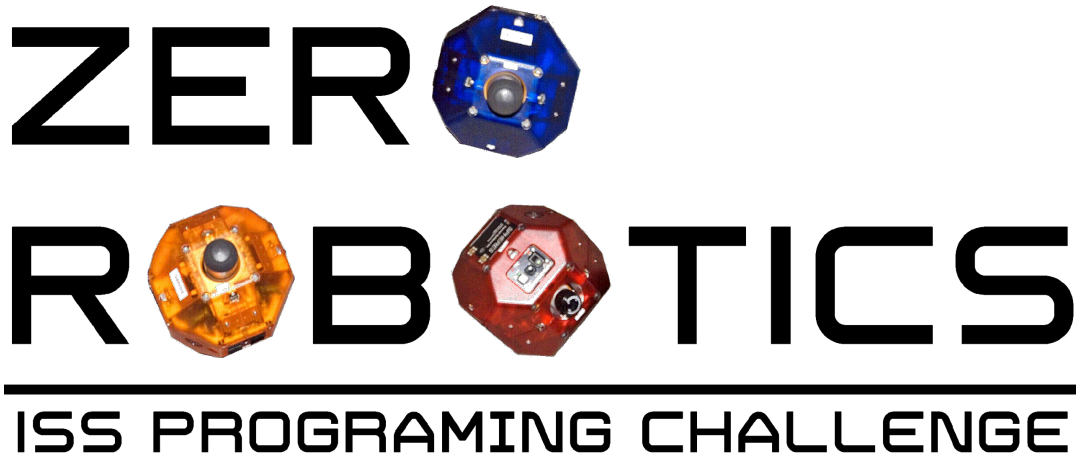


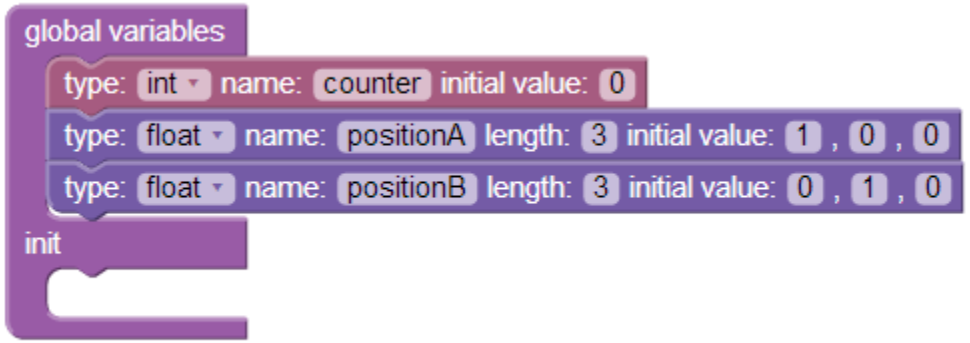
Conditionals with Advanced Logic Operators ("and" and "or") (Project 6)


Goals

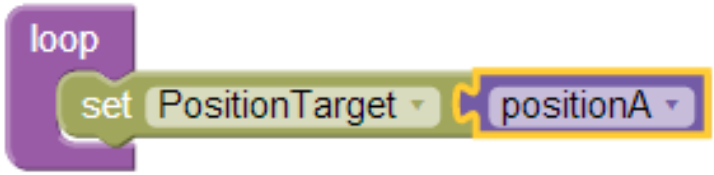
In this tutorial you will learn to use the logic operators **“and ”** and **“or”** in conditionals.

<	“and”	>	“or”

Create a New Project



- Open the ZR IDE
- Select "New Project"
  - Project name: **Project 6**
  - Editor: Graphical Editor
  - Game: FreeMode
- Declare Variables/Arrays on the Init page
  (Go back and look at Project 4 if you need help with how to declare variables)
  - "**counter**" (integer, initialized to **0**)
  - "**positionA**" (float, 3, initialized to **1,0,0**)
  - "**positionB**"(float, 3, initialized to **0,1,0**)
- Back in main, Add a SPHERES Control statement to **`setPositionTarget`** to **`PositionA`**
- Next we will add a conditional statement to tell the satellite when to go to **`PositionB`**.

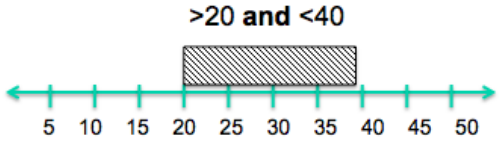


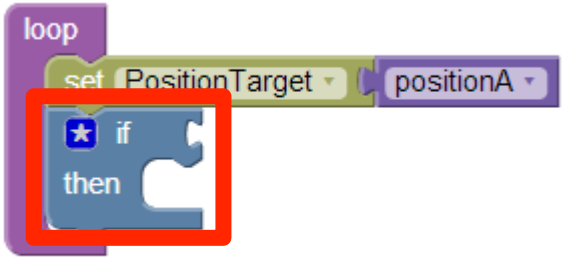
The Logic Operator "and"

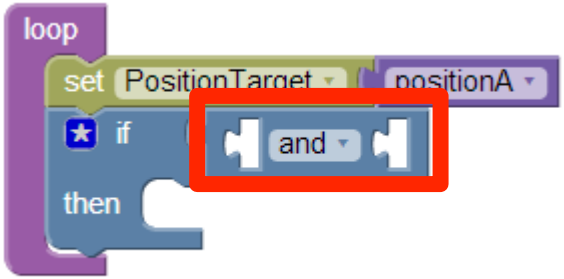
- Create the following "If-Then" statement in your loop using the logic operator **and**:

  "If counter > 20 **and** counter < 40 then…
  (go to positionB.)"

- First steps:
  - Drag an "If-Then" block from the Logic accordion
  - Drag an "and" block from the Logic accordion





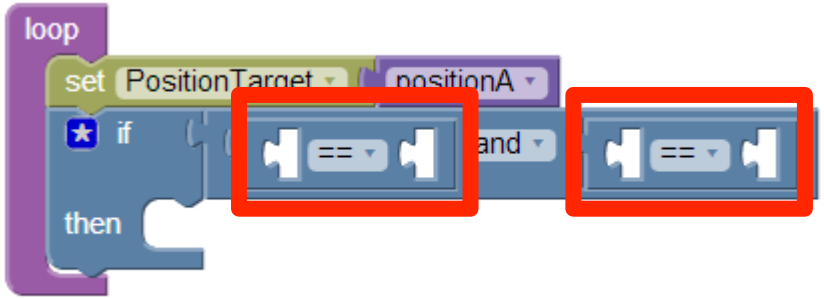


The Logic Operator “and” (cont.)

- Remember the “If-Then” statement is:

“If counter > 20 and counter < 40 then… (go to positionB.)”

- Next:
  - Drag an “__==__“ block from the Logic accordion into the first empty space in the “and” block
  - Drag another “__==__“ block from the Logic accordion into the second empty space in the “and” block
  - Change the first “==“ to a “**>**” in the dropdown menu
  - Change the second “==“ to a “**<**“ in the dropdown menu

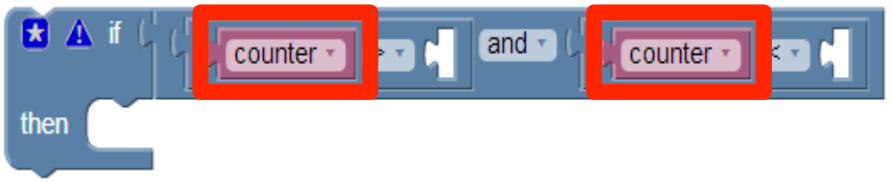


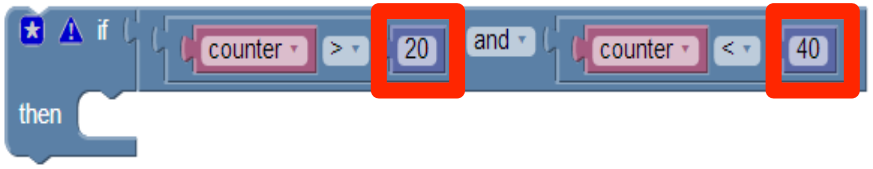
The Logic Operator "and" (cont.)

- Remember the "If-Then" statement is:

"If counter > 20 and counter < 40 then… (go to positionB.)"

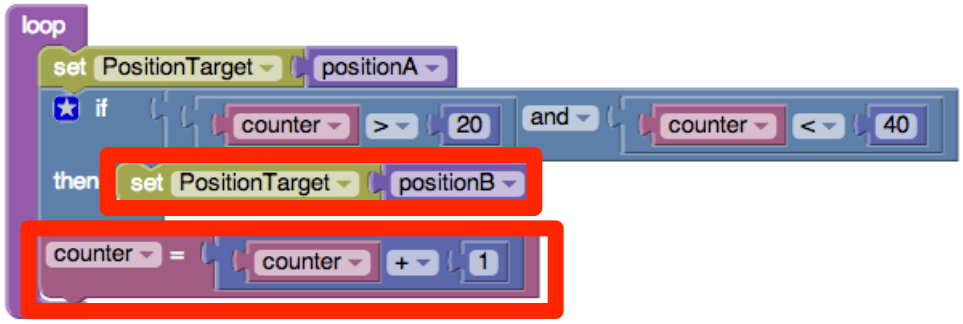
- Next:
  - Drag two pink Variable blocks from the Variables accordion and place them in the first empty slots of both the ">" and "<" blocks
  - Select "counter" in the dropdown menu for each
  - Add two blue Number blocks from the Math accordion and place them in the remaining empty slots of the ">" and "<" blocks
  - Enter 20 in the first Number block
  - Enter 40 in the second Number block





The Logic Operator "and" (cont.)

- Remember the "If-Then" statement is:
  "If counter > 20 and counter < 40 then… (go to positionB.)"
- Drag a SPHERES Control statement into the If-Then block to **`setPositionTarget`** to **`positionB`**
- The last step is to increment the counter (set **`counter = counter + 1`**)
  - Drag the **`"--Select--=0"`** block from the Variables accordion. (Make sure to drop it into the loop **after** the If-Then block.)
  - Drag the "**+**" block from the Math accordion
  - Drag the **`counter`** block from the Variables accordion
  - Drag the number block from the Math accordion and set to 1

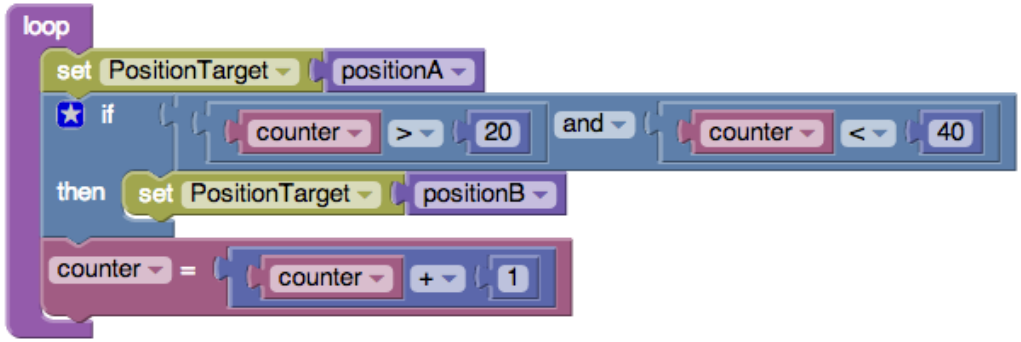


The Logic Operator “and” (cont.)

- What do you expect to happen?
  - Compile, Simulate
  - Maximum Time: 90 seconds
  - View simulation

Blue satellite should move from:
initial position➔ positionA ➔ positionB ➔ positionA

Compare: Your program - versus - C Code

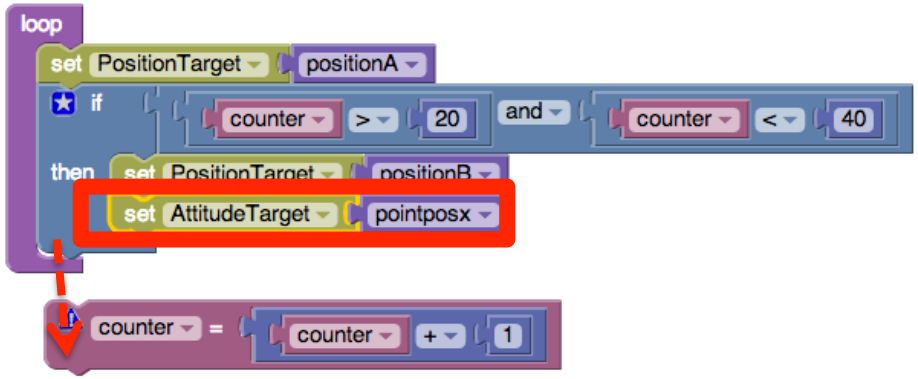



```
1  void loop() {
2    api.setPositionTarget(positionA);
3    if (counter > 20 && counter < 40) {
4      api.setPositionTarget(positionB);
5    }
6    counter = counter + 1;
7  }
```

Modify program

- Modify the program to change both the attitude and position of the satellite
- Create the following arrays:
  - **`float pointposx[3]`**
    - Set initial value to **1,0,0**
  - **`float pointnegx[3]`**
    - Set initial value to **-1,0,0**

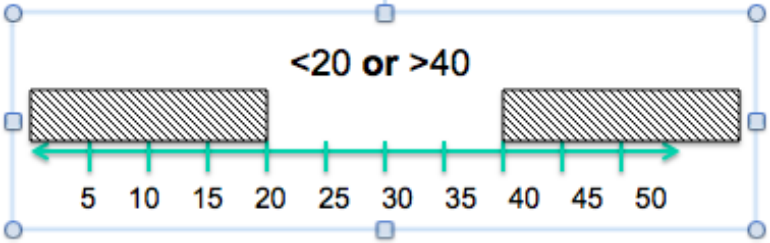
- Add the Spheres Control Function **`setAttitudeTarget`** into the If-then statement (toggled from setPositionTarget)
    - Select: **`pointposx`**
- Drag the **`counter = counter + 1`** statement out of the loop, but do not delete



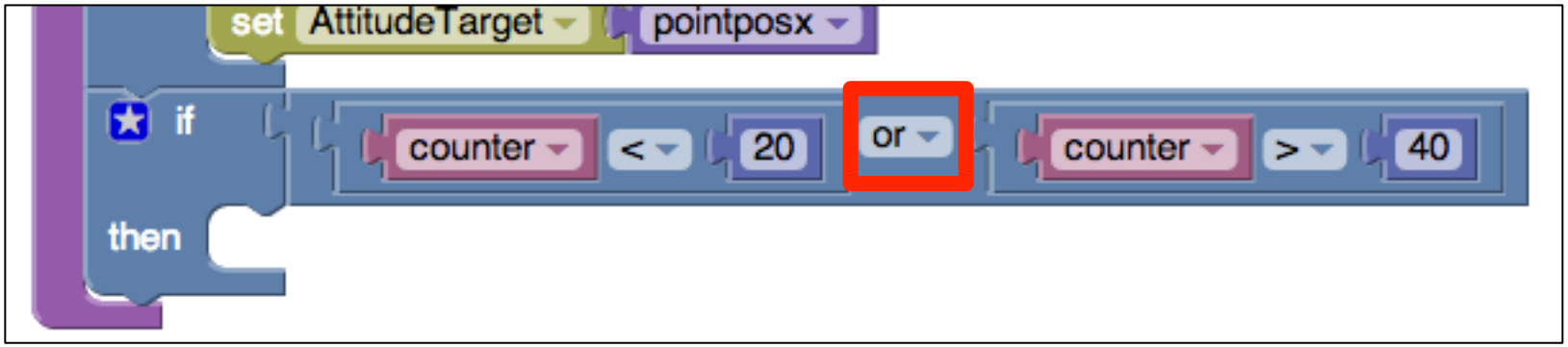
The Logic Operator "or"

- Add the "If-Then" statement:

"If counter < 20 **or** counter > 40 then…
(point in the negative x direction)"

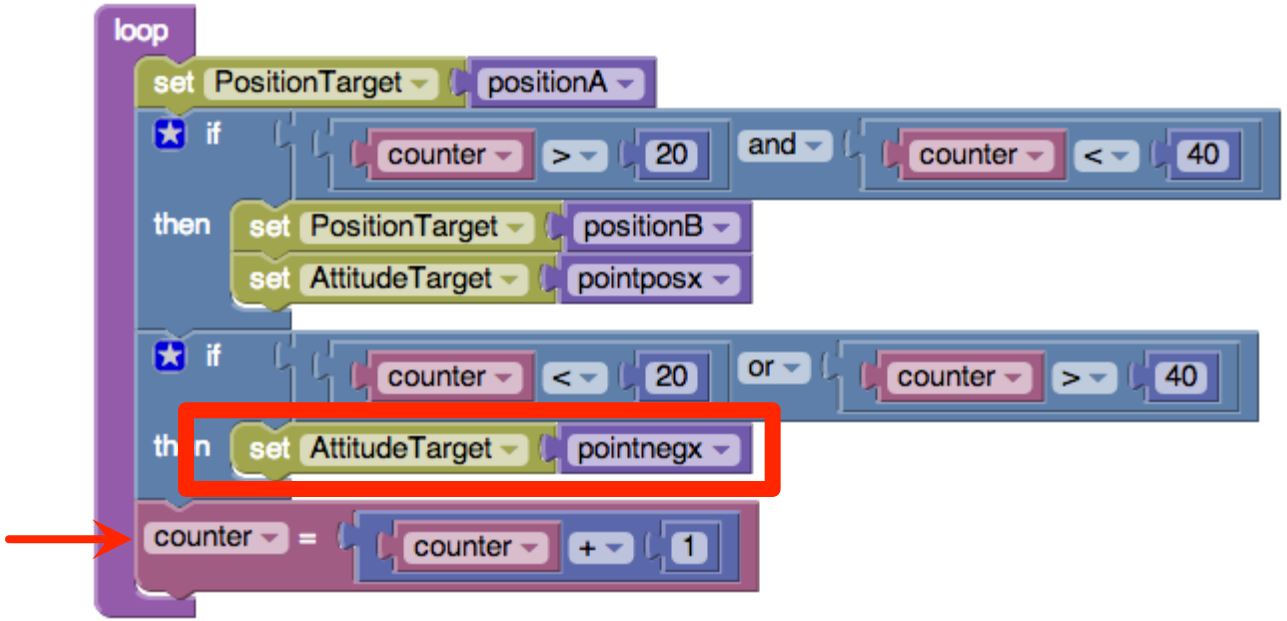


- Hints:
  - Drag the "If-Then" block from the Logic accordion
  - Drag an "and" block from the logic accordion and toggle to "or"
  - Drag "__==__" blocks from the Logic accordion into the empty spaces in the "or" block
  - Change the first "==" to a "**<** the second "==" to a "**>**"
  - Add **`counter`** blocks and numbers



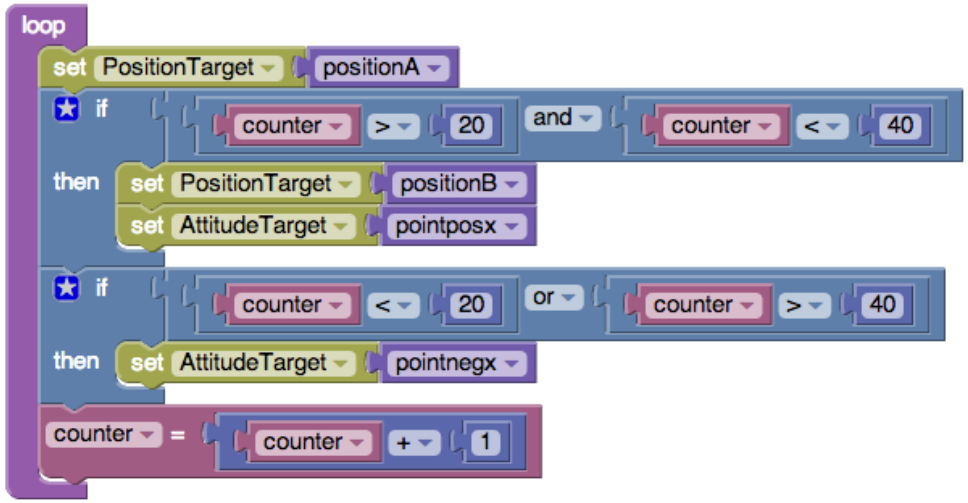
The Logic Operator "or" (cont.)

- Add the Spheres Control Function **`setAttitudeTarget`** into the new If-then statement
  - Select: **`pointnegx`**
- Drag **`counter = counter + 1`** back into the loop after the If-Then statement.



The Logic Operator "or" (cont.)

- What do you expect to happen?
  - Compile, Simulate
  - Maximum Time= 90 seconds
  - View simulation
- Compare: Your program - versus - C Code
  - What is the C code symbol for:
    - and
    - or




```
1  void loop() {
2    api.setPositionTarget(positionA);
3    if (counter > 20 && counter < 40) {
4      api.setPositionTarget(positionB);
5      api.setAttitudeTarget(pointposx);
6    }
7    if (counter < 20 || counter > 40) {
8      api.setAttitudeTarget(pointnegx);
9    }
10   counter = counter + 1;
11 }
```

Review

- Congratulations!
- You have learned two more logic operators: “and” and “or”
- You wrote a program that changes the SPHERES position and attitude