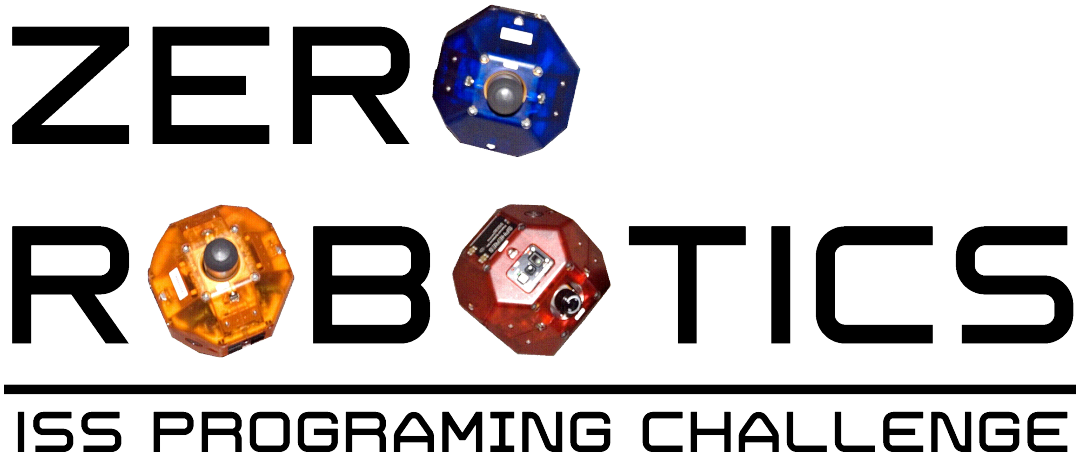
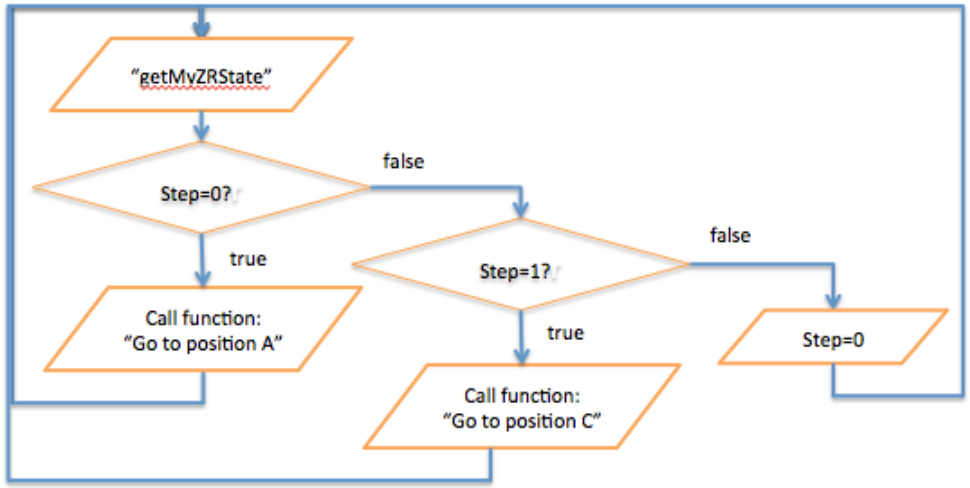


Functions and the Step Counter Model

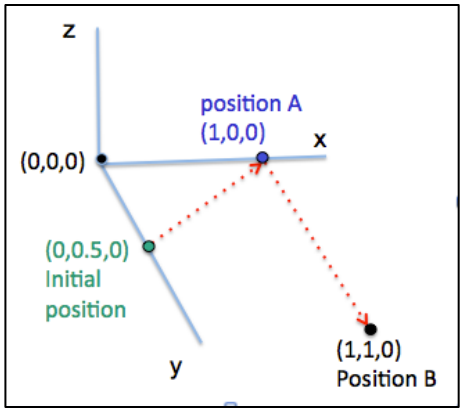

Goals

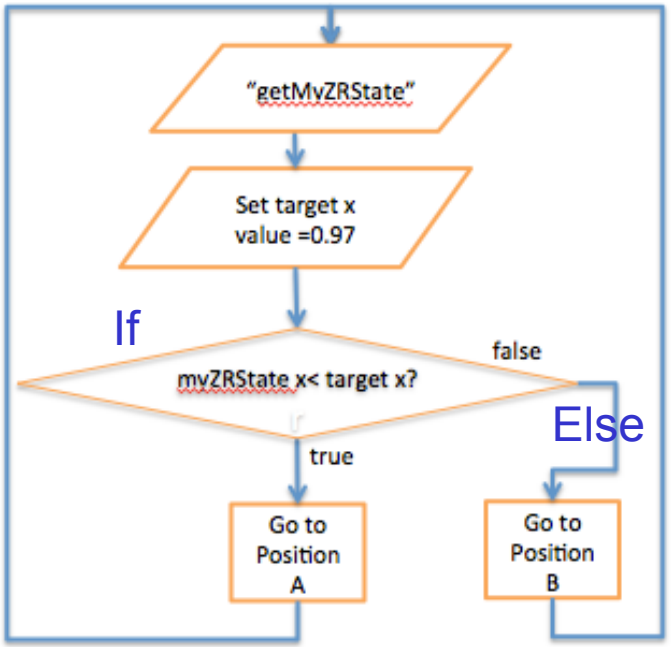
- In this tutorial you will:
  - Learn to use a step counter in your program
  - Practice creating functions
- Important note: There will be several slides at the beginning of this tutorial to read and understand before you begin to create your next program.



Review of previous tutorial

- The sketch and flow diagram on the right describe the program you wrote in a previous tutorial (*Applied Conditionals*)
- What would happen if the location of Position B were changed? Would this program always move the SPHERES satellite to Position B?
- On the next slide you will be given an example of a location where the program would not work and the reason why.





Example that doesn't work

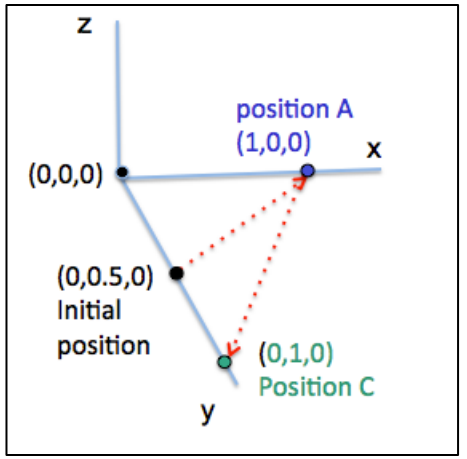
- Look at the new position called Position C.
  - Q: How does the *x*-coordinate of the satellite change as it moves from position A (1,0,0) to position C (0,1,0) in the picture?
  - A: The satellite starts with *x*-coordinate = 1 and moves towards *x* coordinate = 0
- The program sets target x = 0.97 then states:

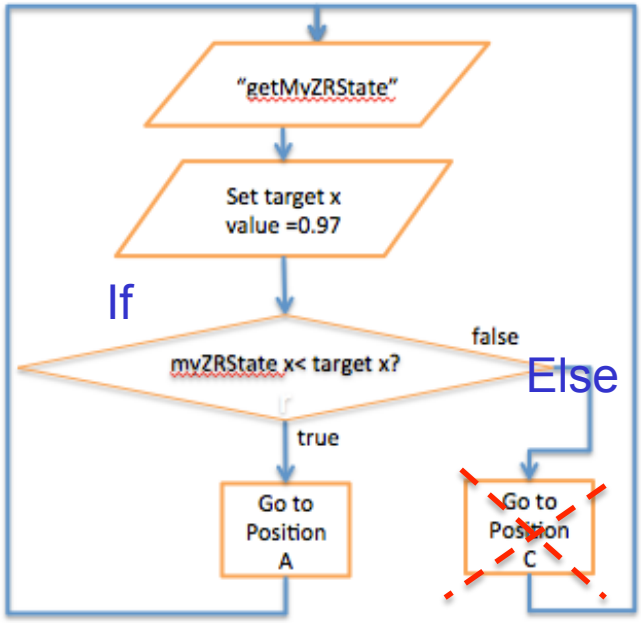
  If myZRState[0] < target [0],

  (which means If SPHERES *x*-coordinate < 0.97)

  Then go to position A

  Else go to position C
- As the satellite moves from position A toward position C, its *x*-coordinate becomes < 0.97 and it will be sent back to position A.
  - The program will not allow the satellite to move to position C (which has *x*-coordinate < 0.97)



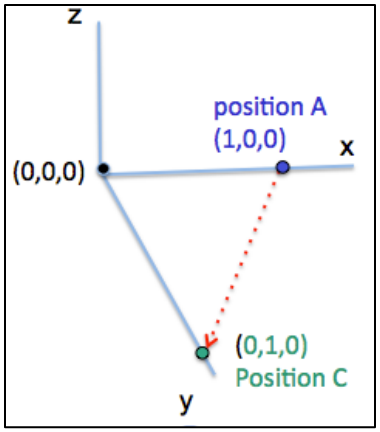


Picking a target value for Position C

- So what is the target value that you would choose to move the satellite from position A to position C?
  - Since the satellite starts with *x*-coordinate=1 and moves towards *x*-coordinate=0:
  - Pick a target close to zero.
  - Pick target[0]=0.03 to include margin for error
- In this case the satellite's *x*-coordinate is **greater than** 0.03 until it reaches the target
- So the conditional statement for this example would be:

"If myZRState[0] **>** target[0]

Then continue to position C"



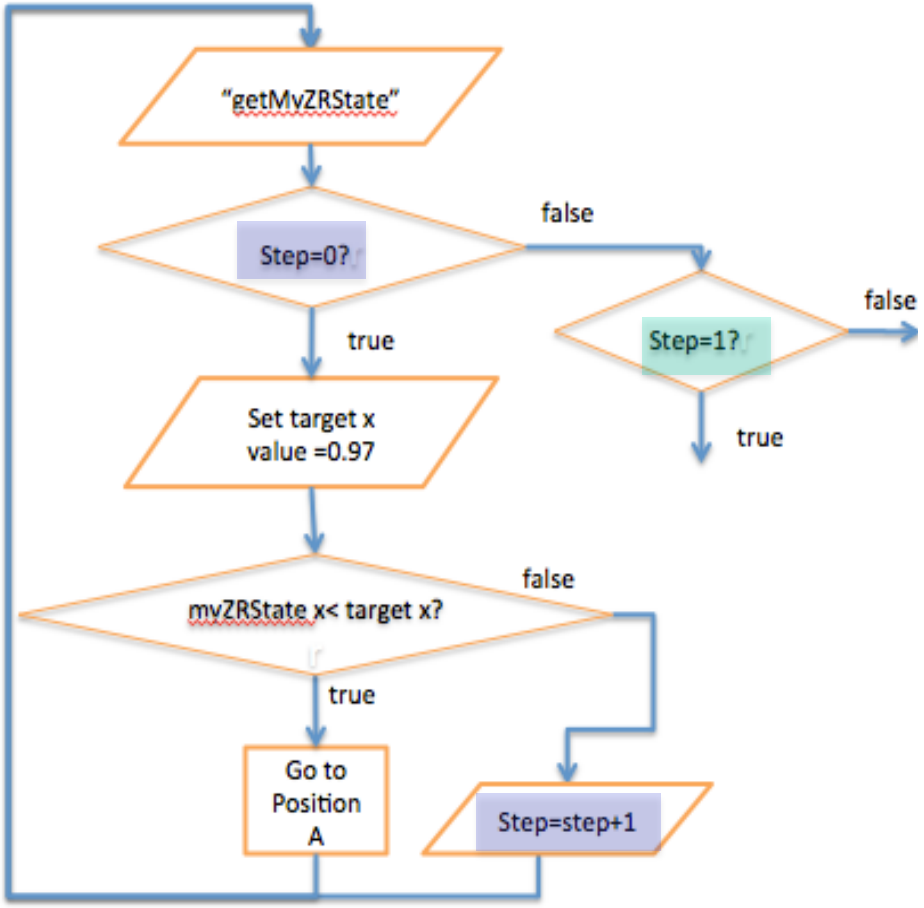
Solution to problem

- We need to write a program that allows us to pick different target values depending on where we want the satellite to move

- We can solve this problem using a step counter (as described on the following slides)

Using a step counter

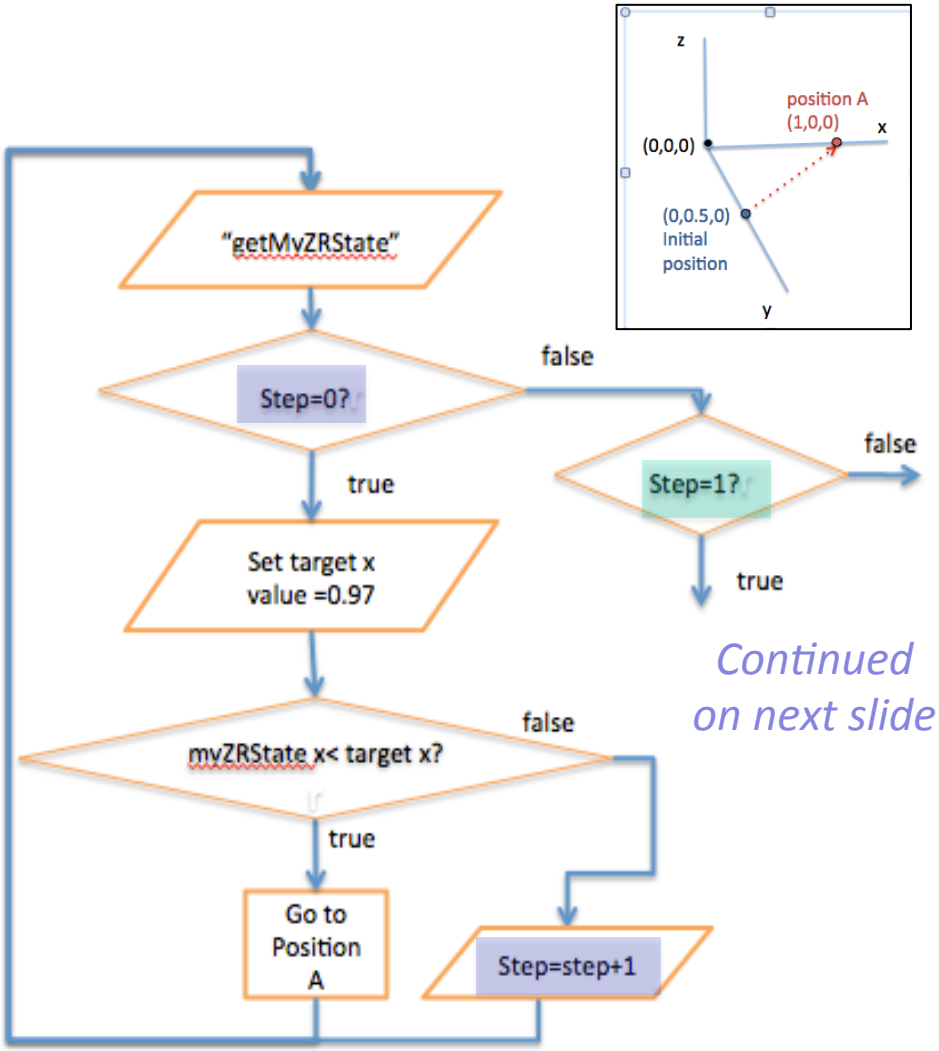
- A step counter is one way you can organize a program. It is particularly useful in a game like ZR.
- To use a step counter:
  - Break up the program into steps (for example, moving to a point is one step)
  - Use a variable to keep track of how many steps have been performed
  - Use conditionals to make sure you execute only the next step in the process
  - This process ensures that all steps happen in the right order
- An example of this process is provided on the next several slides



Using a step counter (cont.)

Look more closely at the example: Let's say the first step in the program (step 0) is for the satellite to go to position A

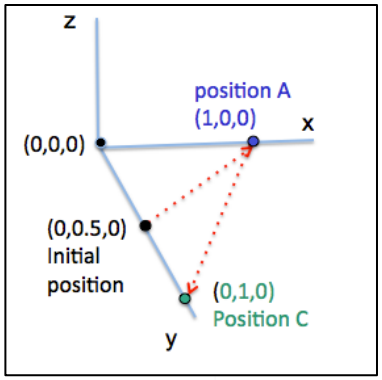
- The flow diagram to the right includes a step counter. Note that:
  - A conditional statement is added to the program to check if the program is in the first step (Step=0?)
  - The step is increased by 1 (Step=step+1) after the satellite completes the first step by reaching position A
  - Another conditional statement is added to check whether or not the program has moved to the next step (Step=1?).

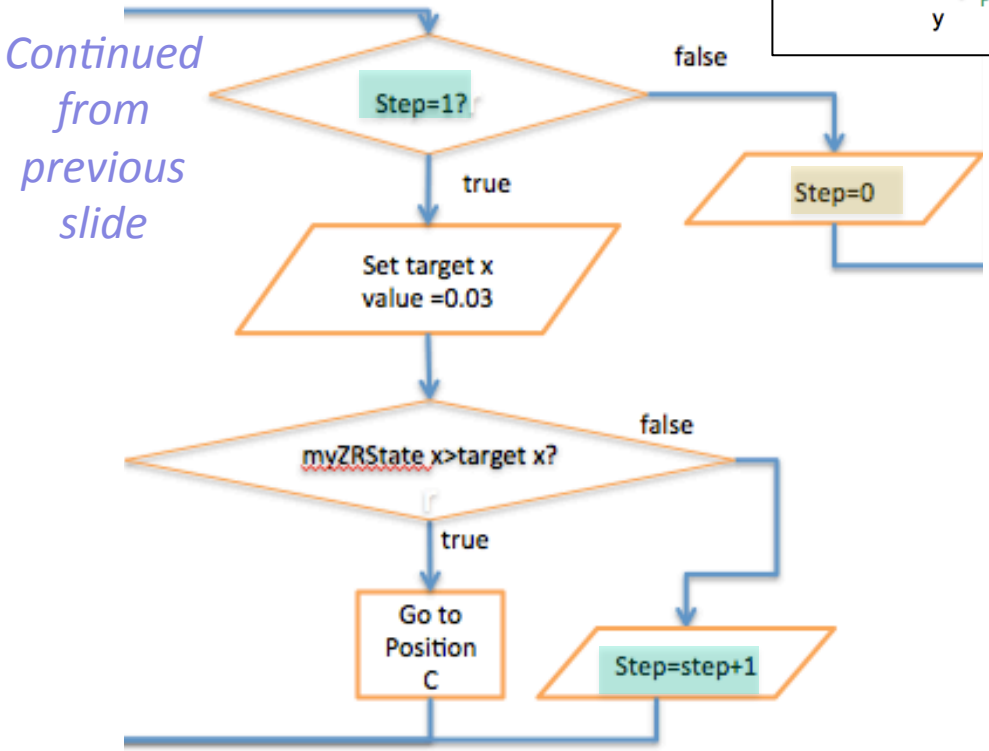


Continued on next slide

Using a step counter (cont.)

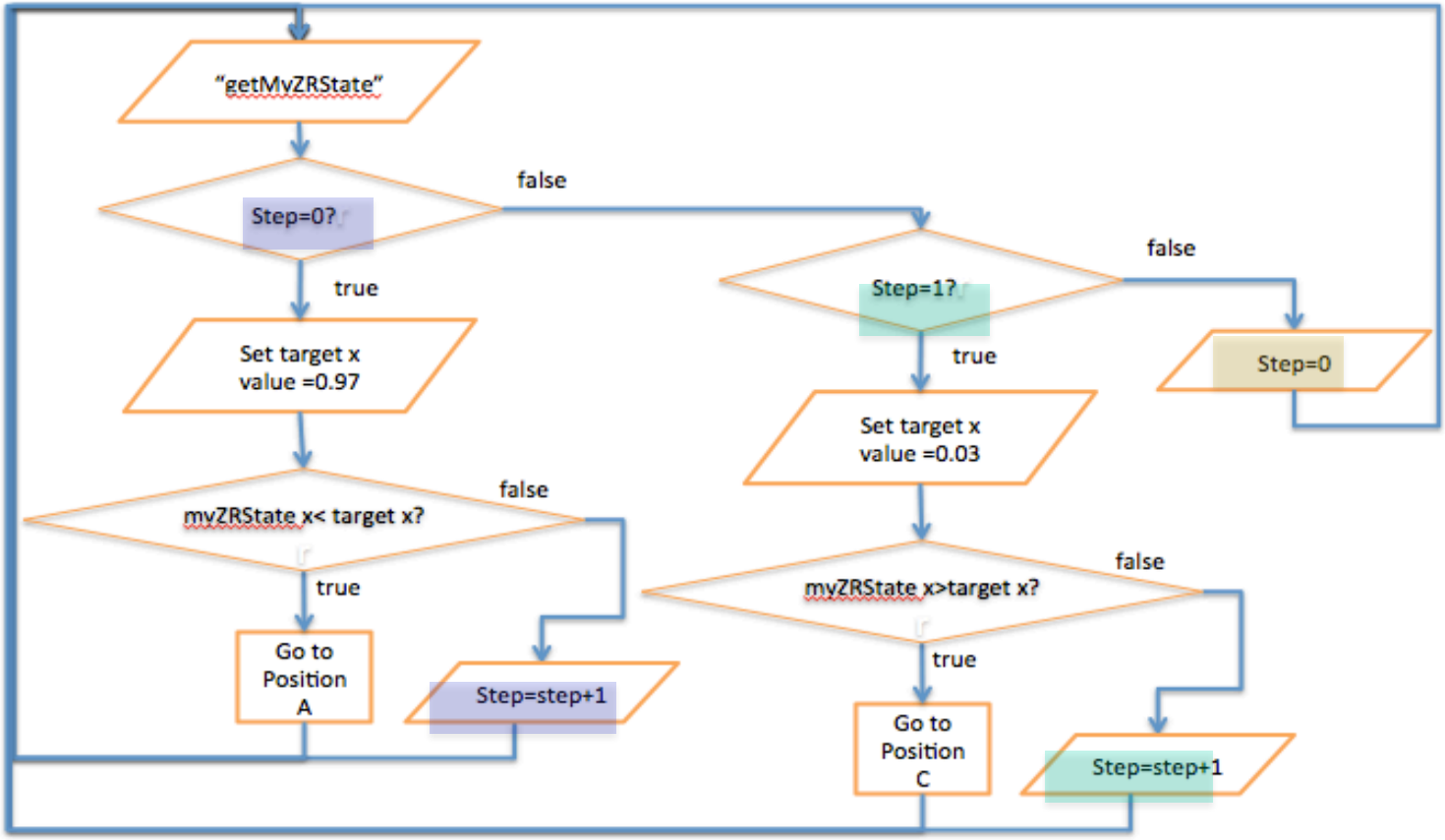
- Next we want the satellite to leave position A and go to position C (as shown in the sketch)
- By using a step counter we don't have the same problem that we had before because the variable target x can be given a new value inside each step.
- For Step=0: target x was set to a value appropriate for position A (see previous slide)
- For Step=1? target x is set to a new value appropriate for position C (shown here)
- Next, Step=0 resets the step counter





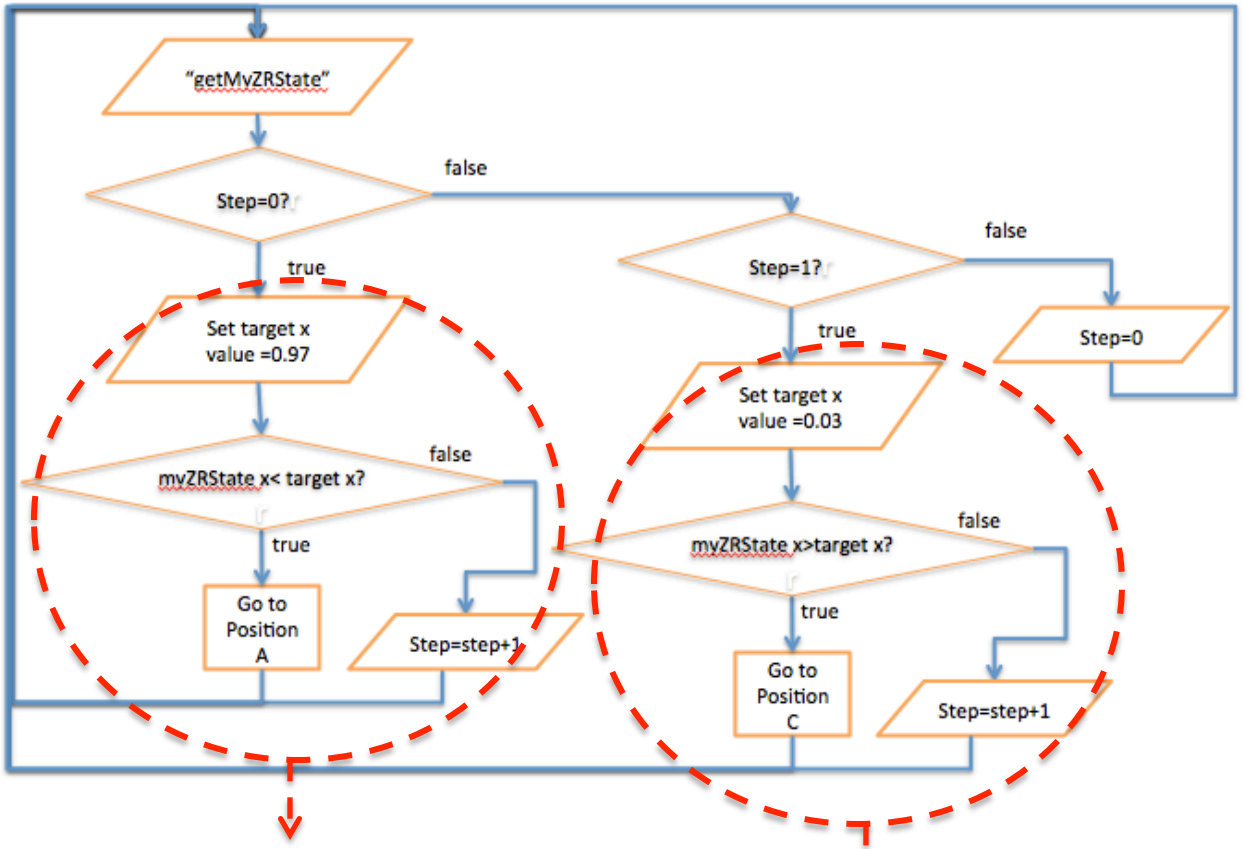
Using a step counter (cont.)

- This slide shows the complete flow chart



Functions

- We can make this program simpler to read by breaking it down into smaller pieces.
- This is done by creating procedural functions
- For example, we can create two functions
  - One that includes the parts of the program that sends the satellite to position A
  - One that includes the parts of the program that send the satellite to position C

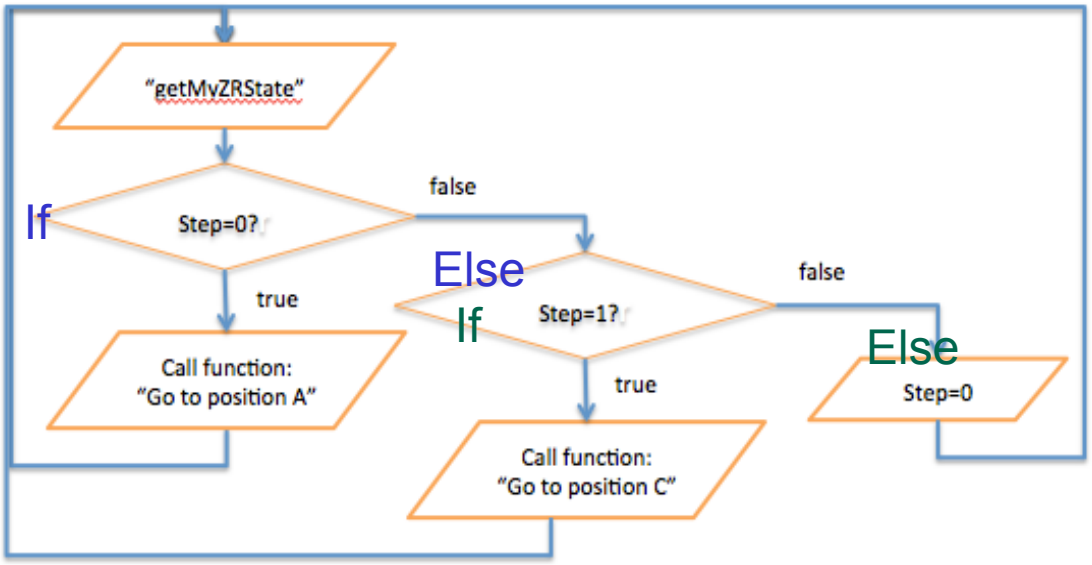


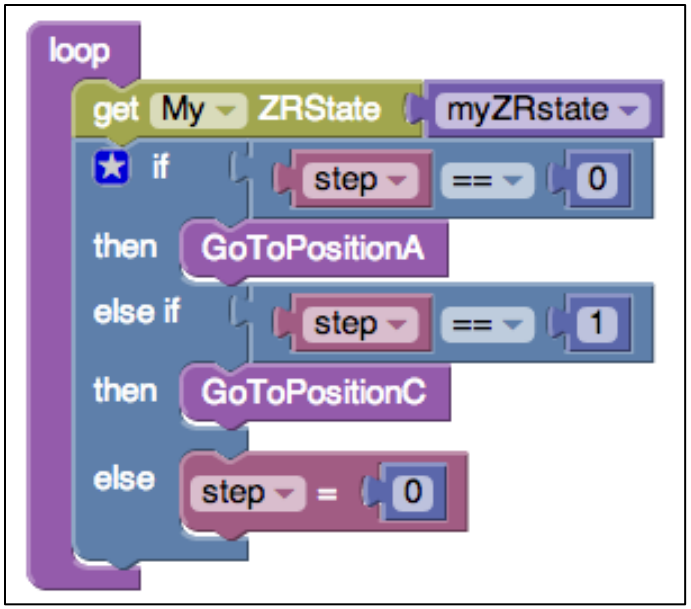
Create function called "Go to position A"

Create function called "Go to position C"

Functions (cont.)

- This is what the example program's flow diagram and graphical editor program would look like if written using two functions: one called "Go to position A" and one called "Go to position C"

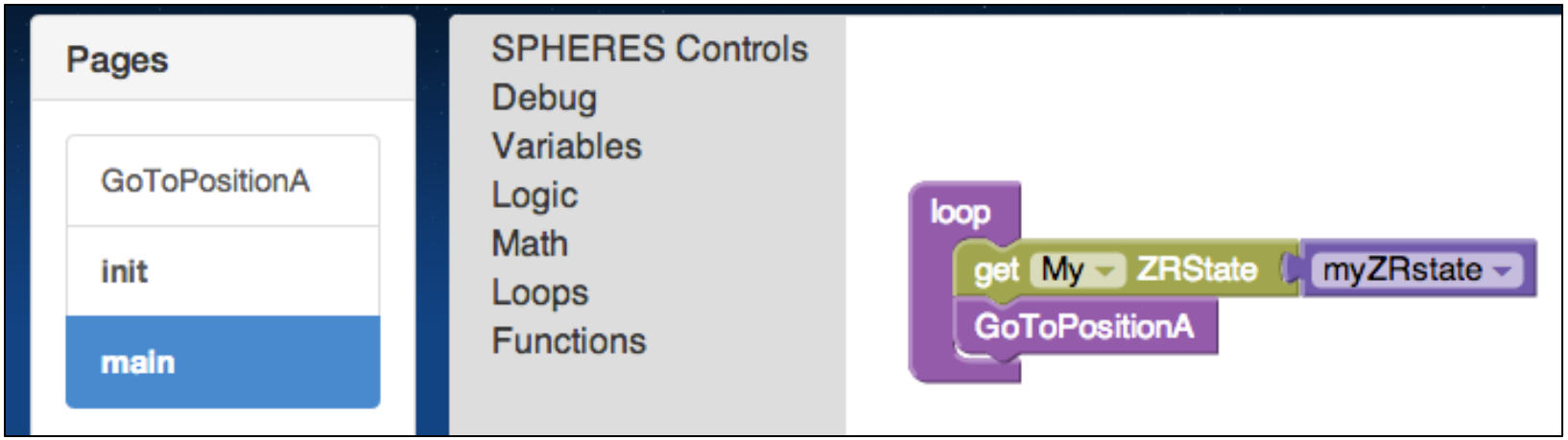




- You don't see Step=step+1 here because it is included inside each function.

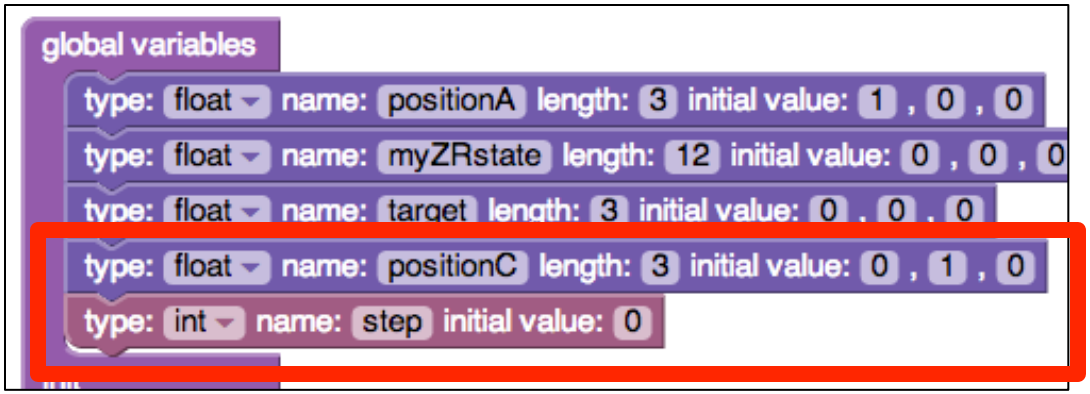
Create a New Project Using "Save As"

- To create this program with a step counter and functions, you will start from the program you created in the previous tutorial
  - Open the ZR IDE
  - Open Project 10
  - On the menu bar select "File" and then "Save As" from the drop down menu.
  - Type in **Project 11** and select **Graphical Editor**, **Free Mode**



Declare Variables and Arrays

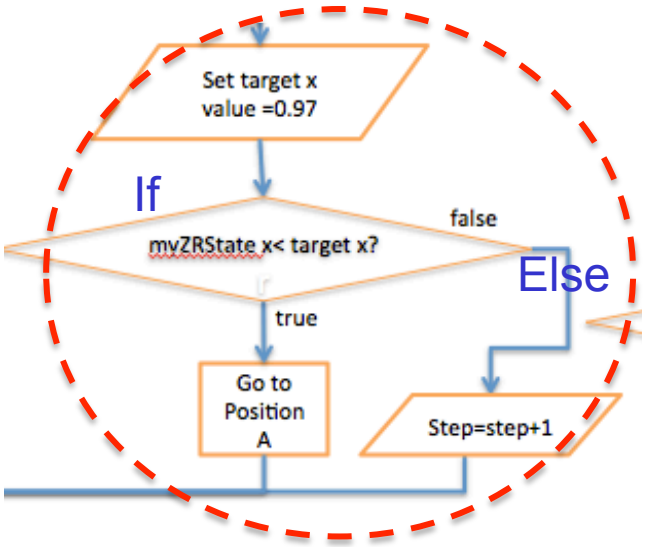
- You will need to create two new variables
  - `float positionC [3]`
    - Create array
    - Set initial value to 0,1,0
  - `Int step`
    - Create variable
      - Leave initial value blank

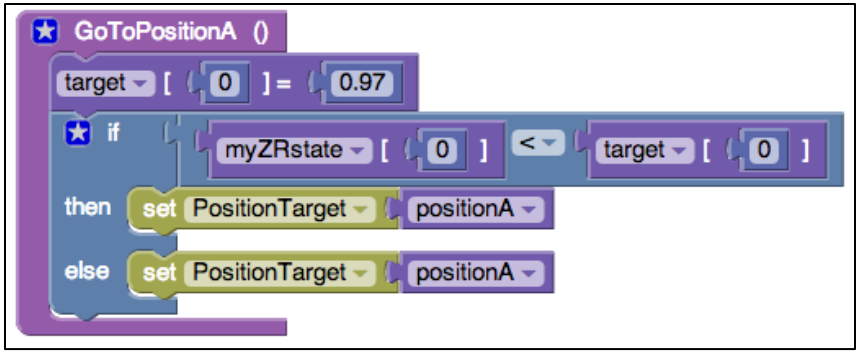


- Verify the remaining variables as follows:
  - `float positionA[3]`
    - Verify initial value is set to 1,0,0
  - `float myZRstate[12]`
    - Leave initial value blank
  - `float target[3]`
    - Leave initial value blank
- Check that your program has the variables shown in the box to the right

Create "GoToPositionA" Function

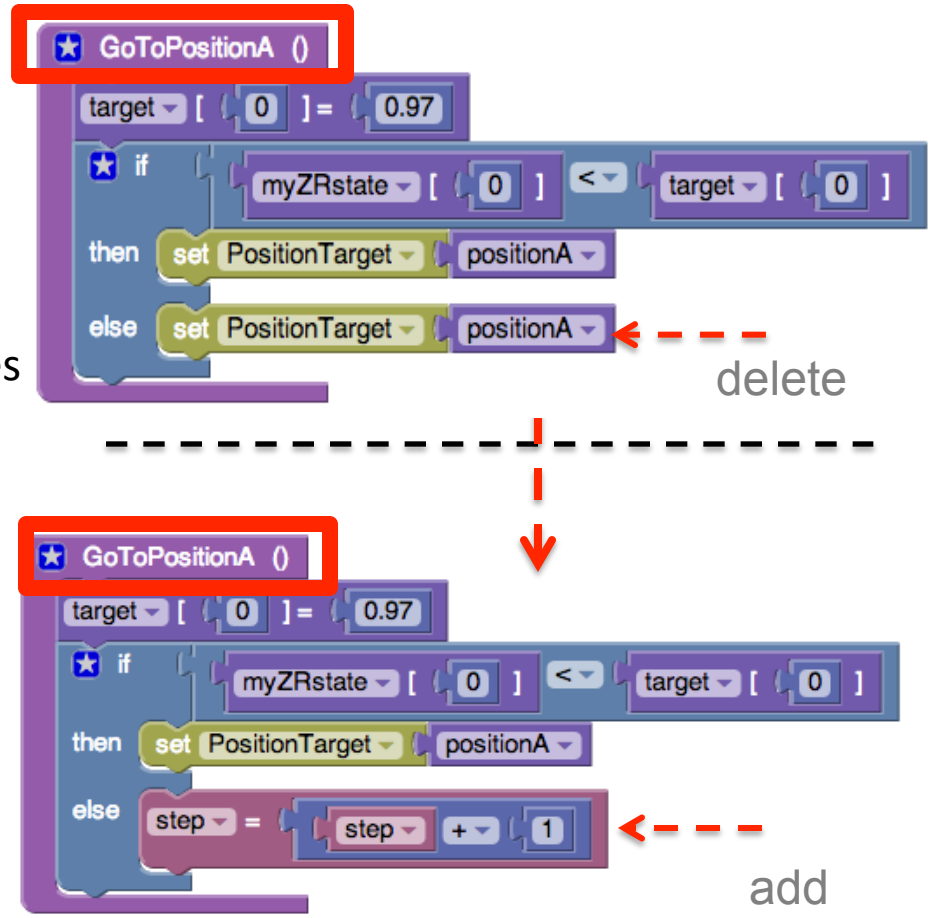
- Your program already has a function to send the SPHERES to position A that is very similar to the one shown in the flow diagram
- Click on the page "`GoToPositionA`"
- Can you see the difference between the part of the flow diagram that we want to make into a function called "`GoToPositionA`" and your program?
- (Hint:
  - Compare your program's **`else`** statement with the flow diagram
    - your old program continues to set the SPHERES position to positionA
    - the new program increments the step counter (step=step+1)





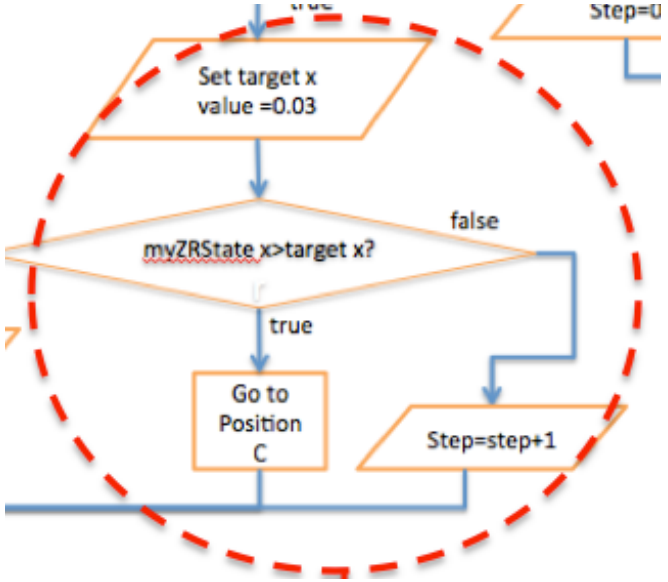
Create "GoToPositionA" Function (cont.)

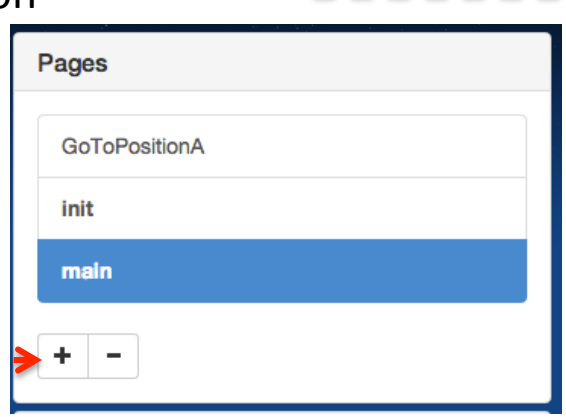
- To modify your program: Open your page "GoToPositionA" and delete the setPositionTarget block from the else statement (drag to trash)
- Add a step counter into the "else" slot
  - Drag a "step = 0" block from the variables accordion into the "else" slot
  - Drag a "__+__" block from the Math accordion into 0 block
  - Drag a "—Select—" block from the variables accordion into the first blank and toggle to "step"
  - Drag a number block (1) from the math accordion into the second blank
- Your function "GoToPositionA" is complete!

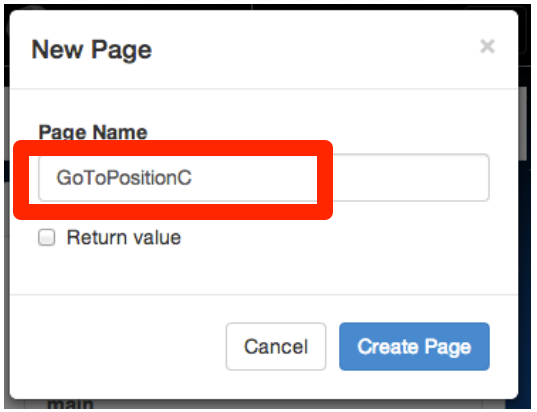


Create "GoToPositionC" Function

- Now let's create the second function in the flow diagram "GoToPositionC"
- First create a new page called "GoToPositionC"
- Click on the "+" button on the "pages" window
- For Page Name type: GoToPositionC. This will be the name of your function
  - Leave Return Value blank
  - Click the "Create Page" button

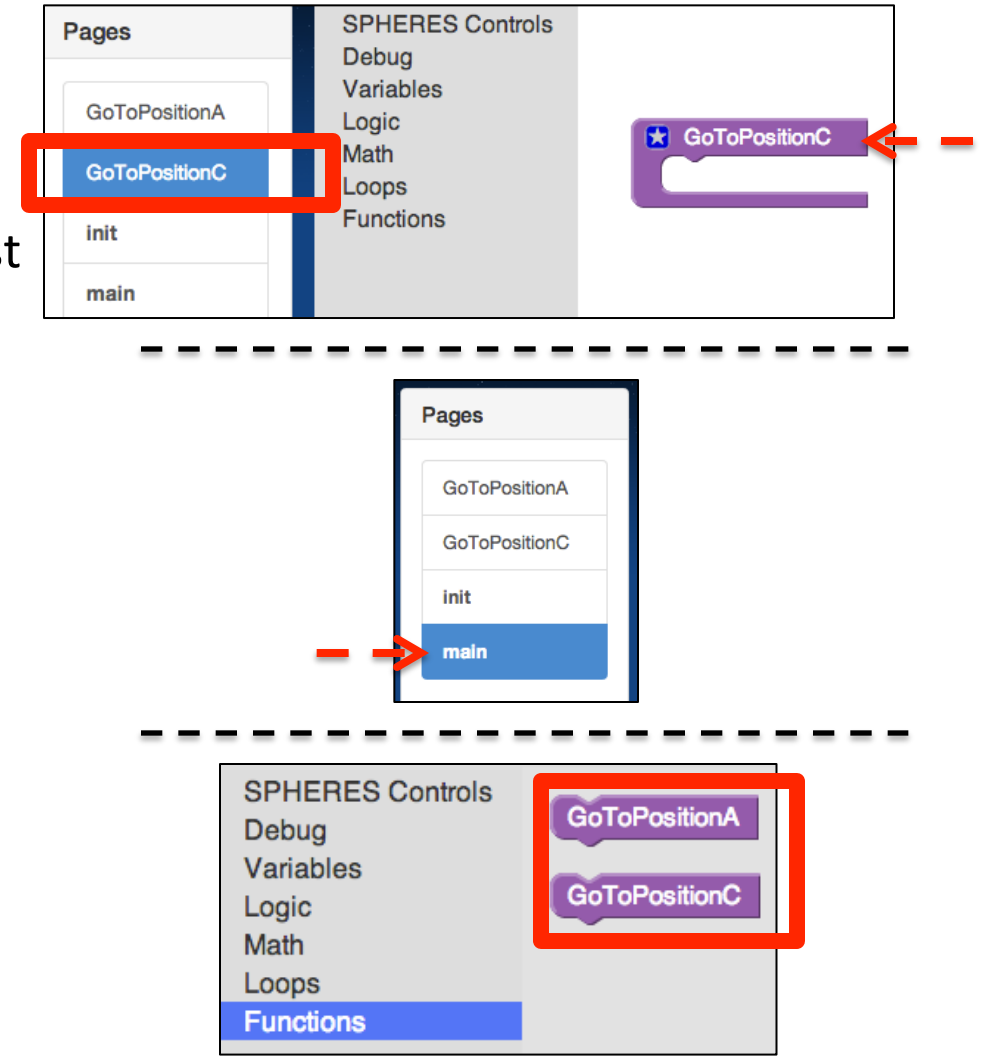






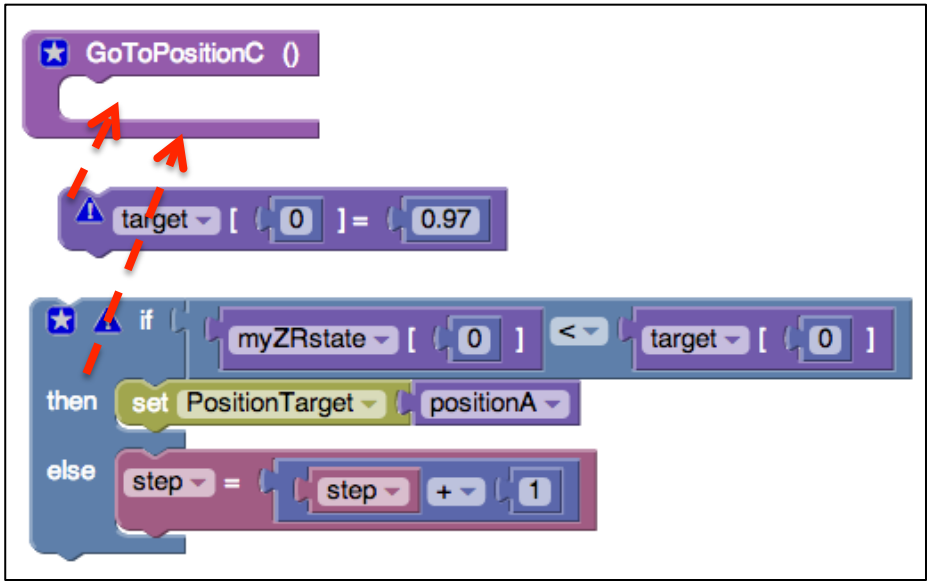
Create "GoToPositionC" Function (cont.)

- Your new page will open to a loop called "GoToPositionC"
  - Do not add anything to the loop yet
- Your new page will show up in the list of pages
- Click on the "main" page to return to your main loop.
- On the "main" page, open the Functions accordion (as shown)
  - You will see both functions: "GoToPositionA"
  - "GoToPositionC"



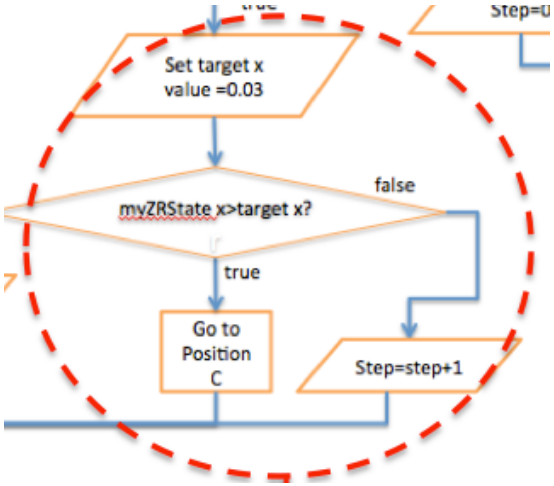
Create "GoToPositionC" Function (cont.)

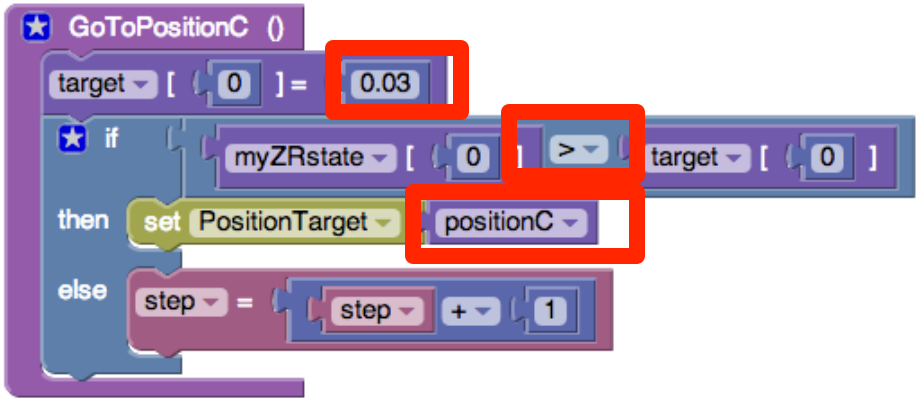
- The two functions are similar so to save time you will copy and paste the code from the "GoToPositionA" page into the "GoToPositionC" page and then edit
  - In the GoToPositionA" loop click on the **`target[0]=.97`** block and select "Edit" then "Copy" from the top menu.
  - On the Pages menu, open the page: "GoToPositionC"
  - Select "Paste"
  - Repeat for the **`If-Then-Else`** block
  - Drag and Drop the pasted code into the "GoToPositionC" loop
- This code now needs to be edited to send the satellite to positionC (instead of positionA) as described on the next slide



Create "GoToPositionC" Function (cont.)

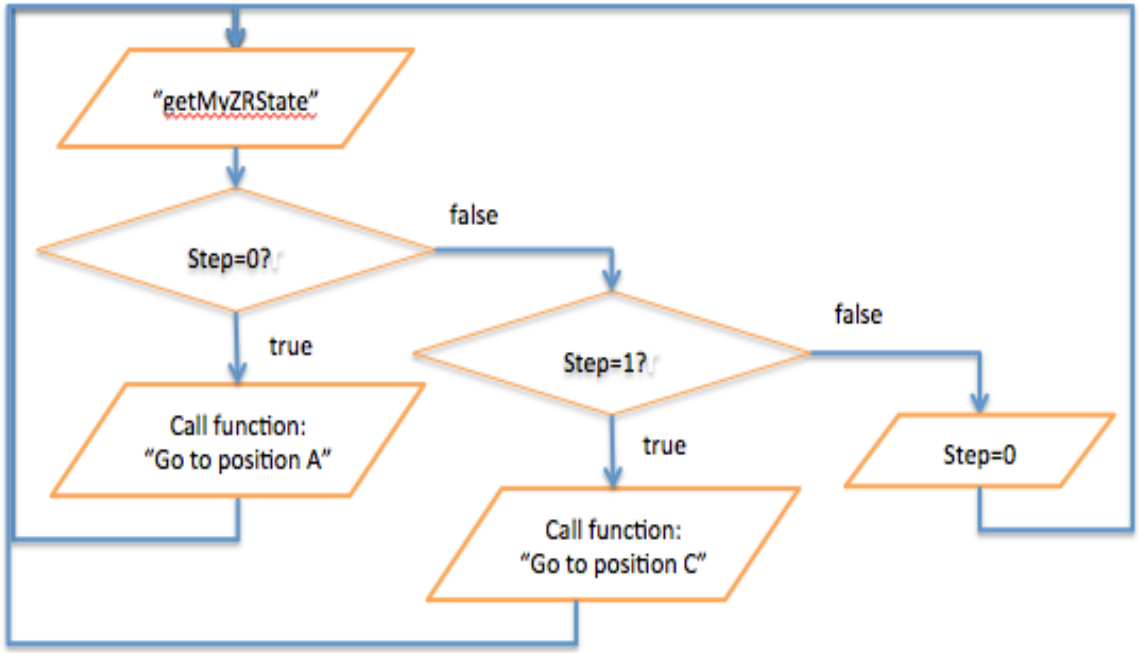
- The portion of the flow diagram that sends the satellite to position C is shown to the right as a reference
- Change "target [0]=.97" to: "target [0]= 0.03"
- Change the conditional statement from myZRState[0]<target[0] to myZRState[0]>target[0] :
  - Use the drop down menu to change the "<" to ">"
- Change the toggle on the setPositionTarget block to "positionC"
- Step=step+1 should already be in your program
- Your function "GoToPositionC" is complete!





Using the Step Counter Model

- The next step is to go back to the main loop and create the program shown below
- This program uses a step counter and "calls" the functions



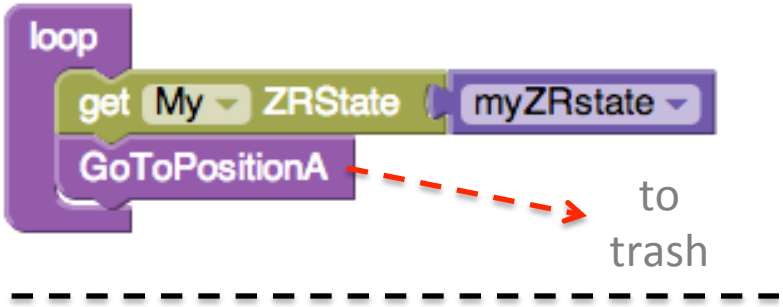
Using the Step Counter Model (cont.)

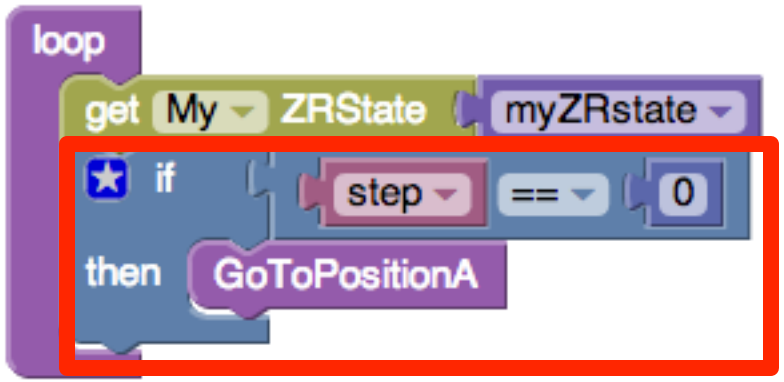
- Click on "main" on the pages menu to go back to the main loop
- Drag the "GoToPositionA" function (left over from program you started with) to trash
- Drag an "**`If-Then`**" block from the Logic accordion into the loop after the "GetMyZRState" block
- Change the "if" statement to:

  "If step == 0" then "GoToPositionA"

as follows:

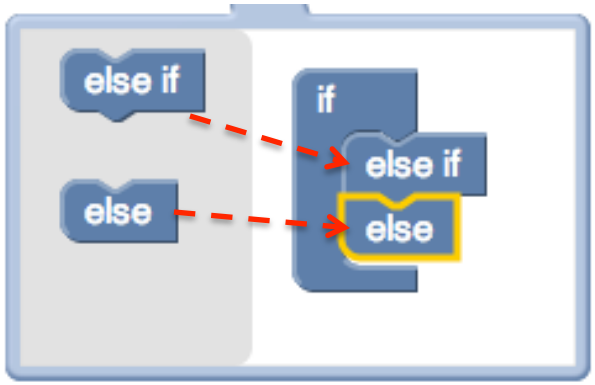
  - Drag a "__==__" Block from the logic accordio
  - Drag a pink "**—Select—**" variable block from the variables accordion into the first "__"
  - Drag number block (0) from the math accordion into the 2nd "__"
  - Drag "GoToPositionA" block from the Functions accordion into the "then" slot





Using the Step Counter Model (cont.)

- Click on the white star on the if statement and add “else-if” and “else” statements

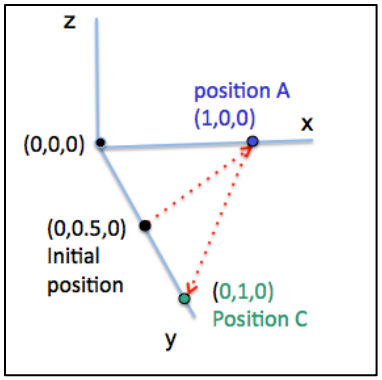


- Change the “else-if” statement to :
  “else-if step == 1 then goToPositionC”
  - Copy “**step==0**” from the if statement above, paste it into the else-if statement and then change the “**0**” to “**1**”
  - Drag “**GoToPositionC**” block from the Functions accordion into the “then” slot

- Change the “else” statement to:
  “else step=0”
  - Drag “`--Select-- = 0`” block from the Variables accordion into the “else” slot and toggle to “`step`”
  - Leave the number=0.

View Simulation

- Your code is complete!
- Compile, Simulate
  - Maximum Time: 180 seconds
  - View simulation



- The C Code for the separate pages “loop”, GoToPositionA”, and “GoToPositionC” is shown below:

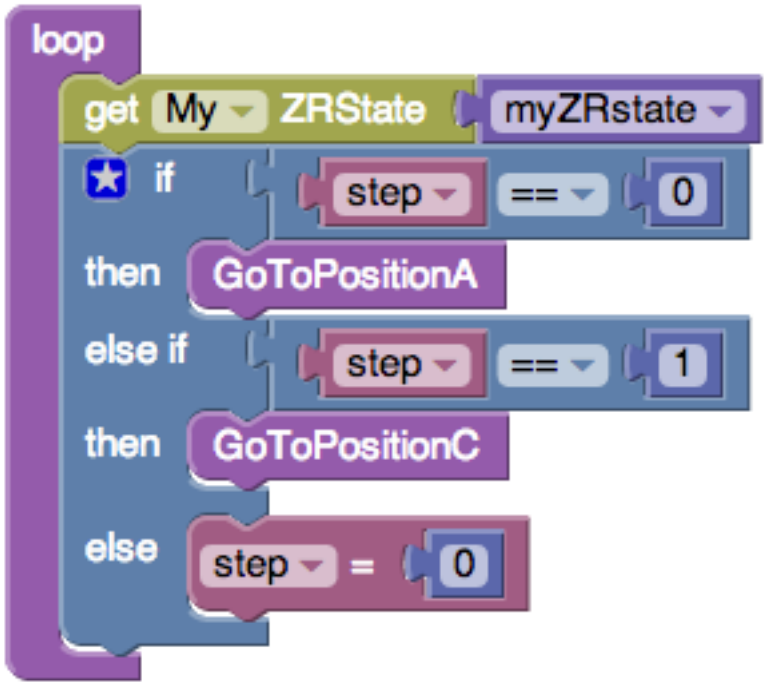
```
1  void loop() {
2    api.getMyZRState(myZRstate);
3    if (step == 0) {
4      GoToPositionA();
5    } else if (step == 1) {
6      GoToPositionC();
7    } else {
8      step = 0;
9    }
10 }
```

```
1  void GoToPositionA() {
2    target[0] = 0.97;
3    if (myZRstate[0] < target[0]) {
4      api.setPositionTarget(positionA);
5    } else {
6      step = step + 1;
7    }
8  }
```

```
1  void GoToPositionC() {
2    target[0] = 0.03;
3    if (myZRstate[0] > target[0]) {
4      api.setPositionTarget(positionC);
5    } else {
6      step = step + 1;
7    }
8  }
```

Using Functions

- The program you just created resets the step counter to zero in the last “else” statement.
- If you wanted to program the satellite to go to another position after going to positionC, can you see how this would be done?
  - You would:
    - Create a new function
    - Add another “else-if” statement
    - Call the new function
- When you program your SPHERES for the game you will probably use a series of Else-If statements with multiple steps
- Using functions will also make it easier for you to figure out which part of your program needs debugging



Review

- Congratulations!
  - You have learned how to:
    - Use a Step Counter
    - Create multiple functions
  - You are just about ready to start programming your game!