Conditionals:
More Fun with “If-Then” and Logic Operators
Goals

• Great job so far! There are a lot of things you already know how to do! In this tutorial and in the following ones you will get a chance to start using what you have learned. Some familiar steps will have less detailed instructions. Look at hints or at previous tutorials if you need help, or ask a friend!

• In this tutorial you will:
  – Review: How your program controls the SPHERES satellites
  – Practice programming with “If-Then” statements
  – Use the logic operators “==” and “!=”
  – Learn about the “debug” feature
Zero Robotics Control System

- The program inside your loop is called **once per second** by the SPHERES control system.

- During each second:

```java
void loop() {
}
```

![Diagram showing the control flow of the Zero Robotics System]
• Let’s create a new project to learn more about the SPHERES control system
• Open the ZR IDE
• Select “New Project”
  – Project name: Project5
  – Game: FreeMode
  – Text Editor
• Create a variable called counter
  – Above `void init()` type `int counter;`
  – Inside `void init()` type `counter = 0;`

```cpp
// Declare any variables shared between functions here

int counter;

void init()
{
    // This function is called once when your code is first loaded.
    // IMPORTANT: make sure to set any variables that need an initial value.
    // Do not assume variables will be set to 0 automatically!
    counter = 0;
}
```
• “==” is a Logic operator that means “equals”

• Create the following “if-then” statement in your loop: If counter is equal to 10 then . . . (See example below)

```java
void loop()
{
  //This function is called once per second. Use it to control the satellite.
  if (counter == 10)
  {
  }
}
```

• Add `counter++;` after the “if-then” statement to increment the counter each second (see example below)

```java
void loop()
{
  //This function is called once per second. Use it to control the satellite.
  if (counter == 10)
  {
  }
  counter++;  // Increment counter each second
}
```
If-Then with the Logic Operator “==” (cont.)

• Next add a debug statement, which prints out messages.
  - To do that we type `DEBUG("")`;
  - Then type `counter is 10` into the “DEBUG” statement (within the quotation marks)
• You have created the following “if-then” statement:
  ```
  if counter == 10 then
  print the message:
  “counter is 10”
  ```
• The debug message can be any text you like
Viewing Debug Statements

- Since you have not set a position target in your program, the satellite will not move. Instead, watch for your debug message as follows:
  - Compile, Simulate
    - Load settings: Tutorial _45
  - **Before** you run your simulation
    - Click the “show console” icon on the bottom of your screen
    - A grey box will appear at the bottom of your screen
  - Watch for your debug message in this box when you run the simulation
    - Your debug message for blue SPHERES 1 (SPH1) will appear after 10 seconds (see example)
  - Run simulation
More on Debugging

- You can use the debug statement to print variables as well. Use the following symbols for each data type.

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>%d</td>
</tr>
<tr>
<td>float</td>
<td>%f</td>
</tr>
<tr>
<td>string</td>
<td>%s</td>
</tr>
</tbody>
</table>

- Use this syntax: `DEBUG(\"text text text symbol\", variable);`
- For example: `DEBUG(\"blahblah %d\", counter);`
- `DEBUG` uses the same syntax as the `printf` function in normal C. For more symbols to format data using `DEBUG`, you can refer to the following page (not created or supported by Zero Robotics):
Adding the logic operator !=

- “!=” is a Logic operator that means “not equal”
- What happens if you add the following if-then statement to your loop?
  
  If counter is not equal to 10 then....
- To do this, first:
  
  - We add another “if” statement right under our first one.
  - This conditional is written `counter != 10`

```cpp
void loop()
{
  //This function is called or
  if (counter == 10)
  {
    DEBUG("counter == 10");
  }
  if (counter != 10) {
  }
  counter++;
}
```
Adding the logic operator != (cont.)

- Next we’ll add another `DEBUG (( "\" \" \" ));` statement under the new “if” statement
- This time write the message:
  
  `DEBUG(("counter equals %d",counter));`
  
  (don’t forget the quotation marks and comma)
- Look at the program you created. What do you expect to happen when you run the simulation? Let’s find out.

```c
void loop(){
    if(counter == 10) {
        DEBUG(("counter is 10"));
    }
    if(counter != 10) {
        DEBUG(("counter equals %d",counter));
    }
    counter++;
}
```

- Compile, Simulate
  - Load settings: Tutorial _45
- **Before** you run your simulation
  - Click the “view console” icon on the bottom of your screen
• Remember that your program is being read every second.

• A “counter equals (number)” message was printed for each second that the counter did not equal 10 (starting from counter == 0)

• A “counter is 10” message was printed for each second that the counter was equal to 10
  - Can you find that message?

• DEBUG statements can help you check whether your program is running the way you expect it to run.
• Congratulations!
• You are becoming a pro at conditional statements!
• You learned two more logic operators == and !=
• You wrote a program that shows your loop runs once per second
• You learned how to use DEBUG statements!

```
DEBUG(("counter == 10");
DEBUG(("counter equals %d", counter));
```