Conditionals: Advanced Logic Operators
• In this tutorial you will:
  – Use the logic operators “and ” and “or” in conditionals
  – Control the satellite’s translation and rotation simultaneously
Create a New Project

- Open the ZR IDE
- Select “New Project”
  - Project name: Project 6
  - Game: FreeMode
  - Text Editor
- Declare Variables/Arrays
  (Go back and look at Project 4 if you need help with how to declare variables)
  - int counter (initialized to 0)
  - float positionA[3] (initialized to 1.0f, 0.0f, 0.0f)
  - float positionB[3] (initialized to 0.0f, 1.0f, 0.0f)
- Add a statement to set the position target to positionA
- Next we will add a conditional statement to tell the satellite when to go to positionB, as follows.

```cpp
void loop()
{
  //This function is called once per second. Use it to control the satellite.
  api.setPositionTarget(positionA);
}
```
• `&&` is a logic operator that means “and”

• Create the following conditional statement in your loop using `&&`:
  “If counter > 20 and counter < 40 then...go to positionB”

```cpp
void loop()
{
  //This function is called once per second. Use it to control the satellite.
  api.setPositionTarget(positionA);
  if (counter > 20 && counter < 40) {
  }
}
```
The Logic Operator “and” (cont.)

- Remember the “If-Then” statement is:
  “If counter > 20 and counter < 40 then... go to positionB.”

- To make it go to positionB, we need to add the following:
  `api.setPositionTarget(positionB);`

- The last step is to increment the counter (outside the if statement.)
  `counter++;`

```java
22  void loop() {
23      // This function is called once per second
24      api.setPositionTarget(positionA);
25      if (counter > 20 && counter < 40) {
26          api.setPositionTarget(positionB);
27      }
28      counter++;
29  }
```
The Logic Operator “and” (cont.)

- What do you expect to happen?
  - Compile, Simulate
    - Load settings: Tutorial _90
  - View simulation

<table>
<thead>
<tr>
<th>Blue satellite should move from:</th>
</tr>
</thead>
<tbody>
<tr>
<td>initial position ➔ positionA ➔ positionB ➔ positionA</td>
</tr>
</tbody>
</table>
Modify program

• Modify the program to change both the attitude and position of the satellite
• Create the following arrays for setting other attitudes:
  – To point in the positive x direction: float pointposx[3] initialized to \{1.0f,0.0f,0.0f\}
  – To point in the negative x direction: float pointnegx[3] initialized to \{-1.0f,0.0f,0.0f\}
• Add the following into the If-then statement:
  “api.setAttitudeTarget(pointposx)”
The Logic Operator ||

- || is a logic operator that means “or”
- Add another “If-Then” statement that states the following: “If counter < 20 or counter > 40 then... point in the negative x direction” (Note: Make sure this goes before the counter statement.)

```cpp
void loop()
{
  //This function is called once per second
  api.setPositionTarget(positionA);
  if (counter>20 && counter<40)
  {
    api.setPoi…
  }
  if (counter<20 || counter>40){
  }
  counter++;
}
```
The Logic Operator \( \| \) (cont.)

- Add the following into the second conditional statement:
  \[ \text{api.setAttitudeTarget(pointnegx);} \]

- What do you expect to happen?
  - Compile, Simulate
    - Load settings: Tutorial _90
  - View simulation

```cpp
void loop(){
  //This function is called once per simulation
  api.setPositionTarget(positionA);
  if (counter>20 && counter<40){
    api.setPointPositionTarget(positionB);
    api.setAttitudeTarget(pointposx);
  }
  if (counter<20 || counter>40){
    api.setAttitudeTarget(pointnegx);
  }
  counter++;
}
```
• Congratulations!

• You have learned two more logic operators: `&&` and `||`

• You wrote a program that controls the satellite’s position and attitude simultaneously