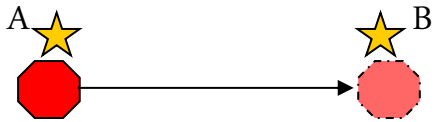

# Getting to Know the ZR IDE

# 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

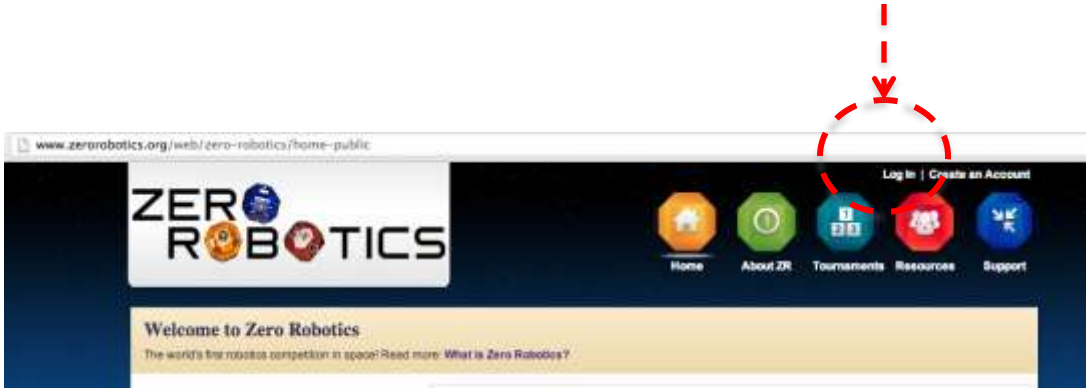


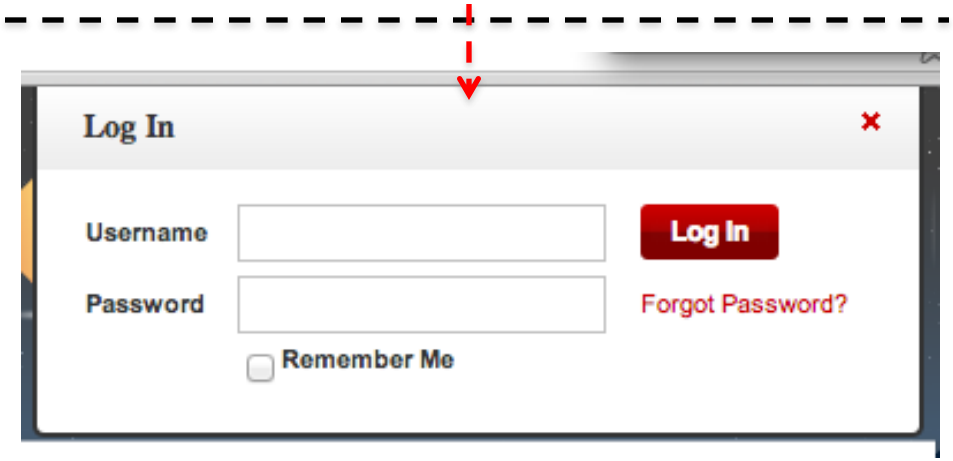

- Compile your code (check it for errors)
- Simulate (run the code in a simulation)

# Login

- Go to the Zero Robotics Website:
  https://www.zerorobotics.org

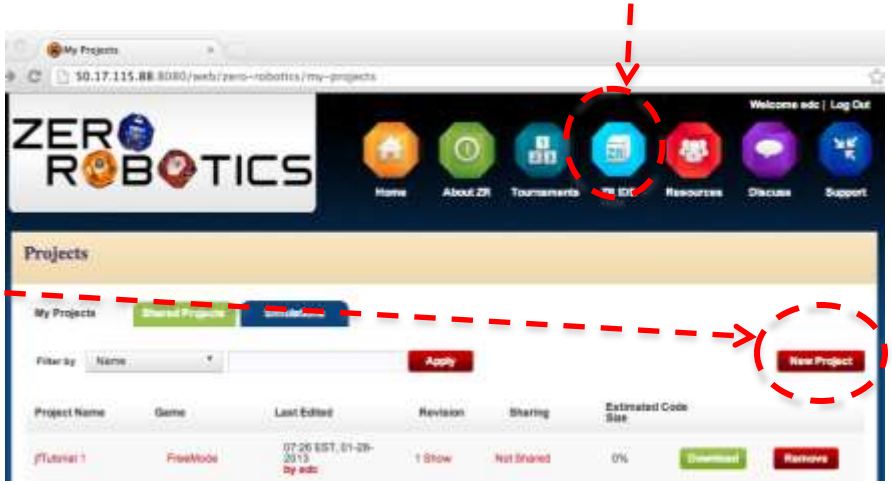
- Log into your account

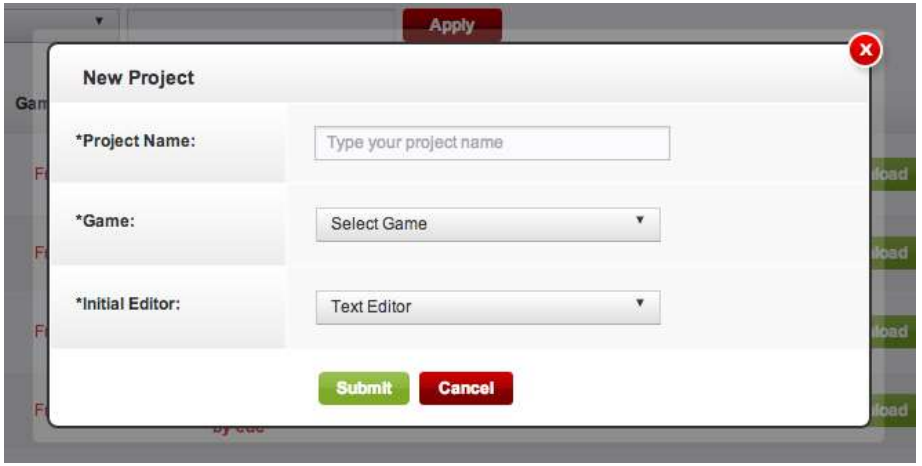


# Create a New Project

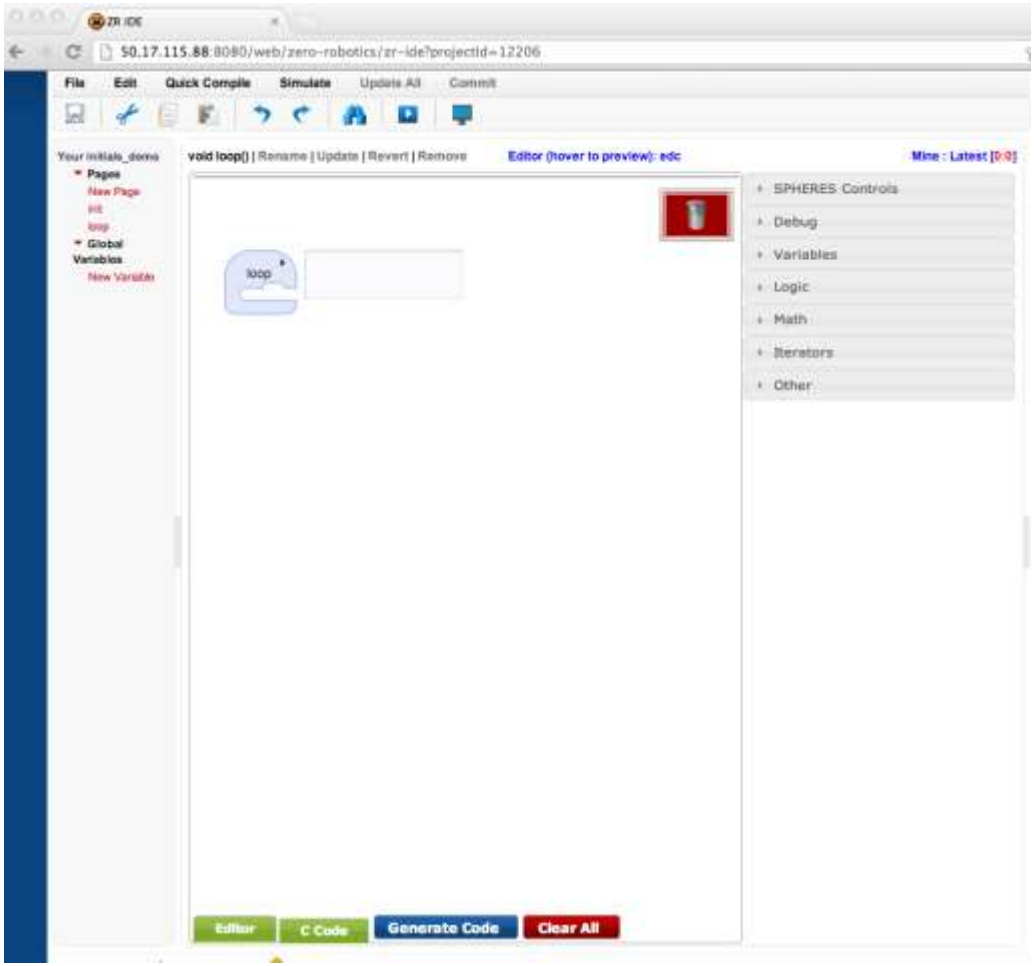
- Select light blue "ZR IDE" SPHERES icon on top ribbon
- Select " New Project"
- Enter
  - Project Name
    - Example: Project 1
  - Game
    - Select "FreeMode"
  - Initial Editor
    - Select "Graphical Editor"
- Click "Submit"



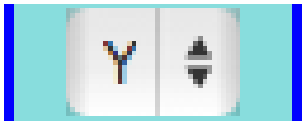
# Graphical Editor IDE

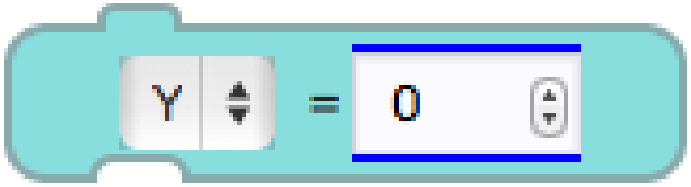
- The Graphical Editor version of the ZR IDE is shown here
- On the next pages, you will:
  - Review what you know about variables
  - Create a new variable



# 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

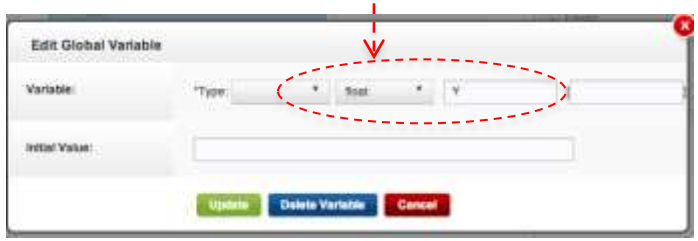
- Declare a variable (called “Y”) to set the position of the SPHERES satellite

1. Click on “New Variable” (below “Global Variables” menu)

1. In the “New Global Variable” box
   - Click on “char” and change to “float”
   - In text box to the right of “float”, type “Y”

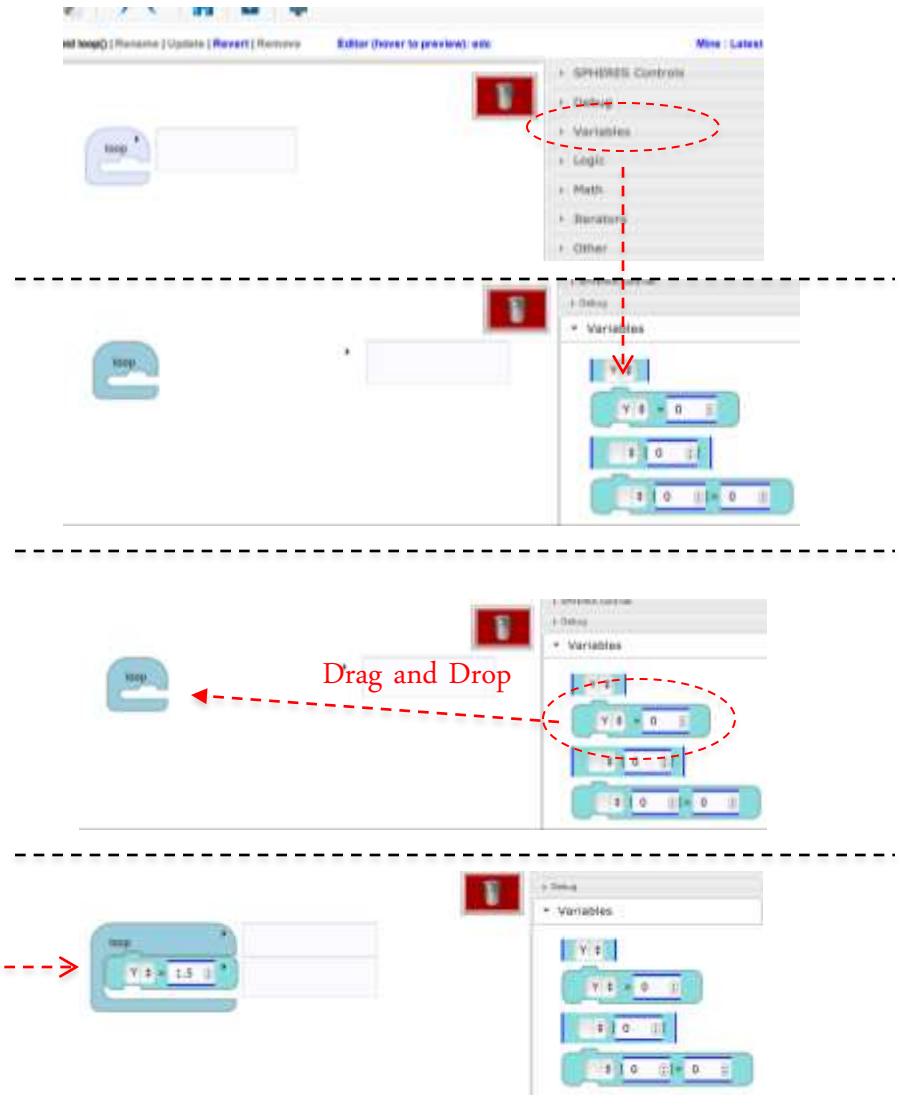
3. Click on “Create” (green button at the bottom of the box)

4. You should now see the new variable “Y” below the “New Variable” link


# Assign a Value to Your Variable

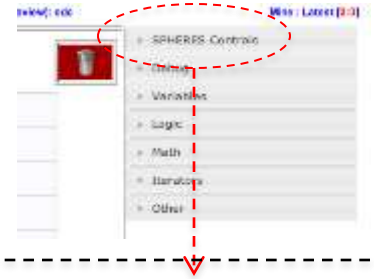
- Now that we've created the variable Y, we need to actually put a number in it
  1. Click on "Variables" accordion in the Editing Control Panel to open the toolbox
  2. Drag the "Y= 0" block into the "loop" icon on the main screen
  3. The block should now appear within the "loop"
  4. Delete 0 from the 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. Select the "setPos 0,0,0" (set position) block and, with the mouse button held down, drag the block and drop it inside the "loop" below the Y = 1.5 block.
  3. Click on the "Variables" accordion
  4. Select the Y block( Y ) and drag and drop the block over the middle zero (0) in the "setPos" block.
- This program you've created tells the satellite to move to the position on the Y axis defined by the variable "Y"

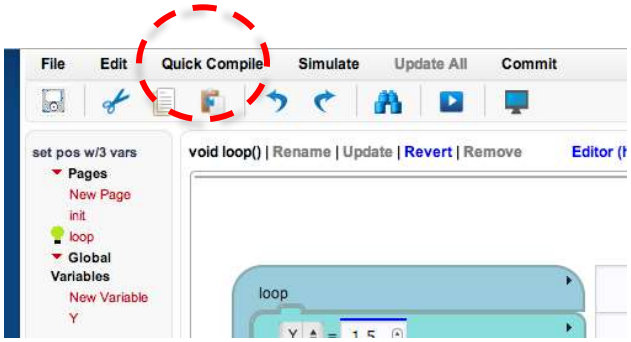
  -The satellite will move along the y-axis and stop at the y=1.5 position





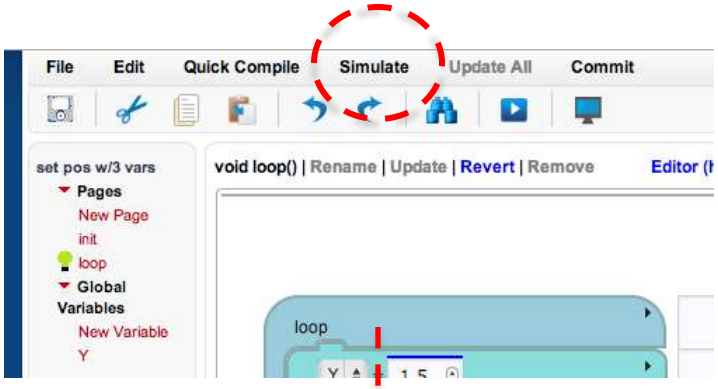
# Quick Compile

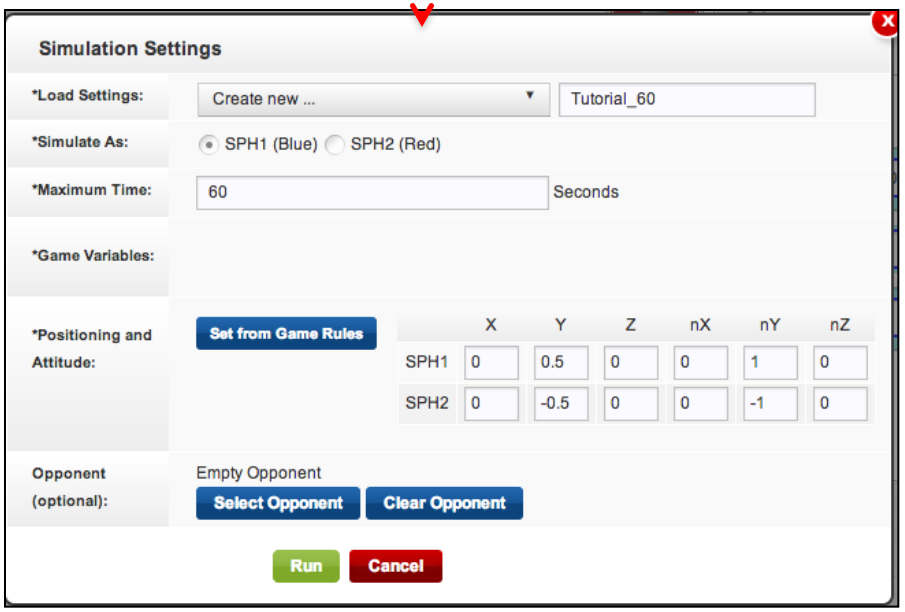
- Now let's see your program in action!
- Quick compile:
  - Click on "Quick Compile" (top menu, third from the left)
  - On the pull down menu, click on "Compile with code size estimate"
- After compiling:
  - At the bottom of the screen, you should see "0 Warnings" and "0 Errors"




# Simulate

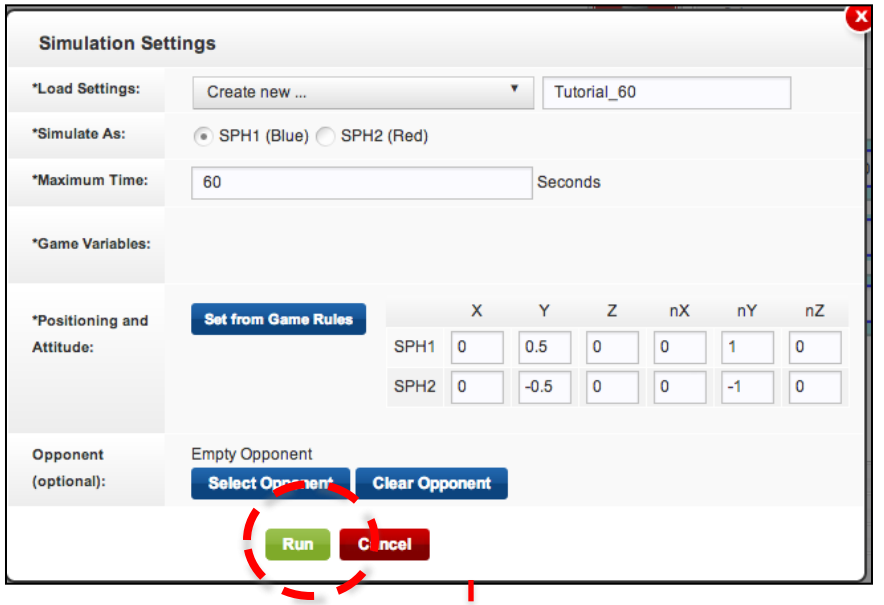
- Click on “Simulate” (top menu, 4th item from left)
- In the Simulation Settings pop-up box:
  - *Load Settings:
    - Select “Create new…”,
    - Type a settings name: “Tutorials_60”
    - (this simulation will run for 60 seconds)
  - *“Simulate As”:
    - Select “SPH1 (Blue)”
  - * “Maximum Time”:
    - Change from 90 seconds to 60 seconds
  - *Positioning and Attitude
    - Click “Set from Game Rules”
    - Leave the text fields alone
  - *Opponent:
    - Should be “Empty Opponent” (select “Clear Opponent” otherwise)



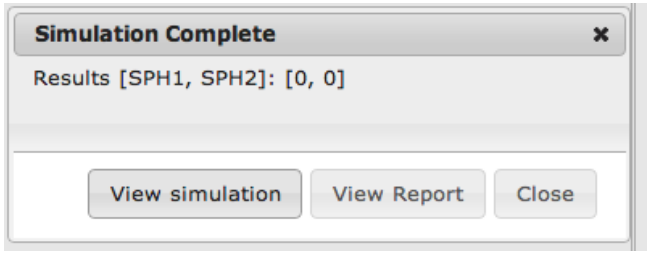



# Simulate (cont.)

- Click on green “Run” button at the bottom
  - This will take a minute
  - You may see messages while you wait

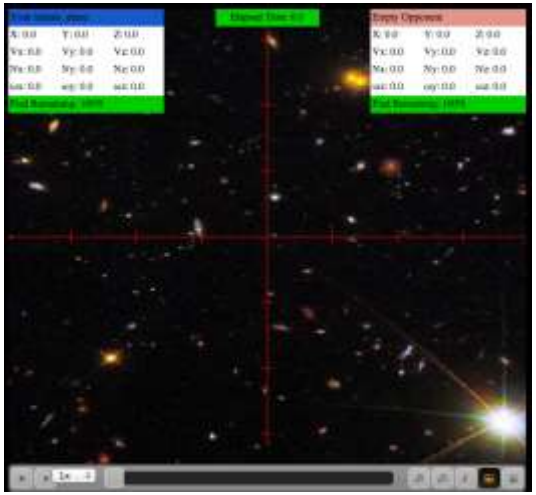



- Click on “View simulation”
- A new browser window should pop up with background picture.



# View Simulation

- 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



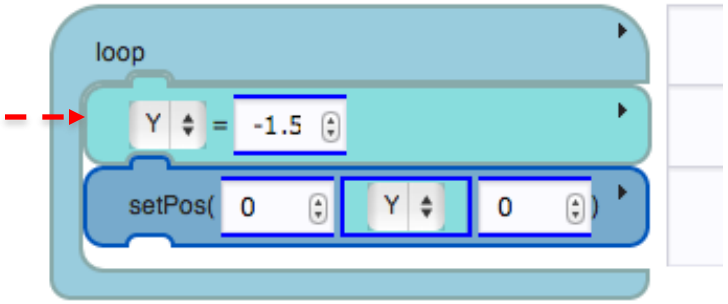
- Click the “Play” arrow at the bottom left of the screen and wait a few seconds.
  - Two SPHERES satellites will appear
    - Satellites start from y=0.5 and y=-0.5
    - Hash marks are 0.25 units apart
  - The blue satellite will start moving to the location you programmed earlier!

# View Simulation (cont.)

- Replay the simulation by clicking the play arrow again
- 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 (X, Y and Z)
  - 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).

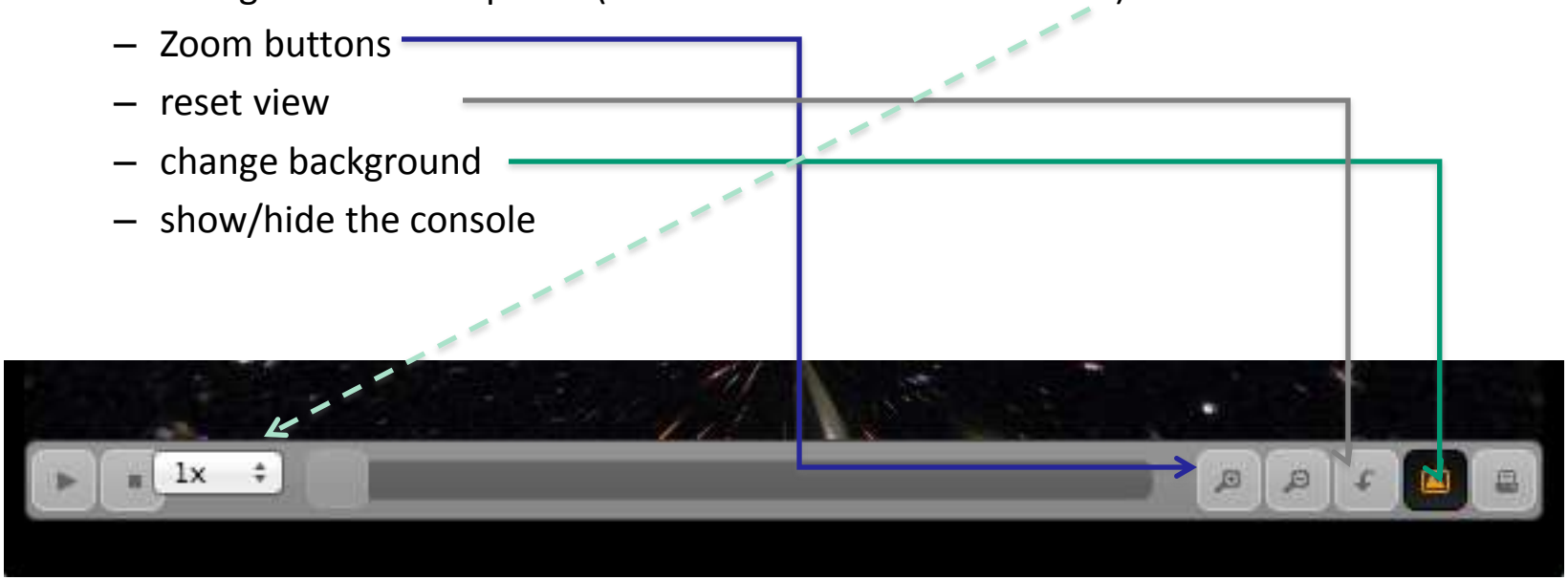
# Modify Program, Compile & Simulate

- Close the Simulation Window
- Return to the Graphical Editor page
- 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”
- “Quick Compile” and “Simulate” as before
- “Run”
  - This time the blue SPHERES satellite should move in the opposite direction along the “y” axis

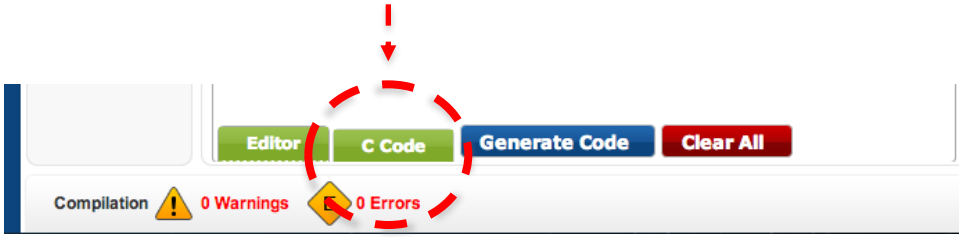


# Modify Program, Compile & Simulate (cont.)

- 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" button on the left)
    - Zoom buttons
    - reset view
    - change background
    - show/hide the console

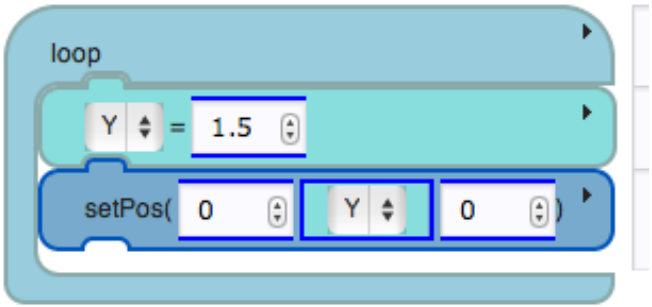


# View program in C Code



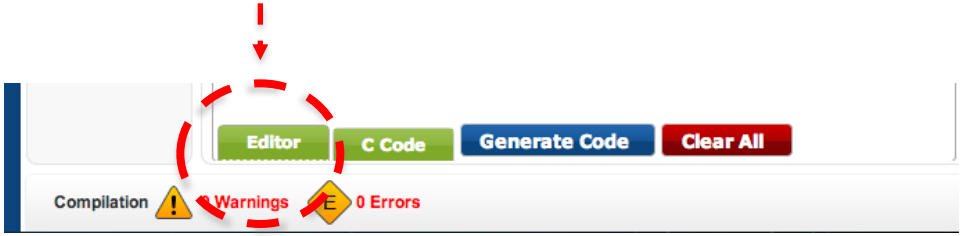

- Close the simulation window
- At the bottom of the Graphical Editor window select “C Code”
- Compare:

Your program - versus - C Code




```
1  void loop()
2  {
3    Y = 1.5;
4    setPos(0, Y, 0);
5    return;
6  }
7
8
```

- To return to the graphical editor, select “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: