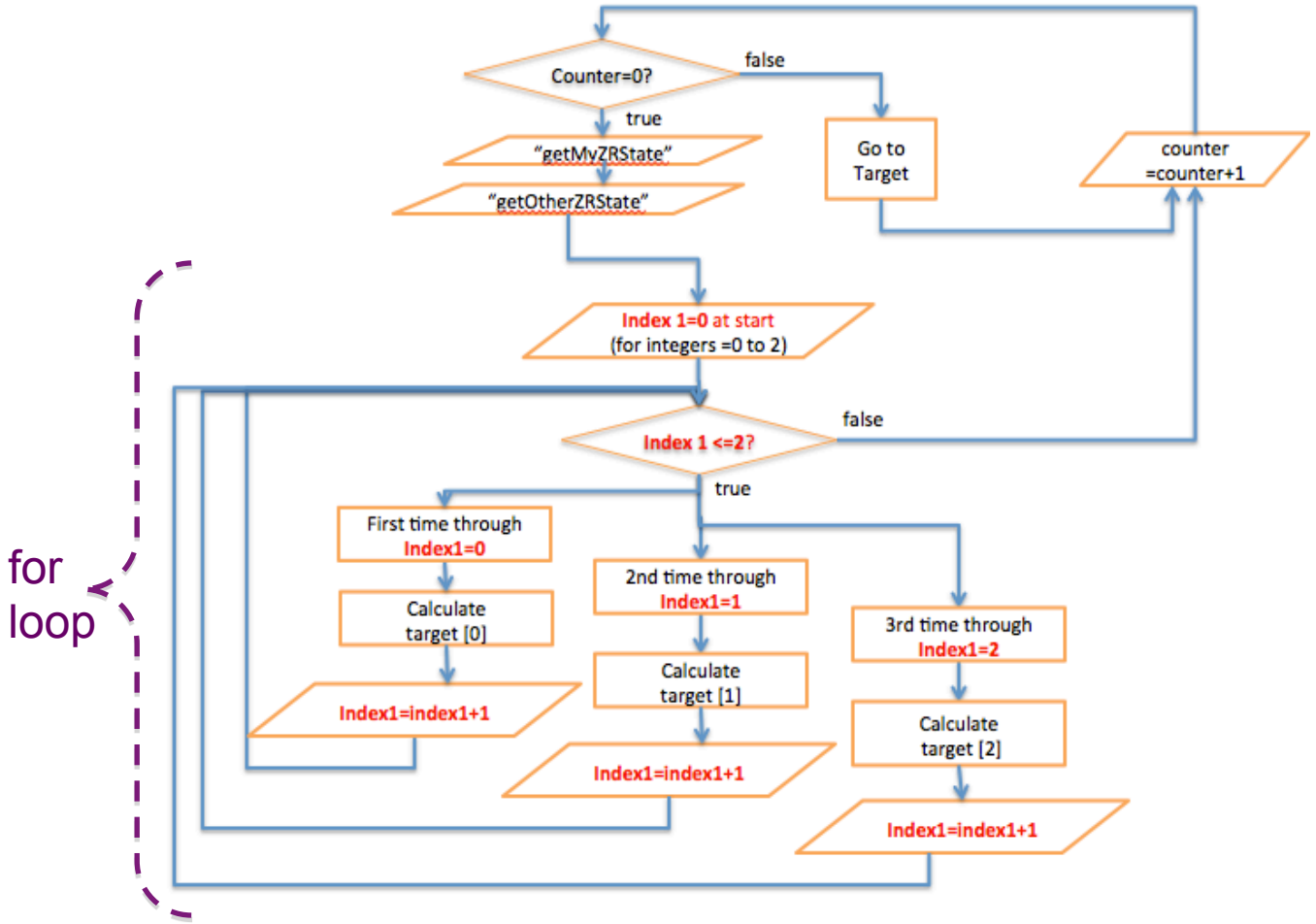
```
for (int index1 = 0; index1 <= 2; index1++){
```

- Either format is acceptable. As the tutorials become more advanced, we will switch to the format above because it is more compact. However, you are free to use either format.

ZERO ROBOTICS
ISS PROGRAMING CHALLENGE

For loop flowchart

- The **for loop** is a loop inside the main SPHERES loop as shown in the flowchart
- The variable **`index1`** is highlighted
- Do you see that the **for loop** in this example executes three times inside the main loop?



target[0] = x coordinate
target[1] = y coordinate
target[2] = z coordinate

ZERO ROBOTICS
ISS PROGRAMING CHALLENGE

Calculating target position

- The next step is to add the calculation for the target position between the brackets in the for loop.
- Write an equation to set the **target** array to the midpoint between the two satellites at their starting position, as shown.
- Recall that to find the midpoint, you must find the average of two points.
- Because **`index1`** goes from 0 to 2, the first time the loop will set **`target[0]`** (the x coordinate), then **`target[1]`** (y), then **`target[2]`** (z.)

```
void loop(){
  //This function is call one per second. Use it to control the sa
  //Use it to control the satellite.
  int index1;

  if (counter == 0 ){
     api.getMyZRState(my_state);
     api.getOtherZRState(other_state);
  }
    for (index1 = 0; index1 <= 2; index1++){
      (target[index1])=(my_state[index1] + other_state[index1])/2;
    }
    counter++;
}
```

Calculating target position (cont.)

- Do you see how this line of code sets each coordinate of *target* to the average of **`my_state`** and **`other_state`**?
- Finally, outside the if statement at the very end of the loop, add **`setPositionTarget(target)`**.

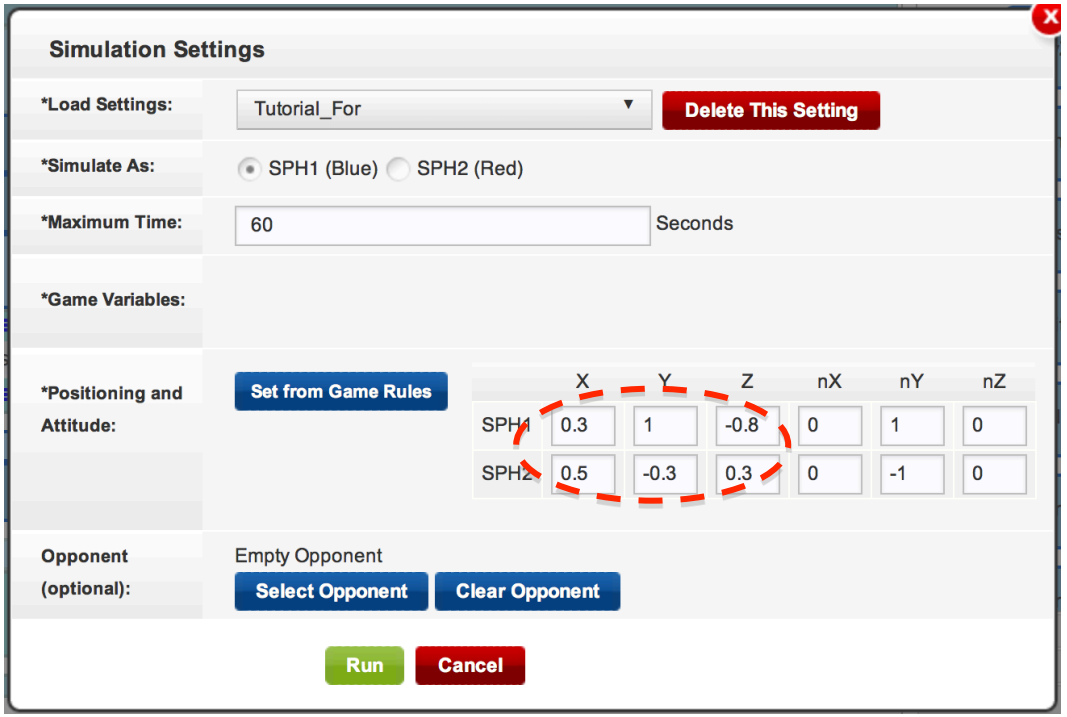
```
void loop(){
  //This function is call one per second. Use it to control the sa
  //Use it to control the satellite.
  int index1;

  if (counter == 0 ){
     api.getMyZRState(my_state);
     api.getOtherZRState(other_state);
  }
    for (index1 = 0; index1 <= 2; index1++){
      (target[index1])=(my_state[index1] + other_state[index1])/2;
    }
    counter++;
     api.setPositionTarget(target);
            }
```

WARNING!

- You must always be careful when using **for loops** to set arrays.
- For example, if you change the ending condition statement in the **for loop** from 2 to 3, the program will try to set **`target[3]`** to a value.
- But **`target[3]`** does not exist.
- This can cause serious problems. A large number of real-world computer crashes are caused by this type of mistake.
- *Make sure you are only putting values into array members that actually exist!*

Simulation

- Compile
- Simulate
- Create new setting “Tutorial_For”
  - Set Maximum Time to 60 seconds
  - Set the starting coordinates of SPH1 to x = 0.3, y = 1, z = -0.8
  - Set the starting coordinates of SPH2 to x = 0.5, y = -0.3, z = 0.3
- Run
- View simulation
- Change the starting coordinates to your own values and try it again.



Review

Congratulations!

- You have found the positions of the satellites in your code.
- You have used a **for loop** to carry out repeated calculations.
- You have programmed one satellite to move halfway toward the other one.