More Simple Arrays
and the setAttitudeTarget Function
(Project 3 – part 1)
• In this tutorial you will:
  - Practice using arrays in programming
  - Learn about a new SPHERES control function:
    - setAttitudeTarget—allows you to rotate the satellite to face in whatever direction you want.
A thruster is used to propel (move) the SPHERES satellite in a certain direction.

There are 12 thrusters on each SPHERES satellite to help it move in 12 different directions.
  - 3 of the 12 thrusters are visible in the photo to the right.

How does this work?
  - A tank of carbon dioxide (CO₂) gas is attached to the SPHERES satellite.
  - Each thruster releases CO₂ from the SPHERES satellite, creating a force on the satellite in the opposite direction.

Multiple thrusters on different sides are activated to rotate the satellite to a specified pointing direction.
• Are you ready to write a program to rotate a satellite (control satellite attitude)?

• When you set the **position** of the SPHERES satellite, you created an array of 3 values \{x,y,z\}.

• To rotate (control the **attitude**) of the SPHERES satellite you will also need an array of 3 values \{Nx,Ny,Nz\}.

• Remember what you learned about arrays before?
• Okay, let’s get started
Create a New Project and a New Variable

- Select “IDE” from the menu at the top of the page
- Select “New Project”
  - Project Name: **Project 3**
  - Editor: Graphical Editor
  - Game: FreeMode
- Click “Create Project”
- Go to the Init page
- Declare an array called “attitude” to set the attitude of the SPHERES satellite as follows:
  - Go to the Init page
  - Drag the array initialization block from the Variables accordion
  - Name: “attitude”
  - Length: 3
  - Leave initial values at 0, 0, 0
- Go to the main page to assign values to your array
- Click on the Variables accordion
- Drag the purple "--Select--[0] = 0" (array assignment) block into the loop
- Use the drop down menu to select "attitude"
- Repeat 2 more times
  (You need 3 attitude [0] = 0 blocks since your array has 3 members)
Assign Values to Your Array, cont.

- Change the `[bracketed]` part of each “attitude[0] = 0” block as follows:
  - In the first (top) block, leave “0” selected (This will represent the “x” direction)
  - In the second block change to a “1” (This will represent the “y” direction)
  - In the third (bottom) block change to a “2” (This will represent the “z” direction)

- In the first (top) block, change the right-most “0” to “1.0”
• The SPHERES Control Function “setAttitudeTarget” allows you to set the direction for the satellite to point its Velcro (-X) face.
• Attitude specifies a pointing direction \((N_x, N_y, N_z)\), not a pointing location.
• Commanding an attitude target makes the satellite fire thrusters to rotate to the target direction, then stop.
Add setAttitudeTarget Function

• Create a statement to set the attitude of the SPHERES satellite
  • Click on the SPHERES Controls accordion
  • Drag the setPositionTarget block into your loop
  • Using the first dropdown menu, change `PositionTarget` to `AttitudeTarget`
  • Use the second dropdown to select `attitude`

• This program you’ve created tells the SPHERE to move to the target attitude defined by the array “attitude[3]”
  • The Velcro face of the SPHERES satellite will rotate to point in the positive x direction
• Compile, Simulate
• In the Simulation Settings pop-up box:
  – “Maximum Time”:
    • Change from 90 seconds to 20 seconds
• Click “Simulate” button

• NOTE: Before playing the simulation
  – Click on the zoom-in tool at the bottom of the screen 10-12 times
• Start the simulation
  – The visible face on the SPHERES satellite will change as the satellite rotates to point in the positive x direction.
  – Look at the scoring box (top-left corner of the screen with blue label) which provides information about the blue SPHERES satellite:
    • Started at Ny = 1.00
    • Ended at Nx = 1.00 (pointing in positive y direction) (pointing in positive x direction)
Modify Program, Compile & Simulate

- Close the simulation window to return to the Graphical Editor

- Next try pointing in the negative x direction

- Change: “attitude[0] = 1” to: “attitude[0] = -1”

- Important Notes:
  For these exercises, point the satellite by setting only one of the values [0], [1], [2] to +/-1 and leave the rest set to 0 as shown in the table.

- “Compile” and “Simulate” as before

- “View Results”

<table>
<thead>
<tr>
<th>Set</th>
<th>+/- x direction</th>
<th>+/- y direction</th>
<th>+/- z direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>set [0] =</td>
<td>+/-1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>set [1] =</td>
<td>0</td>
<td>+/-1</td>
<td>0</td>
</tr>
<tr>
<td>set [2] =</td>
<td>0</td>
<td>0</td>
<td>+/-1</td>
</tr>
</tbody>
</table>
• Remember the program you wrote with “setPositionTarget” (from last time) and its C code?

```c
void loop() {
    position[0] = 2;
    position[1] = 2;
    position[2] = 0;
    api.setPositionTarget(position);
}
```

• Can you figure out the C code for the “setAttitudeTarget” program you just wrote?

Your C code here
If the C code shown below matches what you thought - You are right!

```c
void loop() {
  attitude[0] = -1;
  attitude[1] = 0;
  attitude[2] = 0;
  api.setAttitudeTarget(attitude);
}
```
• Congratulations!
• You are getting good at programming with arrays!
• You know how to program a SPHERES satellite to rotate and point in a specific direction!