ZERO ROBOTICS

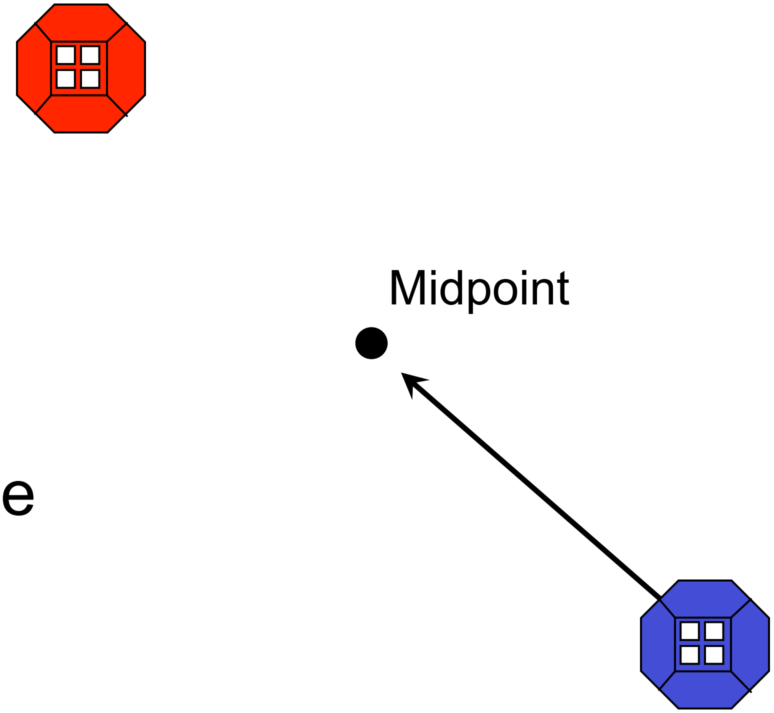
ISS PROGRAMING CHALLENGE

For Loops
(Project 8)

Goals

In this tutorial you will:

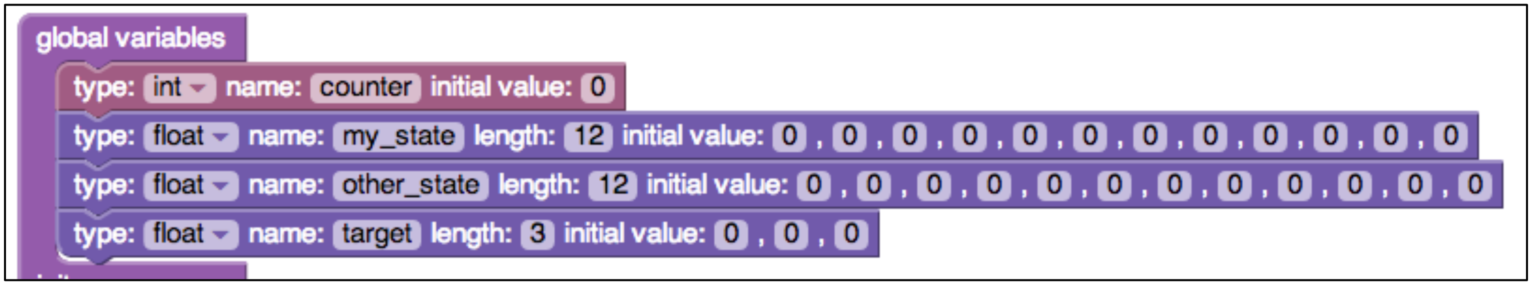
- Use a **for loop** to repeat an action a set number of times
- Find the position of the other satellite
- Program your satellite to move toward the other satellite, but stop halfway



Create a new project

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

leave initial values blank



global variables

type: int name: counter initial value: 0

type: float name: my_state length: 12 initial value: 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0

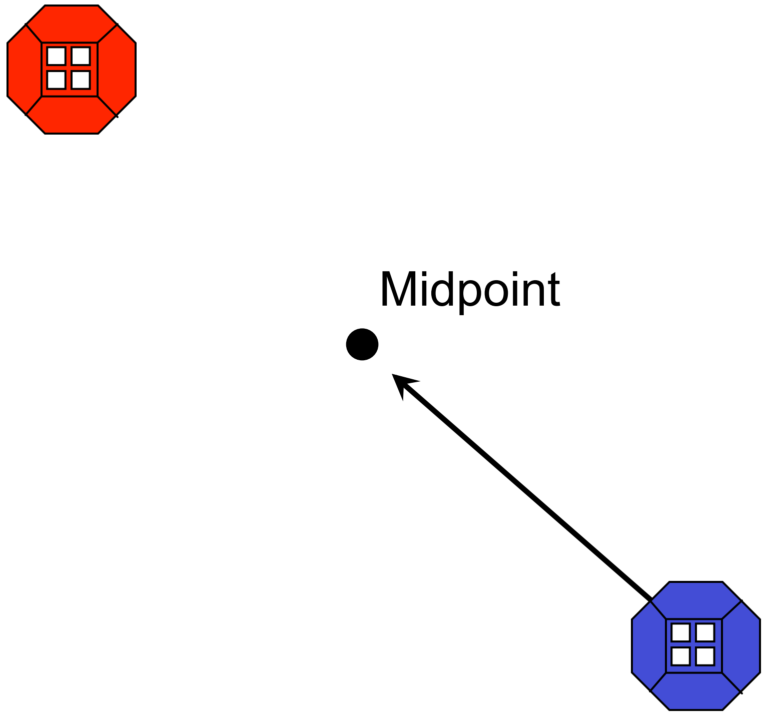
type: float name: other_state length: 12 initial value: 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0

type: float name: target length: 3 initial value: 0 , 0 , 0

Program outline

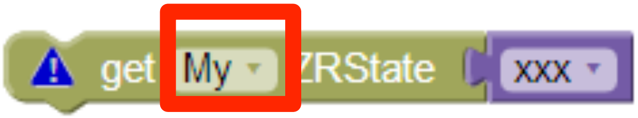
In this tutorial, you will move your blue satellite half the distance toward the red satellite.

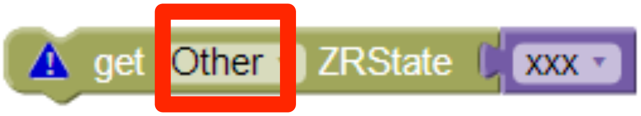
- First, you will use two API functions, **`getMyZRState`** and **`getOtherZRState`**, to find the starting positions of the two satellites.
- You will find the coordinates of the midpoint between the satellites.
- You will move to that position using **`setPositionTarget`**.



`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.)



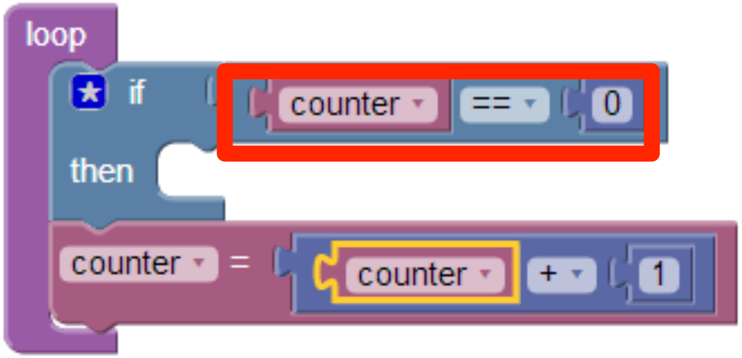


Array members:

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

Set up counter

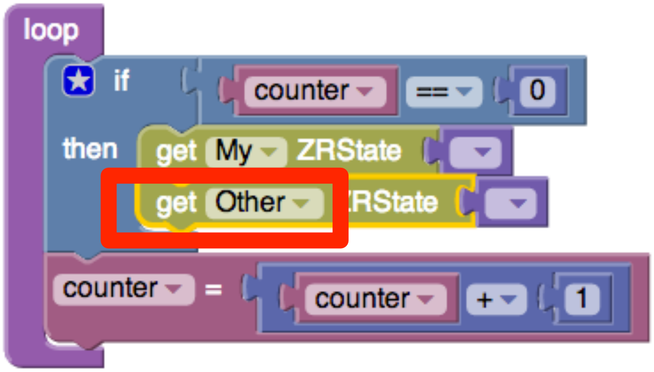
- Go to the “Logic” accordion
  - Drag an “if - then” block into the loop
  - Drag an “__==__“ block onto the “if” end of this block.
- Go to the Variables accordion
  - Drag a **`pink variable`** block (“--Select--”) into the first empty space
- Go to the math accordion
  - Drag a number block into the second empty space. (Set to 0)
- Any calculations put in this “if-then statement” will happen only once, at the start when *counter* is 0. This will be important to keep your target from changing as your position changes.
- Finally, add **`counter = counter + 1`** outside the “if-then” block as shown.



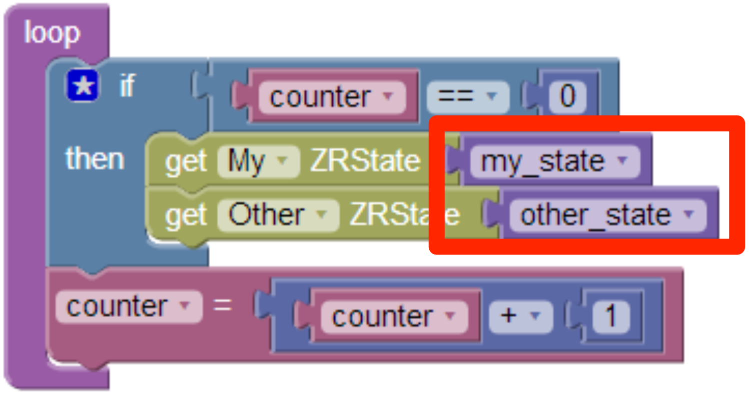
```
loop
  if counter == 0
  then
  counter = counter + 1
```

Set *my_state* and *other_state*

- Now you will find the positions of the two satellites so you can calculate your target.
- Go to the SPHERES Controls accordion and drag two **`getMyZRState`** blocks into the if-then block.
- Change the first drop-down menu on the second block to “ **`Other`**”
- Change the drop-down menus of **`getMyZRState`** to **`my_state`** and **`getOtherZRState`** to **`other_state`**.
- The arrays **`my_state`** and **`other_state`** have now been set to the states of the two satellites.



```
loop
  if counter == 0
  then get My ZRState
       get Other RState
  counter = counter + 1
```

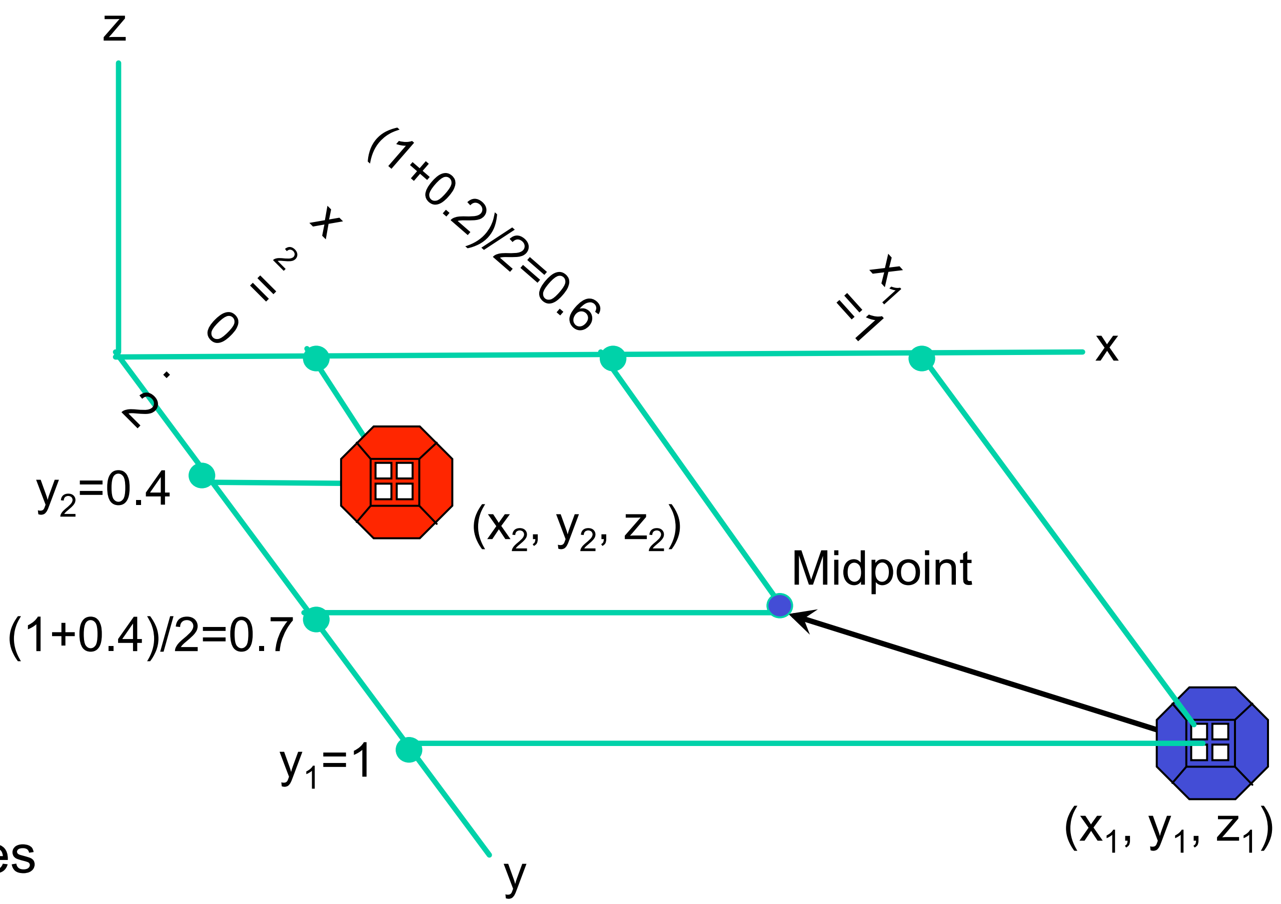


```
loop
  if counter == 0
  then get My ZRState my_state
       get Other ZRSta e other_state
  counter = counter + 1
```

Calculating the target coordinates

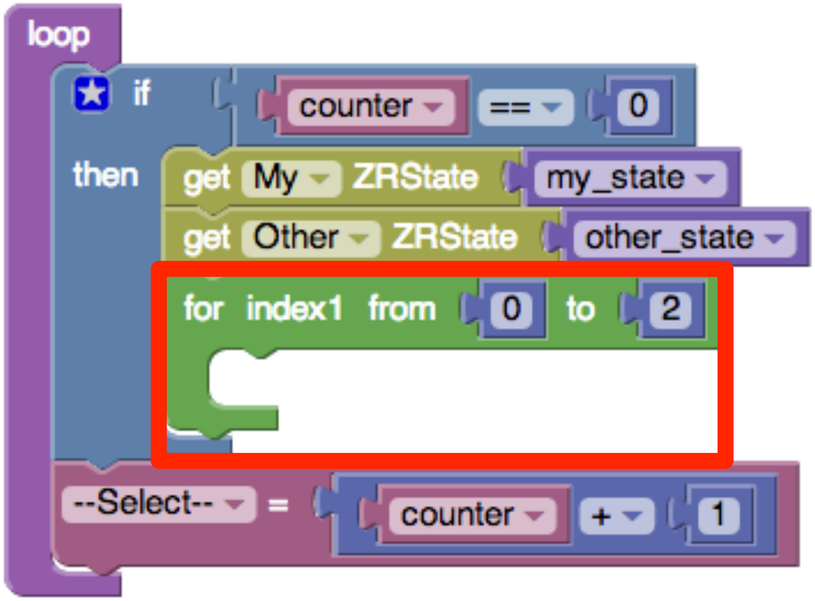
- The target is the midpoint between the two spheres.
- We can find the coordinates of the midpoint by taking the average of each coordinate as shown.
- example, the x coordinate is

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



Using **for loops**

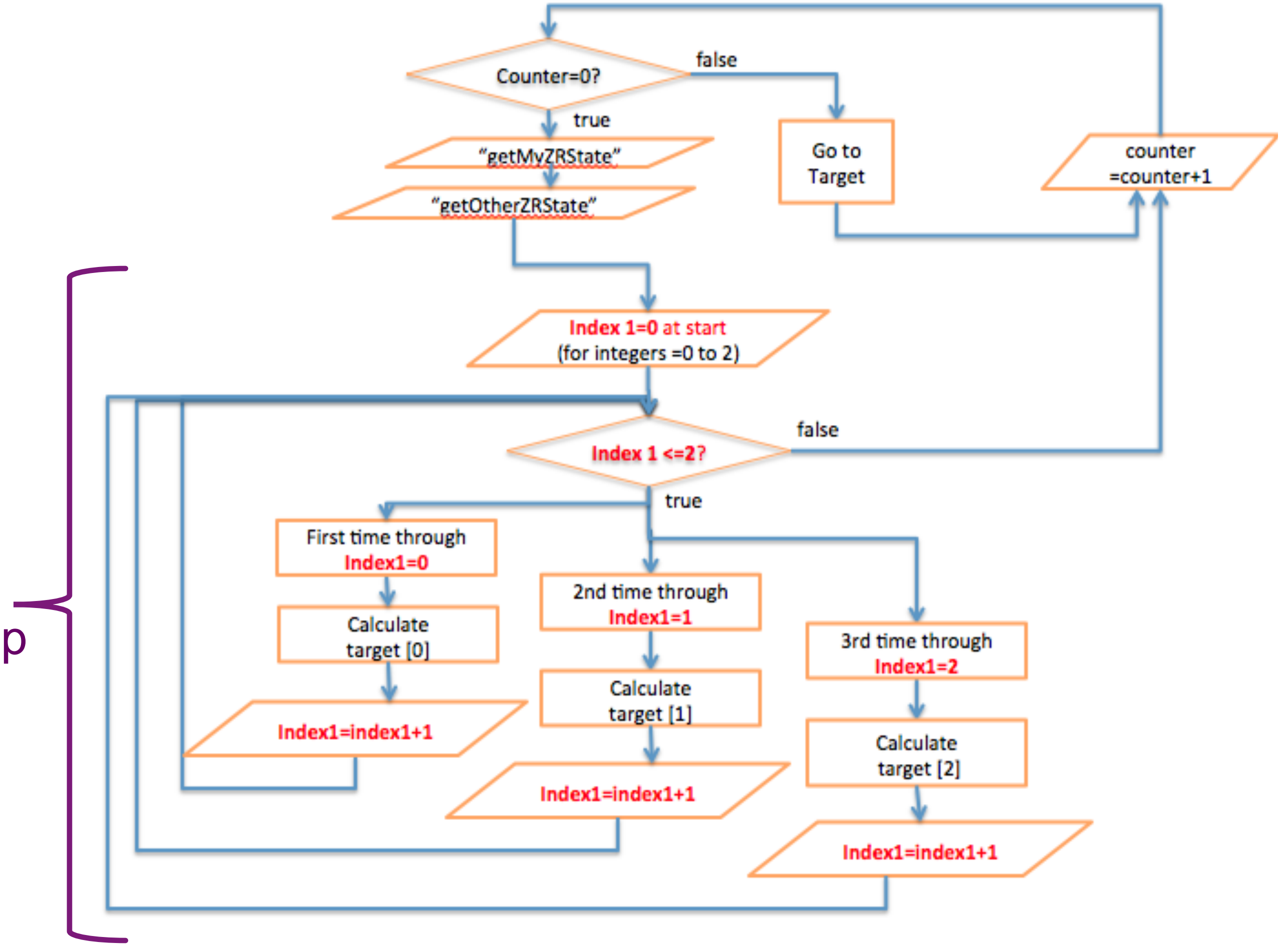
- Go to the Loops accordion and drag a "`for index1 from 0 to 9`" block <u>inside</u> the if-then block below **`getOtherZRState`**.
- Change the number blocks to **"0 to 2"** as shown.
- Everything inside the "for loop" block will be executed three times.
- The statement automatically creates a new **`int`** variable called **`index1`** that increases like a counter each time (shown in the following slides).



```
loop
  if counter == 0
  then get My ZRState my_state
       get Other ZRState other_state
       for index1 from 0 to 2
  --Select-- = counter + 1
```

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?



for loop

Counter=0?

false

true

"getMyZRState"

"getOtherZRState"

Go to Target

counter =counter+1

Index 1=0 at start (for integers =0 to 2)

Index 1 <=2?

false

true

First time through Index1=0

Calculate target [0]

Index1=index1+1

2nd time through Index1=1

Calculate target [1]

Index1=index1+1

3rd time through Index1=2

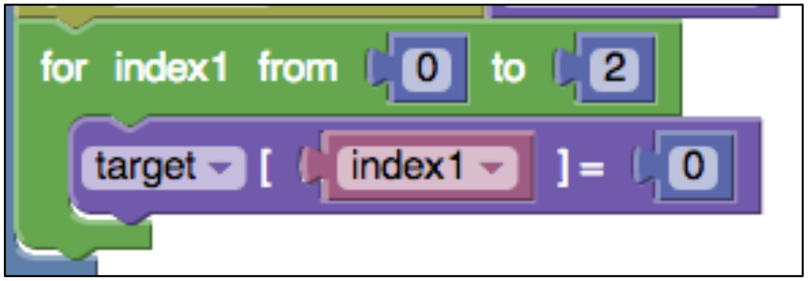
Calculate target [2]

Index1=index1+1

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

Calculating target position

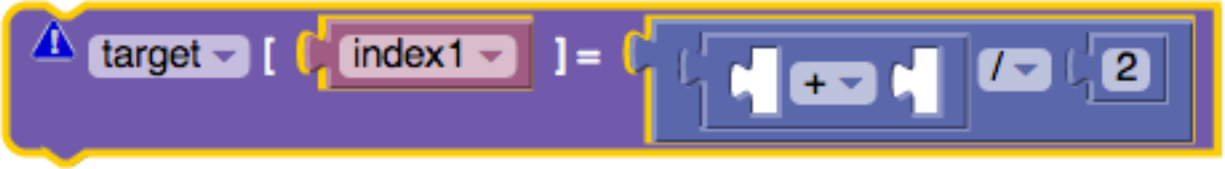
- Go to the Variables accordion
  - Drag a purple array **“`Select [0]=0`”** block into the **for loop**.
  - Change the drop-down menu to **`target`**.
- Drag a pink variable (“--Select--”) block into the first empty space and change its drop-down menu to **`index1`**.
- Because **`index1`** goes from 0 to 2, the first time the loop will set **`target[0]`** (the x coordinate), then **`target[1]`** (y), then **`target[2]`** (z.)



```
for index1 from 0 to 2
    target [ index1 ]= 0
```

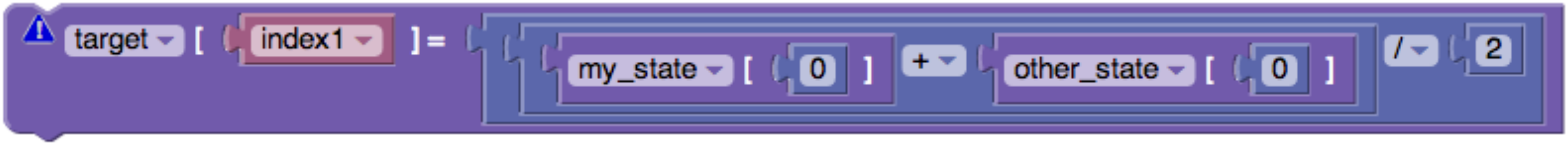
Calculating target position (cont.)

- Go to the Math accordion and drag a "**__/__**" block onto the 0 in the block you just added. (toggled from the "__+__" block)
- Drag a "__+__" block into the first empty space in the block (the numerator.)
- Drag a number block into the second empty space set to 2.



target [index1]= ((+) / 2)

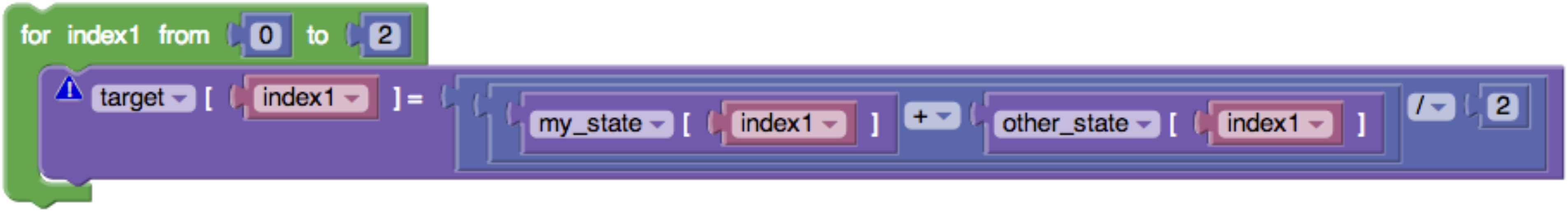
- Drag a `--Select--[0]` block from the Variables accordion onto each side of the "**__+__**" block.
- Change to: `my_ state [0]+ other_state [0]`



target [index1]= ((my_state [0] + other_state [0]) / 2)

Calculating target position (cont.)

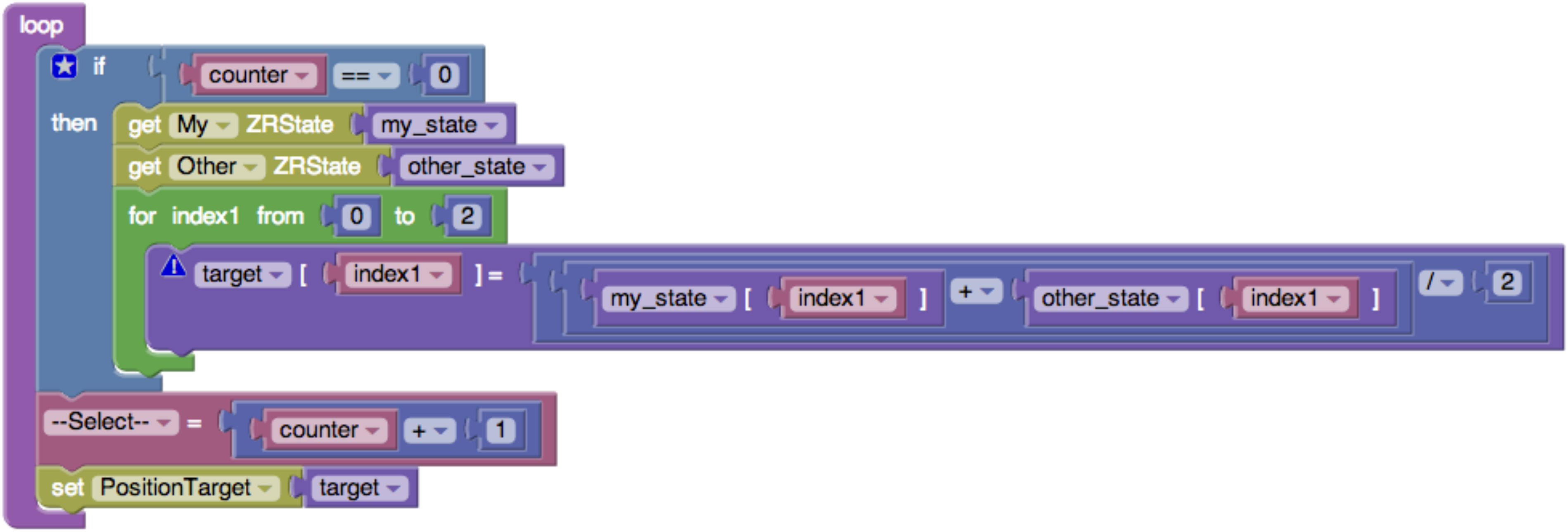
- Now drag two **pink variable** (“—Select--”) blocks onto the 0 in the index of the **my_state[0]** and **other_state[0]** blocks and change them to **index1**.



```
for index1 from 0 to 2
    ⚠ target [ index1 ] = my_state [ index1 ] + other_state [ index1 ] / 2
```

- 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)** (shown on next slide).

Your final program

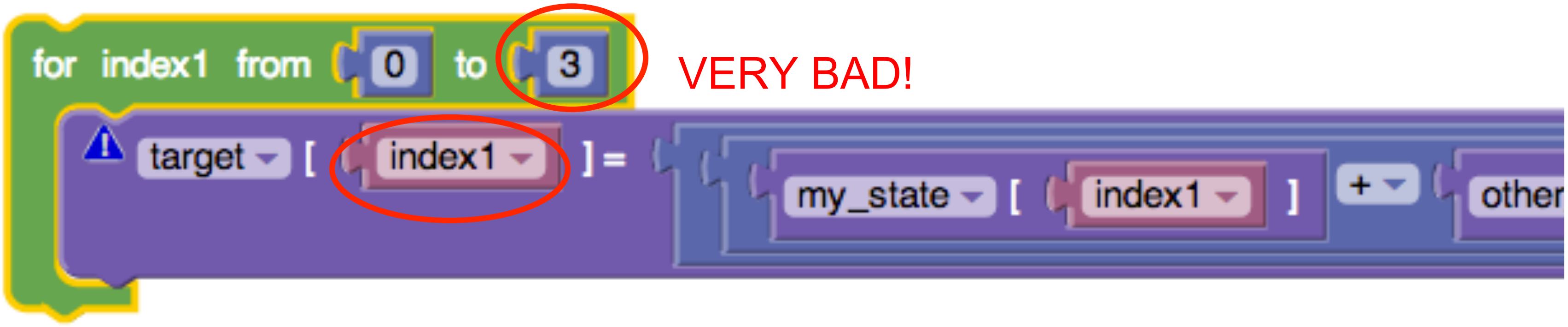


```
loop
  if counter == 0
  then
    get My ZRState my_state
    get Other ZRState other_state
    for index1 from 0 to 2
      target [ index1 ] = my_state [ index1 ] + other_state [ index1 ] / 2
  --Select-- = counter + 1
  set PositionTarget target
```

- Before you simulate: See instructions on the next 2 pages including!
  - Warning
  - Changing the starting coordinates in the simulation settings window

WARNING!

- You must always be careful when using **for loops** to set arrays.
- For example, if you change the 2 in the **for loop** block to a 3, the program will try to set **`target[3]`** to a value.
- But **`target[3]`** does not exist. *(target [0], target [1],target [2])*
- This can cause serious problems.
- Make sure you are only putting values into array members that actually exist!



```
for index1 from 0 to 3
    target [ index1 ] = my_state [ index1 ] + other
```

VERY BAD!

Simulation

- Compile
- Simulate
  - Set Maximum Time to 60 seconds
  - Set the starting coordinates of

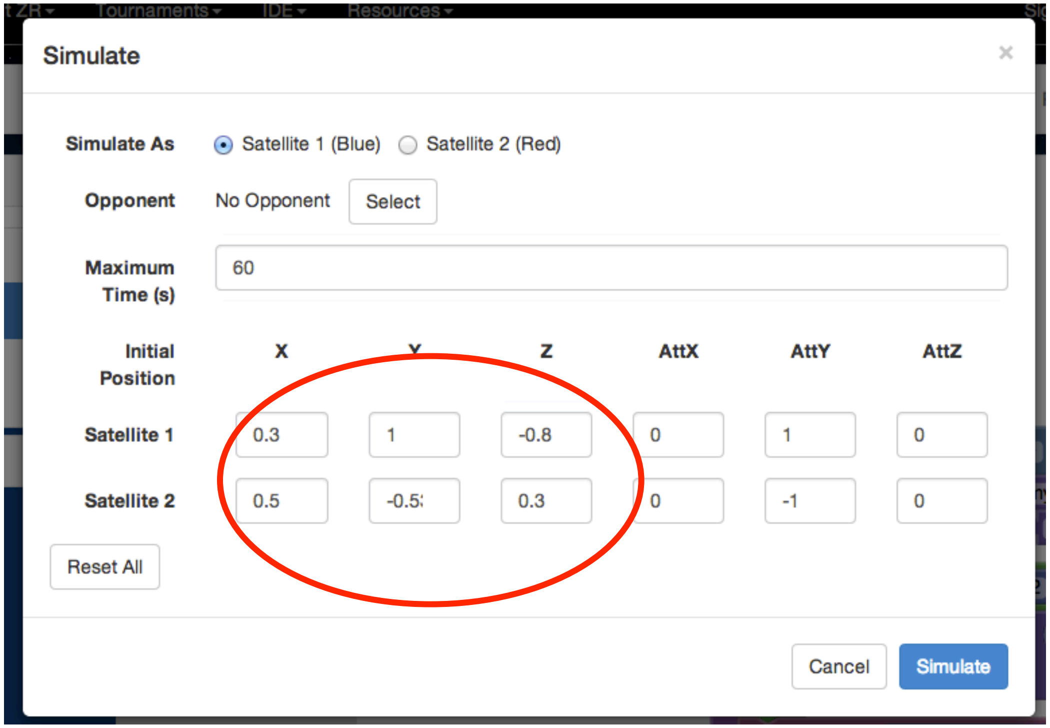
Satellite1:

- **x = 0.3, y = 1, z = -0.8**

- Set the starting coordinates of

Satellite 2:

x = 0.5, y = -0.3, z = 0.3



Simulate

Simulate As ● Satellite 1 (Blue) ○ Satellite 2 (Red)

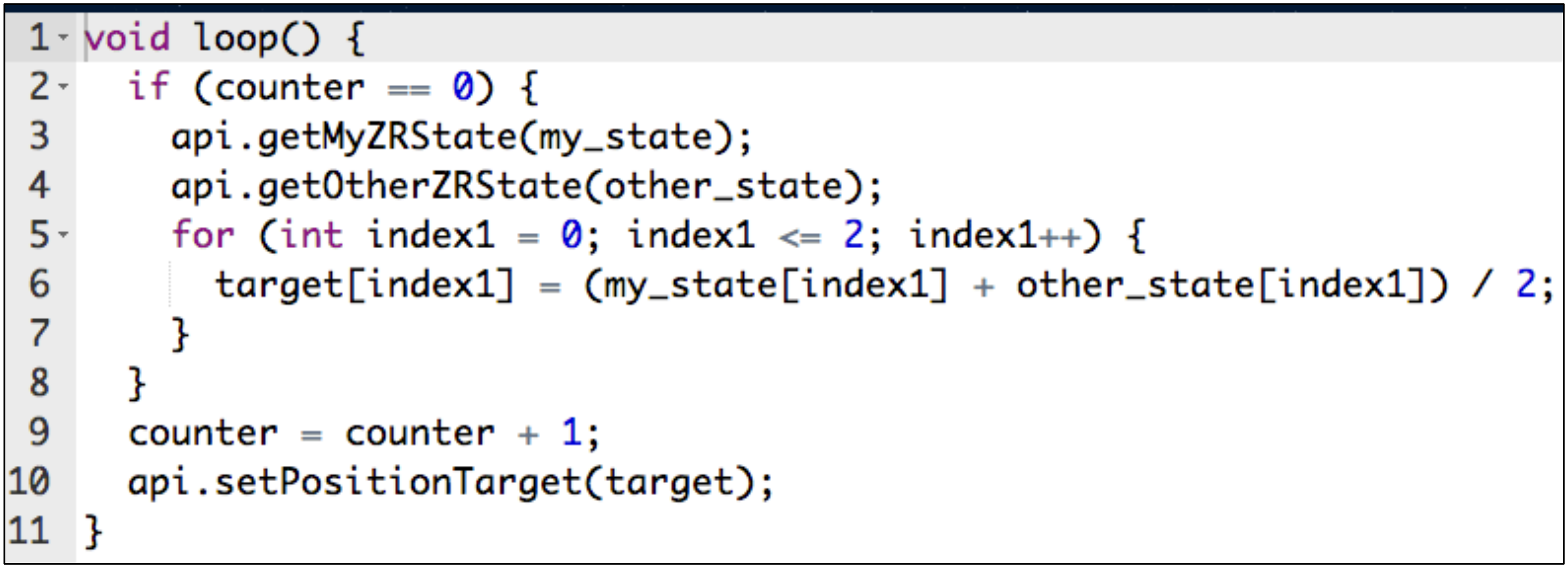
Opponent No Opponent

Maximum Time (s) 60

Initial Position	X	Y	Z	AttX	AttY	AttZ
Satellite 1	0.3	1	-0.8	0	1	0
Satellite 2	0.5	-0.5	0.3	0	-1	0

- View simulation
- Change the starting coordinates to your own values and try it again.

Your program in C code



```
void loop() {
  if (counter == 0) {
    api.getMyZRState(my_state);
    api.getOtherZRState(other_state);
    for (int index1 = 0; index1 <= 2; index1++) {
      target[index1] = (my_state[index1] + other_state[index1]) / 2;
    }
  }
  counter = counter + 1;
  api.setPositionTarget(target);
}
```

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.