ZERO ROBOTICS

ISS PROGRAMING CHALLENGE

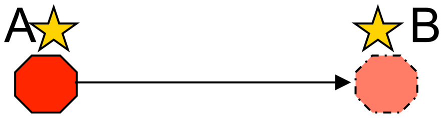
Getting to Know the ZR IDE

(Project 1)

Goals

In this tutorial you will use the ZR IDE (Integrated Development Environment) to:

- Create a new project
- Create a new variable
- Create code to move a SPHERES satellite along a single axis

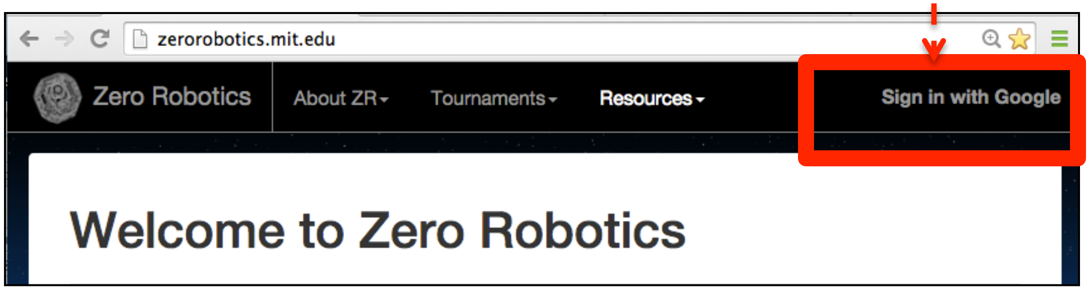


- Compile your code (check it for errors)
- Simulate (run the code in a simulation)

Log In

- Go to the Zero Robotics website:

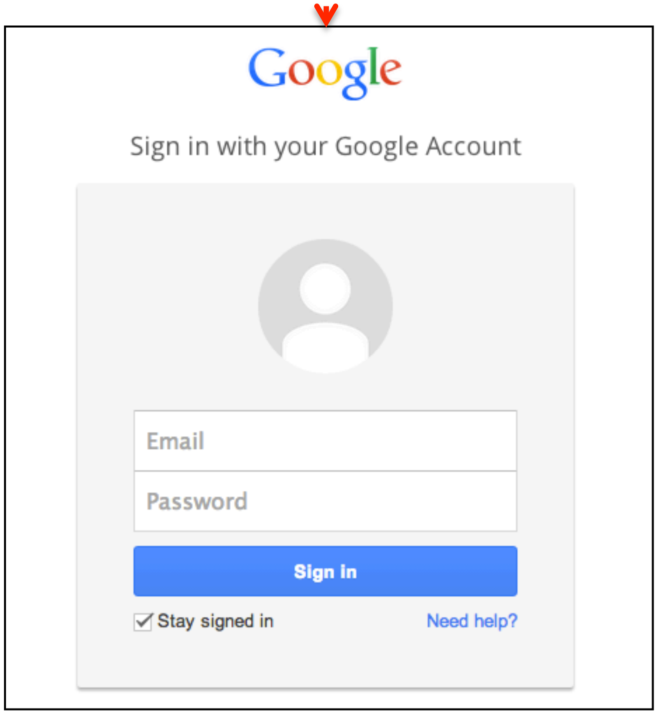
  zerorobotics.mit.edu



zerorobotics.mit.edu

Welcome to Zero Robotics

- Click Sign In with Google
- If required: Log into your account with your email and password

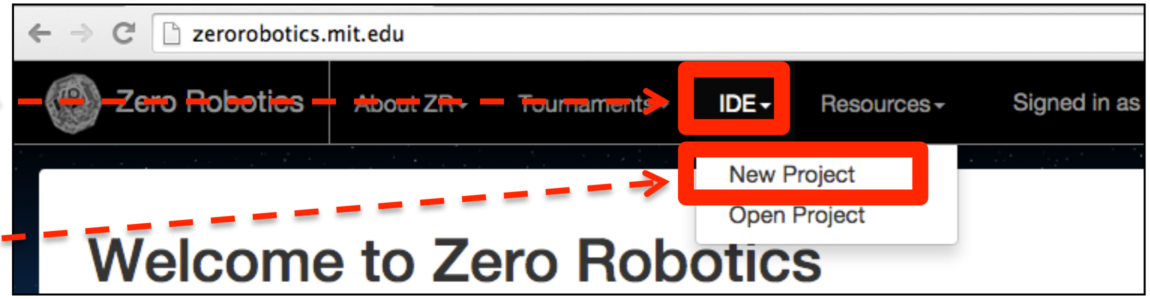


Google

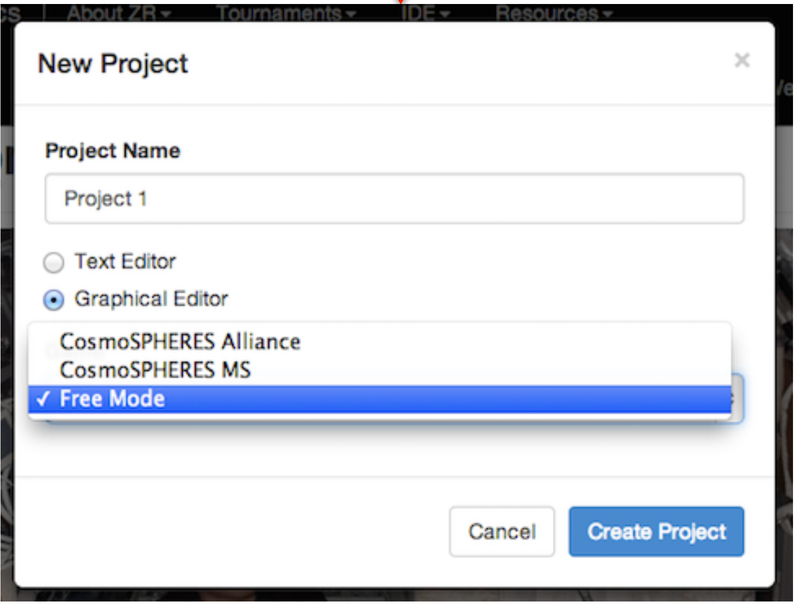
Sign in with your Google Account

Create a New Project

- Select the IDE dropdown menu on top ribbon
- Select " New Project"



- Enter
  - Project Name
    - Type: Project 1
  - Select "Graphical Editor"
  - Select "FreeMode"
- Click "Create Project"



Project Name: Project 1

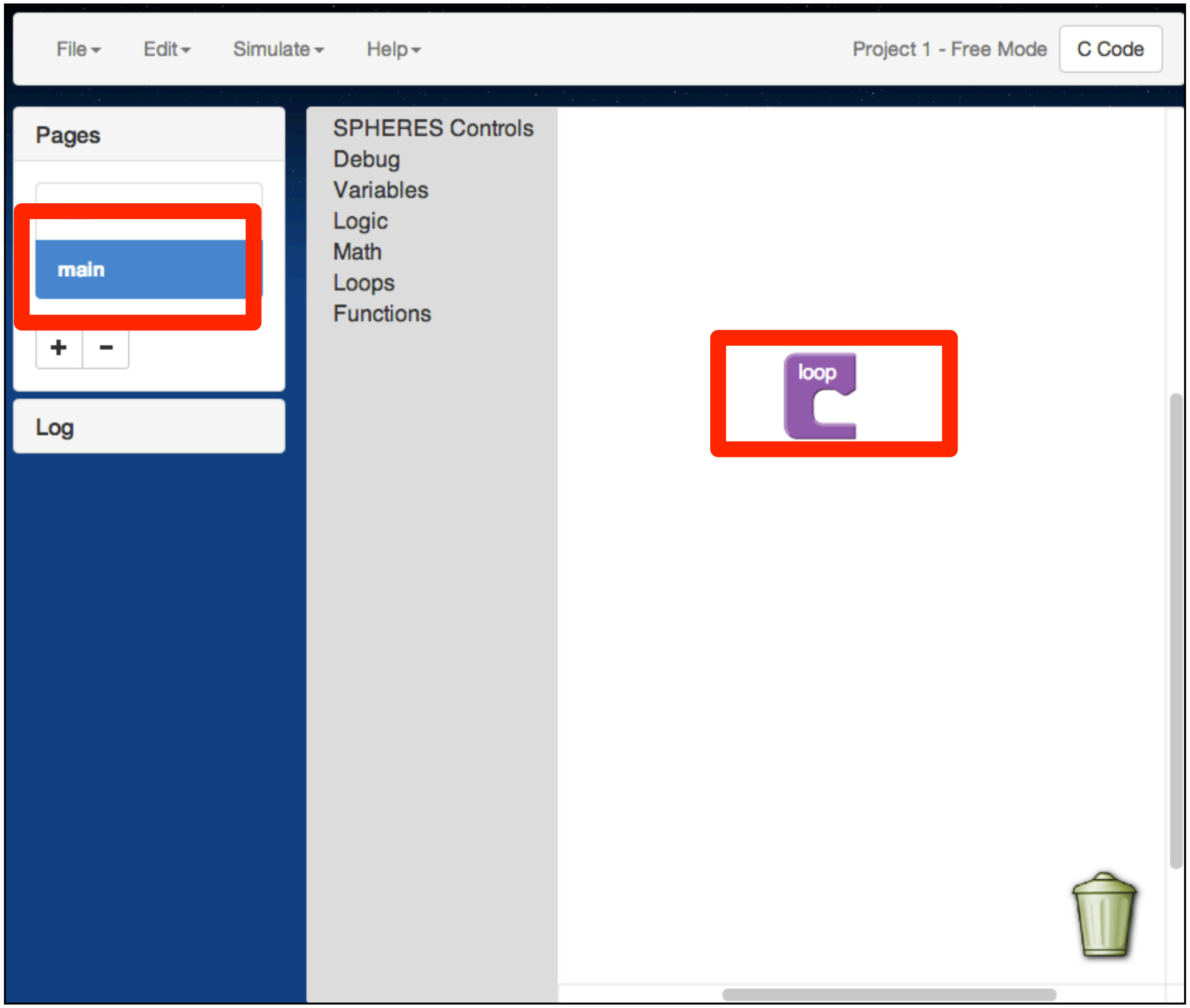
○ Text Editor
● Graphical Editor

CosmoSPHERES Alliance
CosmoSPHERES MS
✓ Free Mode

Graphical Editor IDE

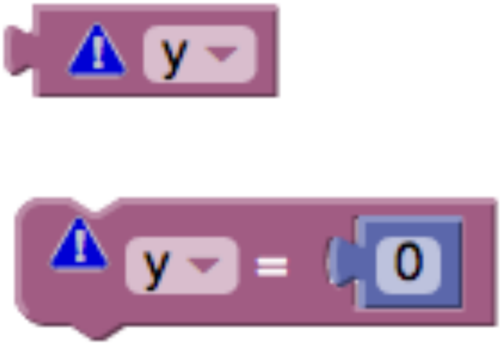
- The Graphical Editor version of the ZR IDE is shown here
  - This is the “main” page
  - You will write code in the “loop” shown here

- On the next pages, you will:
  - Review what you know about variables
  - Create a new variable



Declaring Variables

- A variable is a container that holds a single piece of a certain type of data.
- Before you use a variable in your program you must “make” it first. To do this, you must tell the computer:
    - The **type** of information the variable will hold (say, a number)
    - The **name** of the variable — like a label on the container so you can find it and use it
- This is called **declaring** the variable.



Variable Types

The two variable types you will use most often are:

- Integers (int)
  - A whole number, positive or negative, including the number 0.
  - Integers are NOT allowed to have decimals
- Floating-Point Numbers (float)
  - A number, either positive or negative, that has at least 1 digit after the decimal.
  - Floats allow for decimal values
  - Numbers should end with f to show that they are float values
- Attempting to put the wrong type of data into a variable (for example, putting a float value into a variable declared as an int) will cause an error.

ints:
0, 1, 2…
-1, -2, -3…
17, 100

floats:
1.1f, 2.0f,
-5.111111f,
3.69f

Naming Variables

Rules for naming variables in C++

- Use only letters, numbers, and underscores _
- Do not use spaces or punctuation symbols
- Begin the name with a letter not a number (1,2,3) or underscore _
- Do not make two variables with the same name, even if they are different types
- Do not make a variable whose name already means something else in C++, like “int” or “switch” — you will learn more of these keywords later

Variable Names Quiz

Which of these variable names are OK?

- Y
- 3position
- five
- float
- Position_3
- _Position3
- p%

Naming Variables

Answers:

–Y – Good

–3position – Bad (starts with a number)

–five – Good

–float – Bad (C++ keyword – specifically, a type of variable)

–Position_3 – Good

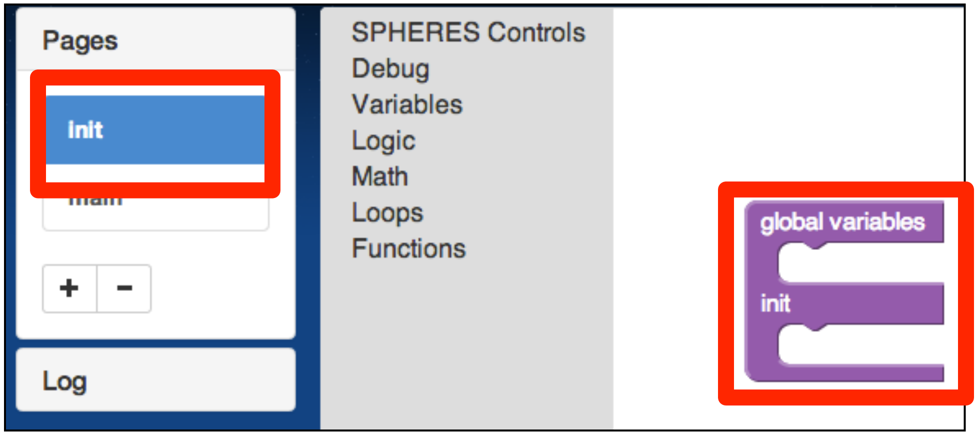
–_Position3 – Bad (starts with an underscore)

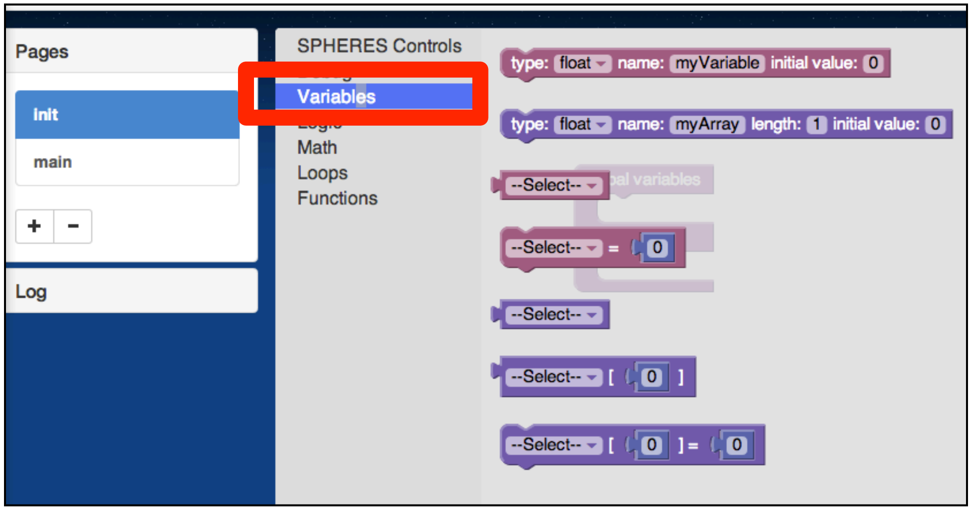
–p% – Bad (illegal symbol – %)

Declaring a New Variable

- Declare a variable (called “Y”) to set the position of the SPHERES satellite

1. Variables are declared on the “Init” page (short for initialization page)
   - Click on **`init`** to open the init page
   - You will see a “global variables” loop





2. Click on the Variables accordion
   - All the pink blocks are used for **variables**
   - All the purple blocks are used for **arrays**, which you will learn about later

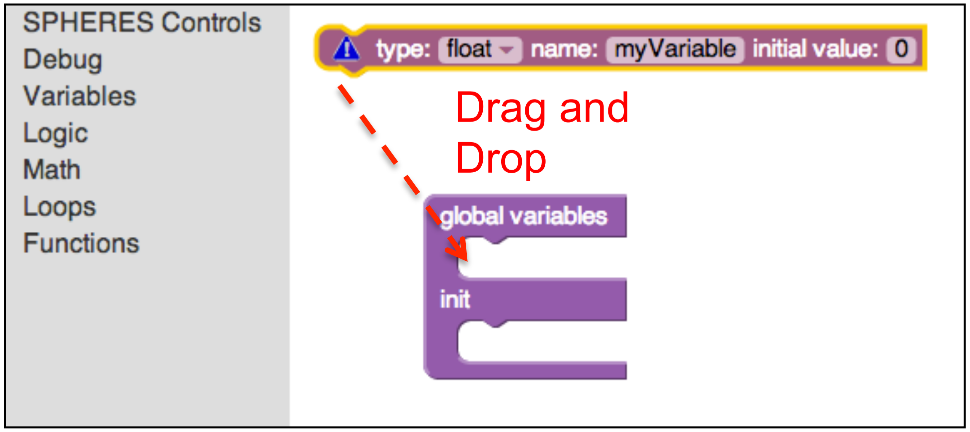
Declaring a New Variable

3. The top pink block is used to declare a variable:
   - Click on the top pink block and it will appear on the **init** page
   - Drag and drop the block into the top slot of the global variables loop as shown
     - Align the puzzle end of the block with the slot (slot will highlight in yellow) to drop the block in place

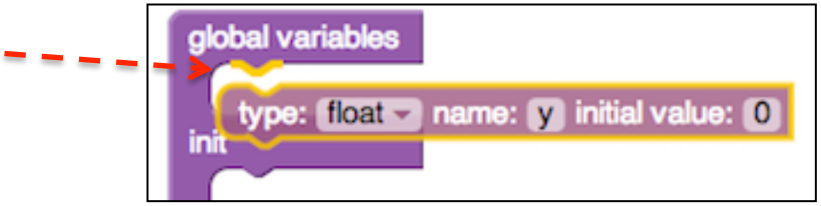
4. Declare what **type** of variable the variable will be: (The two types you know so far are **int** and **float**.)
   - Type: choose "float" as shown

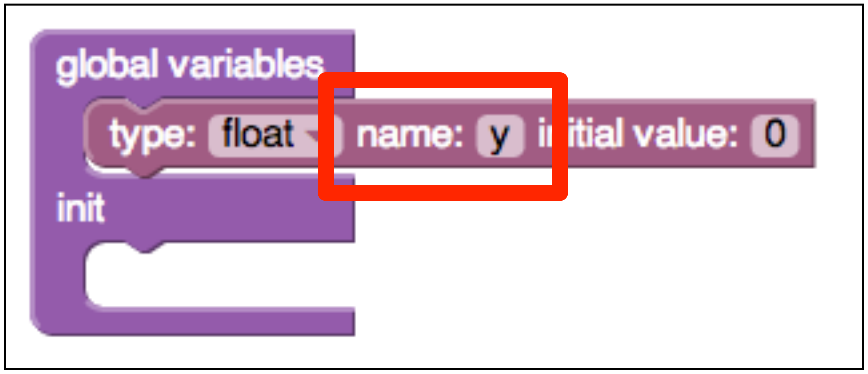
5. Enter **name** :
   - Name: enter the letter "y"

6. Initial value: Leave initial value = 0



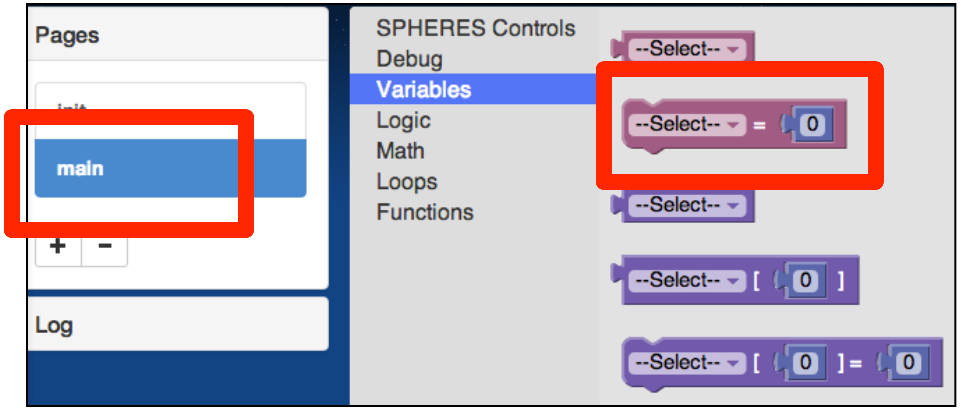
Drag and Drop

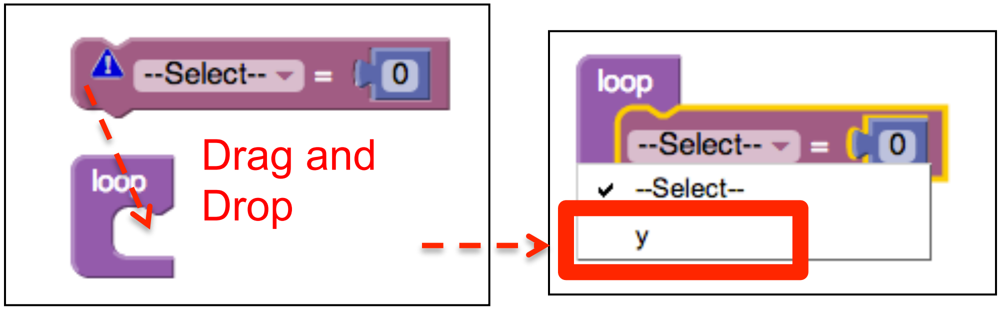


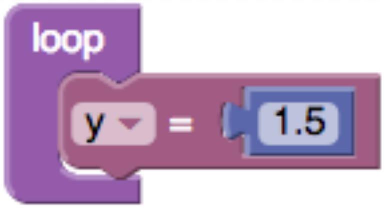


Assign a Value to Your Variable

- Now that we've created the variable Y, we need to put a number in it.
- We will do this on the main page
    1. Click on "**main**" page
    2. Click on Variables accordion
    3. Click on the pink "**`--Select-- = 0`**" block
    4. Drag and Drop the block into the **loop**
    5. Use the drop down menu to select "**y**"
    6. Delete "0" from the number field and type "**1.5**"

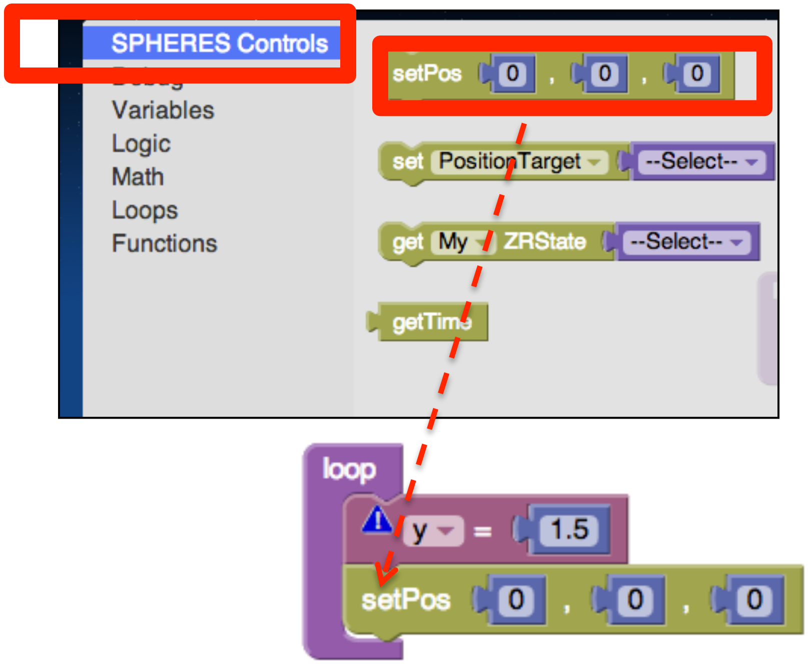






Add “SPHERES Controls” Function

- Create a statement to set the position of the SPHERES satellite
  1. Click on the “SPHERES Controls” accordion
  2. Click on the “setPos 0,0,0” (set position) block and drag the block and drop the block inside the “loop” below the Y = 1.5 block.



Add "SPHERES Controls" Function

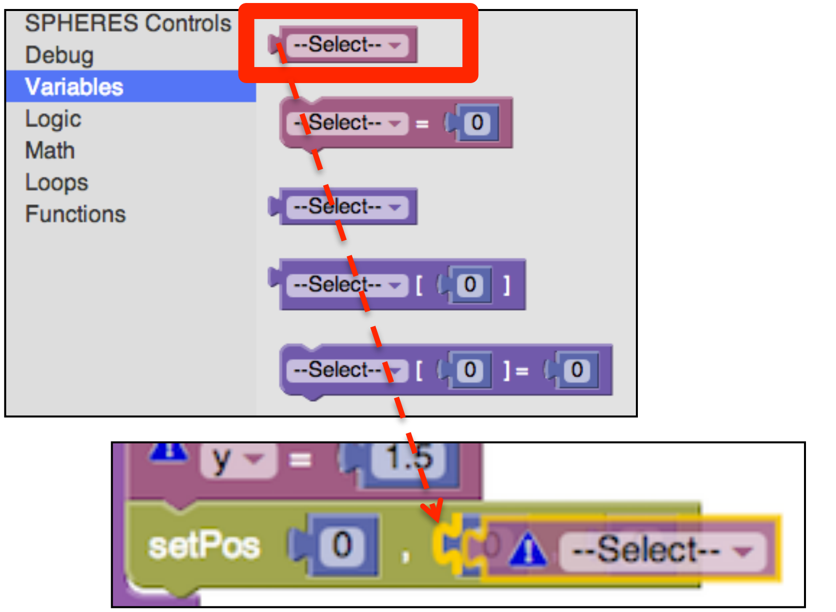
3. Click on the "Variables" accordion

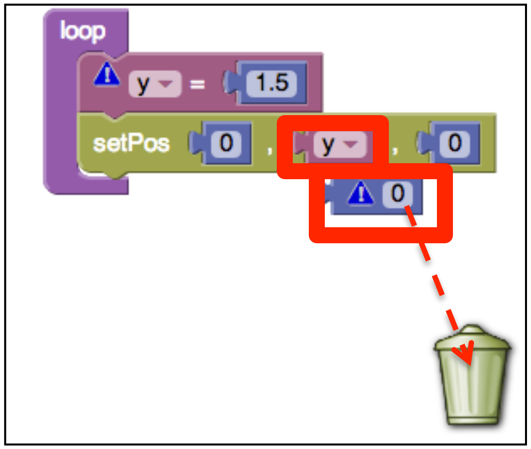
4.Click the pink "**--Select–**" variable block and drag and drop the block into the middle zero (0) in the "**setPos**" block.

- Align the puzzle end of the block with the slot (slot will highlight in yellow) to drop the block in place

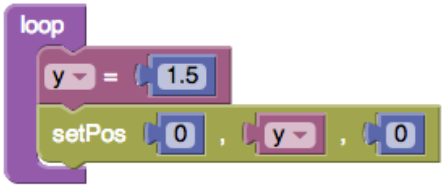
5.Toggle the drop down menu in the block to "**y**"

6.Drag the middle zero (0) that popped out of the block to the trash can



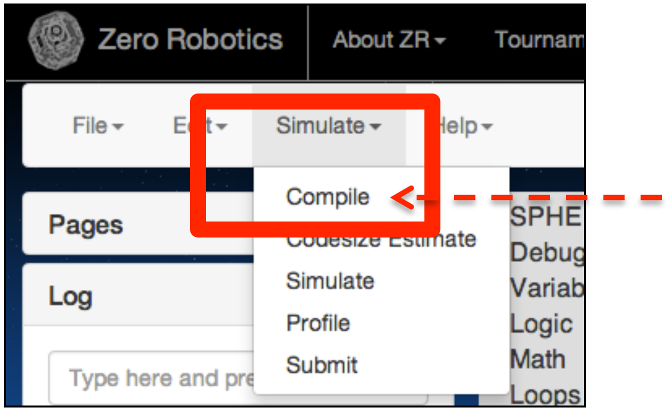


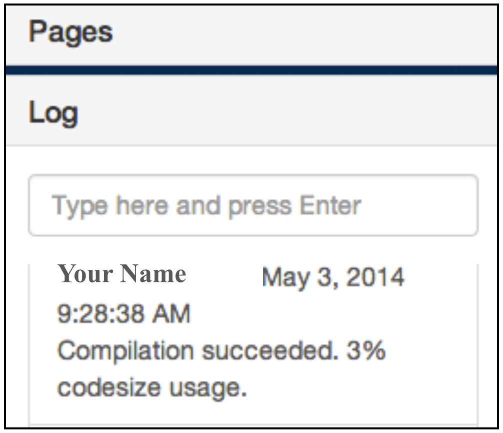
- This program you've created tells the satellite to move to the position on the Y axis defined by the variable "Y"
  - The satellite will move along the y-axis and stop at the y=1.5 position



Quick Compile

- Now let’s see your program in action!
- Compile:
  - Click on “Simulate” (top menu, third from the left)
  - On the pull down menu, click on ”Compile”
- A “Running” window will pop up while the program is being compiled
- After compiling:
  - The log will open with a compilation succeeded or failed message.
  - If compilation failed check your code and try again
    - Common errors:
      - ➢ Variables not declared on init page
      - ➢ Drop down menus on puzzle blocks not toggled correctly

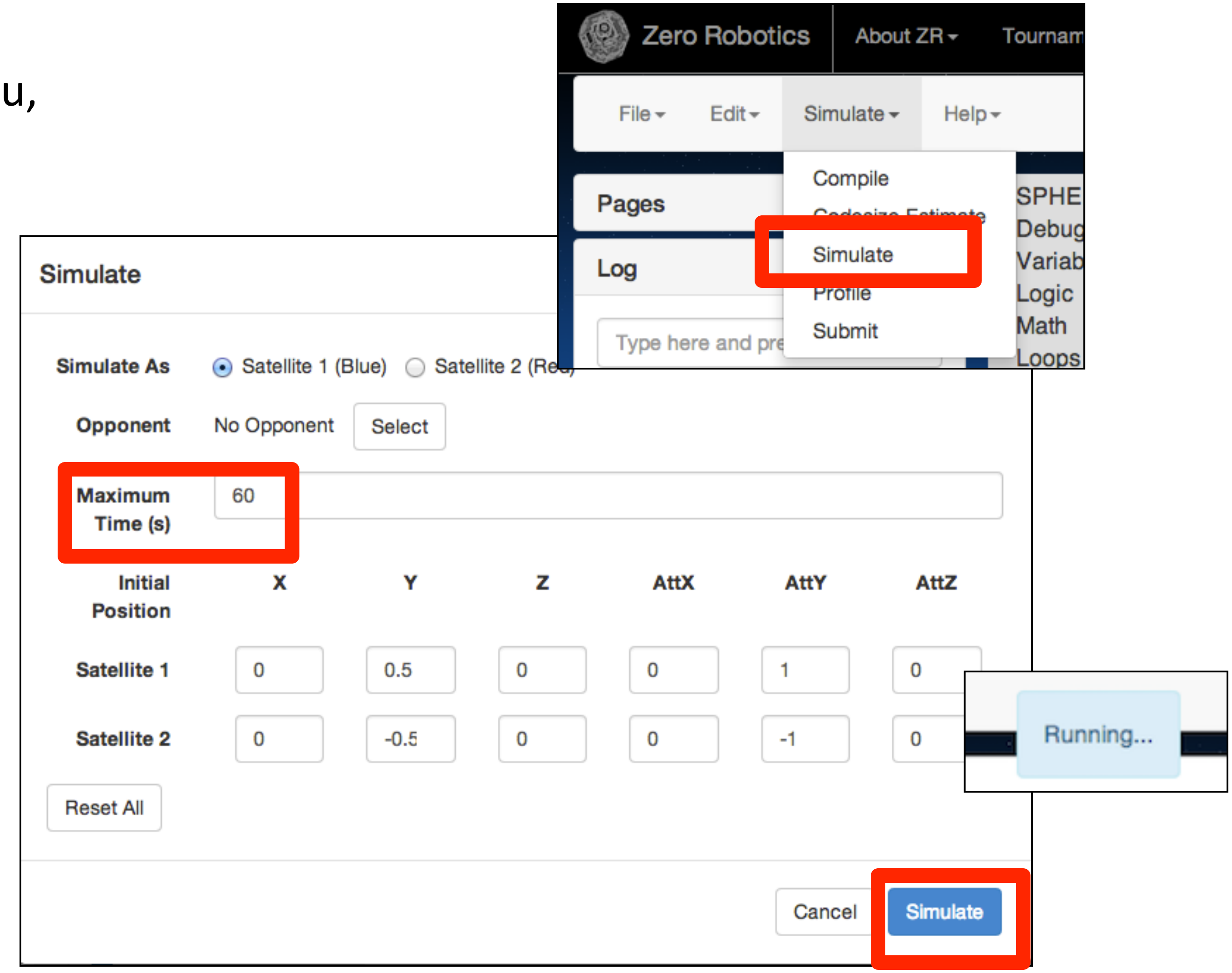




```
Your Name    May 3, 2014
9:28:38 AM
Compilation succeeded. 3%
codesize usage.
```

Simulate

- Click on “Simulate” (top menu, 3rd item from left)
- The Simulation window will open
- Change “Maximum Time” setting to 60
- Click “Simulate”
- a “Running” window pop up while the simulation is being constructed



Simulate

Simulate As ● Satellite 1 (Blue) ○ Satellite 2 (Red)

Opponent No Opponent Select

Maximum Time (s) 60

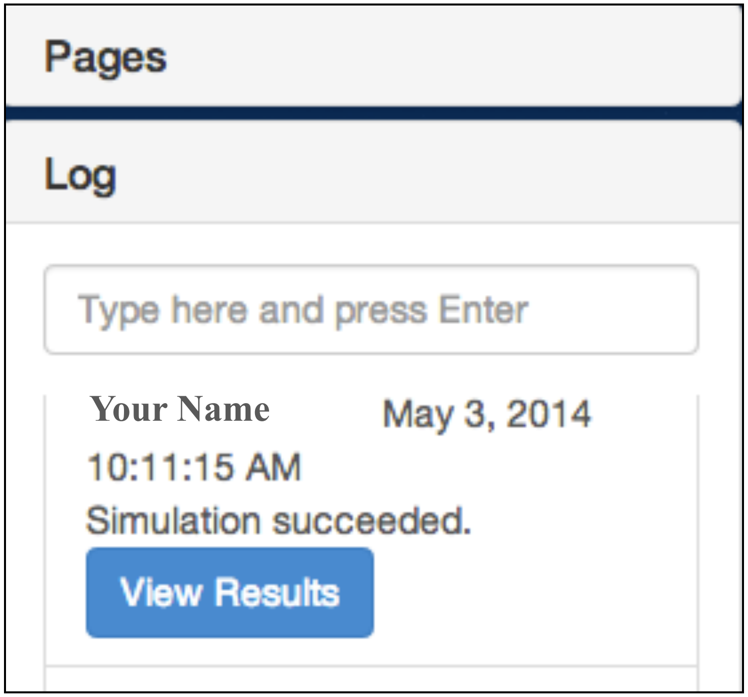
Initial Position	X	Y	Z	AttX	AttY	AttZ
Satellite 1	0	0.5	0	0	1	0
Satellite 2	0	-0.5	0	0	-1	0

Reset All

Cancel Simulate

Simulate (cont.)

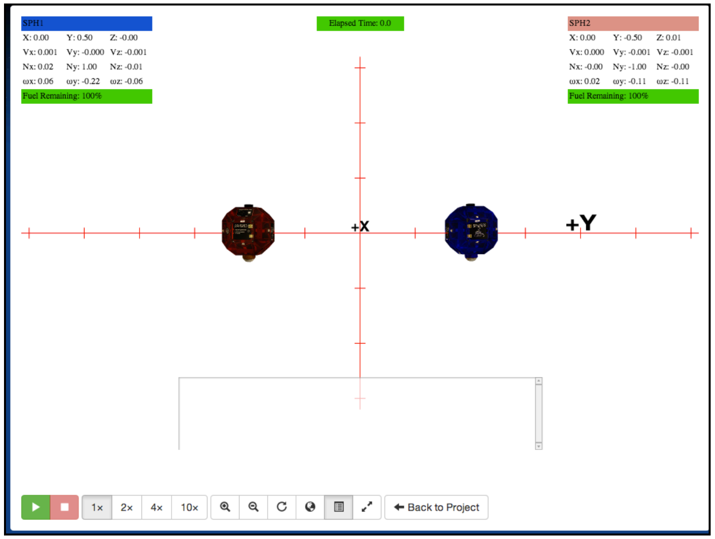
- When complete:
  - The log will open with a simulation succeeded or failed message.
  - Click on "View Results"



Your Name May 3, 2014
10:11:15 AM
Simulation succeeded.

View Simulation

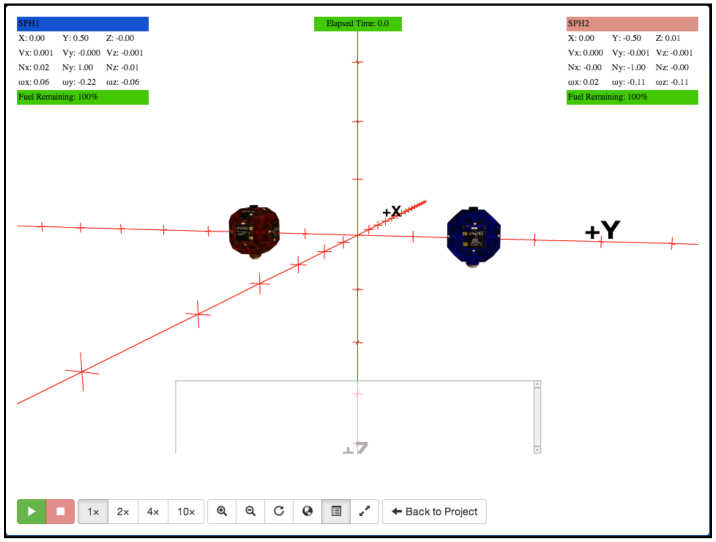
- Watch the simulation
  - Satellites start at y=0.5 and y=-0.5
  - Hash marks are 0.25 units apart
  - The blue satellite will start moving to the location you programmed earlier!
- The initial view shows y and z axis
  - horizontal line (the y-axis)
  - vertical line (the z-axis)
- To see the x axis:
  - Click and hold the left mouse button anywhere on the background and move the mouse until x, y and z axis are visible



SPH1
X: 0.00 Y: 0.50 Z: -0.00
Vx: 0.001 Vy: -0.000 Vz: -0.001
Nx: 0.02 Ny: 1.00 Nz: -0.01
ωx: 0.06 ωy: -0.22 ωz: -0.06
Fuel Remaining: 100%

Elapsed Time: 0.0

SPH2
X: 0.00 Y: -0.50 Z: 0.01
Vx: 0.000 Vy: -0.001 Vz: -0.001
Nx: -0.00 Ny: -1.00 Nz: -0.00
ωx: 0.02 ωy: -0.11 ωz: -0.11
Fuel Remaining: 100%



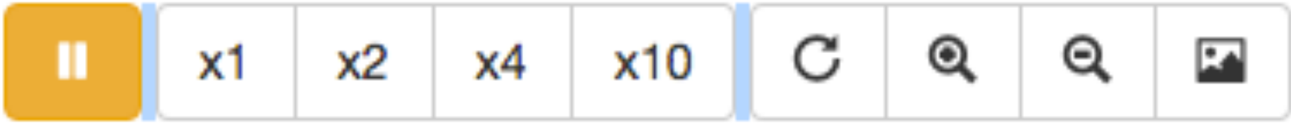
SPH1
X: 0.00 Y: 0.50 Z: -0.00
Vx: 0.001 Vy: -0.000 Vz: -0.001
Nx: 0.02 Ny: 1.00 Nz: -0.01
ωx: 0.06 ωy: -0.22 ωz: -0.06
Fuel Remaining: 100%

Elapsed Time: 0.0

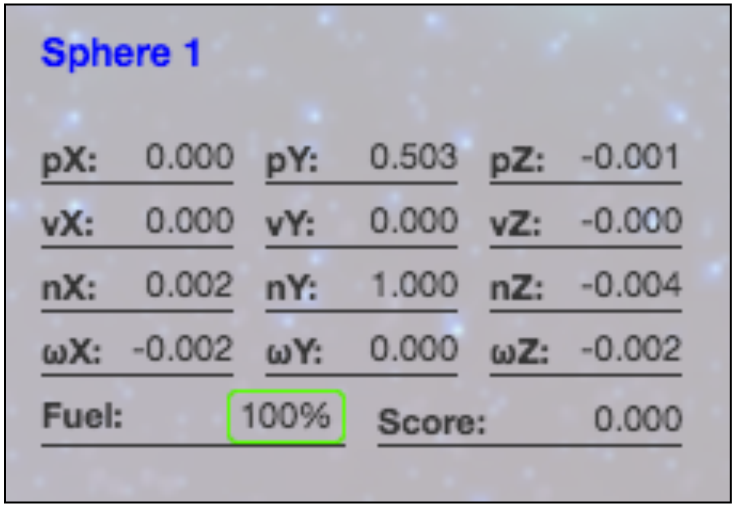
SPH2
X: 0.00 Y: -0.50 Z: 0.01
Vx: 0.000 Vy: -0.001 Vz: -0.001
Nx: -0.00 Ny: -1.00 Nz: -0.00
ωx: 0.02 ωy: -0.11 ωz: -0.11
Fuel Remaining: 100%

View Simulation (cont.)

- Replay the simulation by clicking the refresh button.



- Experiment with your views by clicking on and moving the screen
- Watch the scoring box (top-left corner of the screen with blue label) which provides information about the blue SPHERES satellite:
  - where the satellite is (pX, pY and pZ)
  - how fast it's moving (Vx, Vy, Vz)
  - We'll explain the other labels later (they tell you which way the satellite is pointing and how fast it's rotating).

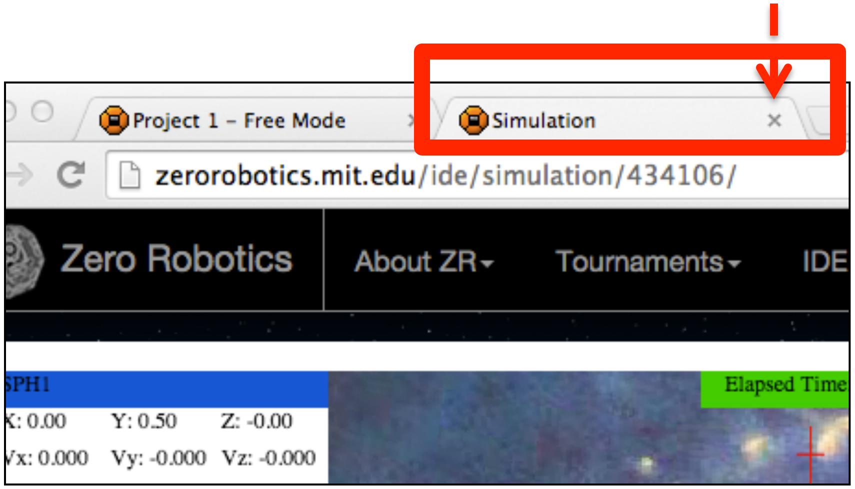


Sphere 1

pX:	0.000	pY:	0.503	pZ:	-0.001
vX:	0.000	vY:	0.000	vZ:	-0.000
nX:	0.002	nY:	1.000	nZ:	-0.004
ωX:	-0.002	ωY:	0.000	ωZ:	-0.002
Fuel:	100%	Score:	0.000		

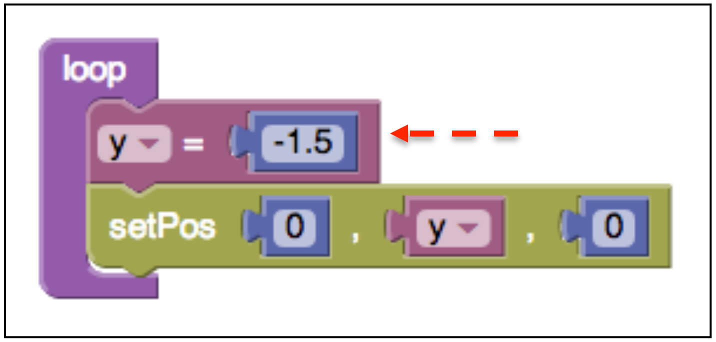
Modify Program, Compile & Simulate

- Return to the Graphical Editor page by closing the simulation window (top menu bar)



X: 0.00 Y: 0.50 Z: -0.00
Vx: 0.000 Vy: -0.000 Vz: -0.000

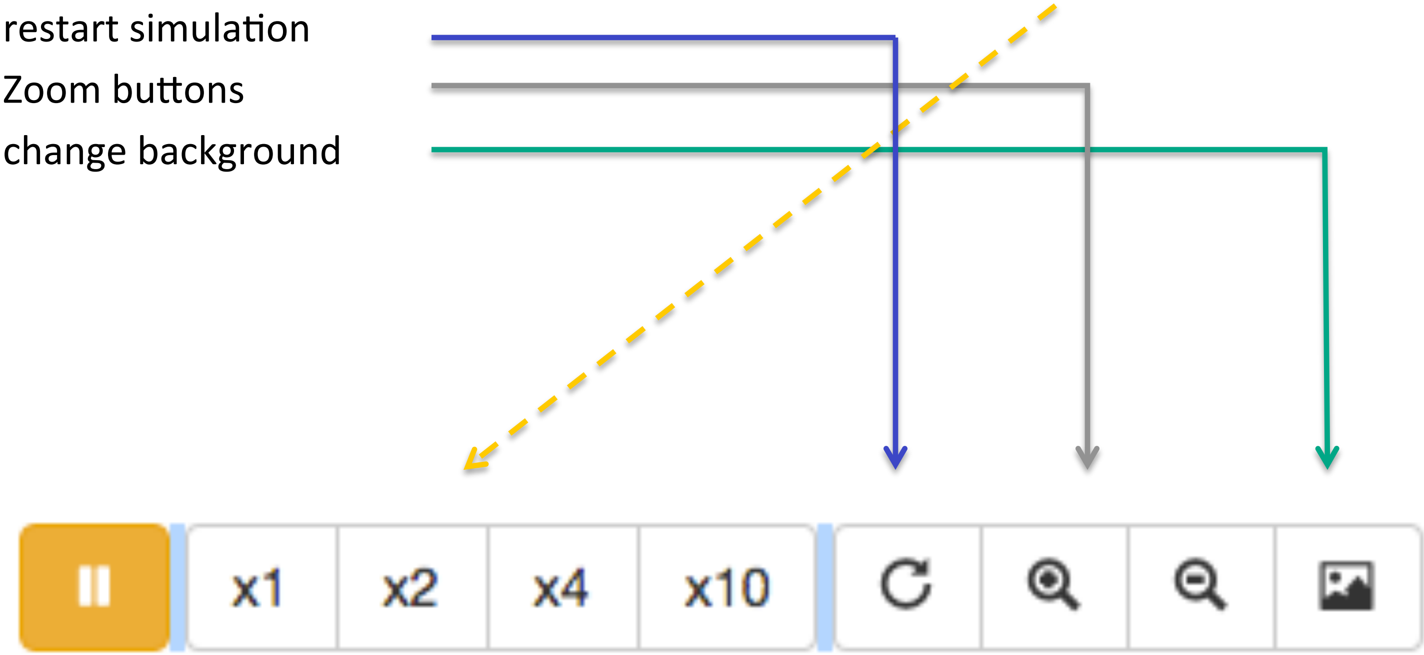
- Make the following changes to program the satellite to move 2.0 meters in the other direction along the y-axis this time:
  - Change “Y= 1.5” to “Y= -1.5”
- “Compile” and “Simulate” as before
- “View Results”
  - This time the blue SPHERES satellite should move in the opposite direction along the “y” axis



```
loop
  y = -1.5
  setPos 0 , y , 0
```

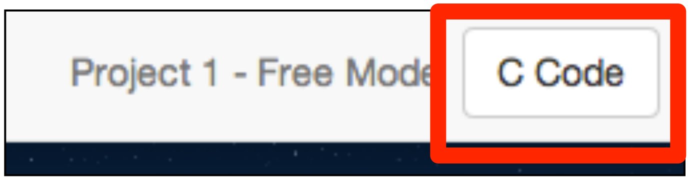
Modify Program, Compile & Simulate (cont.)

- Play!
  - Now it’s your turn to experiment with changing the “Y” values and running the simulation to see what happens
  - Experiment with the simulation buttons and views at the bottom to:
    - change simulation speeds (see the “1x, 2x, 3x, 4x, 10x” buttons)
    - restart simulation
    - Zoom buttons
    - change background



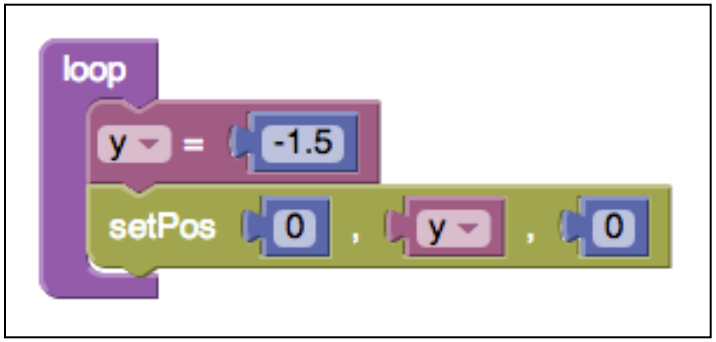
View program in C Code

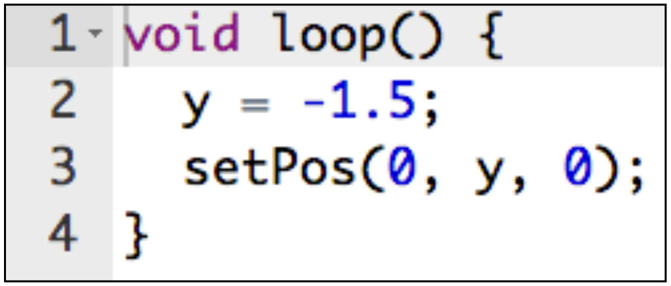
- Close the simulation window to return to the Graphical Editor page
- On the right side of the top menu bar select “**C Code**”



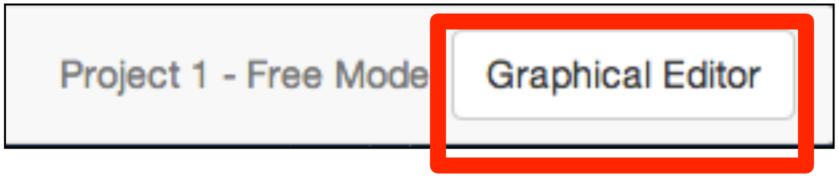
- Compare:

Your program - versus - C Code





```
void loop() {
  y = -1.5;
  setPos(0, y, 0);
}
```



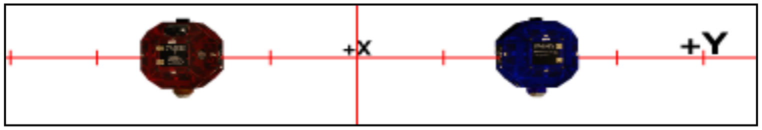
- Select Graphical Editor to return to the graphical editor

Review

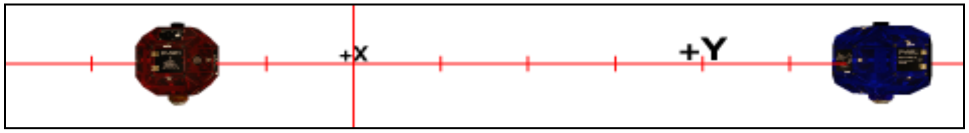
- Congratulations!
- You have successfully created and run a program in the ZR IDE
- You created the variable Y and used it to set the position of the SPHERES

- Y = 1.5 moved the Blue SPHERES satellite like this:

Start:

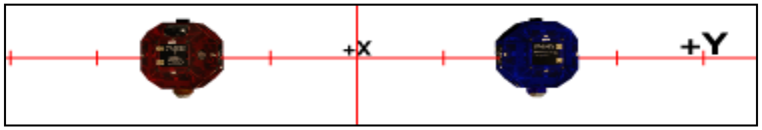


End:



- Y = -1.5 moved the Blue SPHERES satellite like this:

Start:



End: