ZERO ROBOTICS

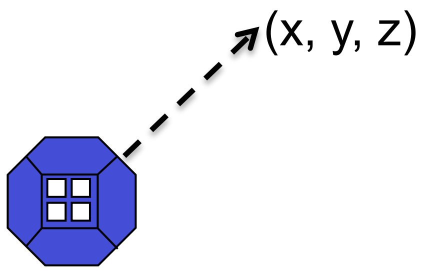
ISS PROGRAMING CHALLENGE

Variables, Arrays, and the setPositionTarget Function

Goals

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

- Create a new project
- Create a new variable
- Create an array
- Learn about a SPHERES control and *setPositionTarget* function
- Compile your code (check it for errors)
- Simulate (run the code in a simulation)

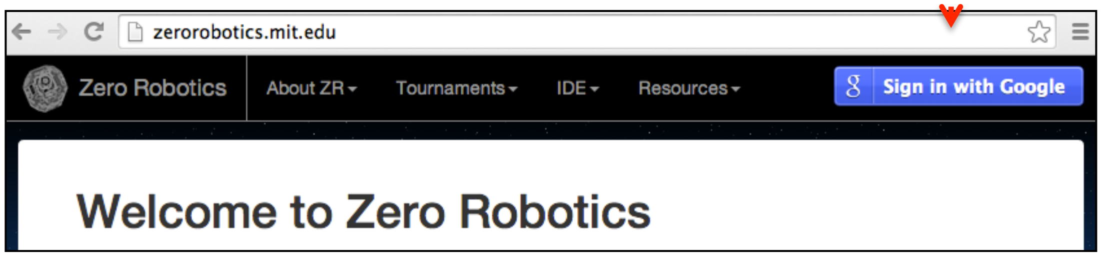


ZERO ROBOTICS
ISS PROGRAMING CHALLENGE

Log In

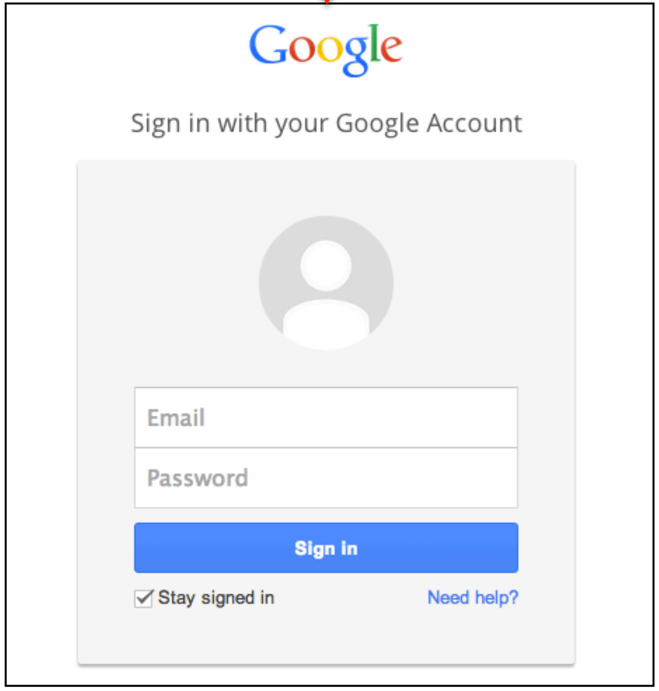
- Go to the Zero Robotics website:

  www.zerorobotics.mit.edu



Welcome to Zero Robotics

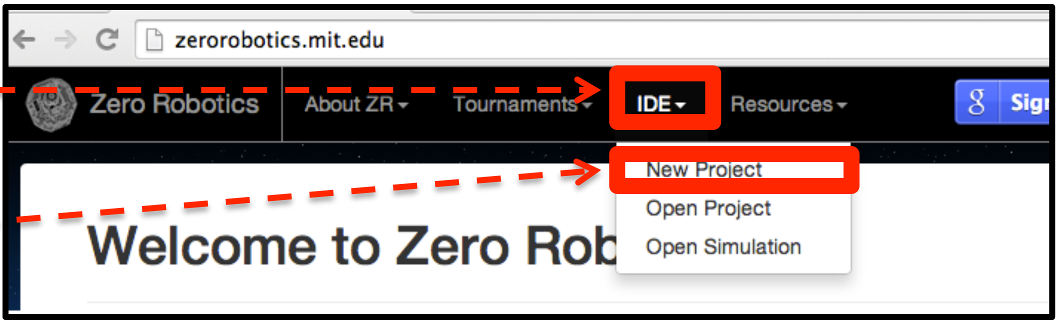
- Log into your account with your email and password

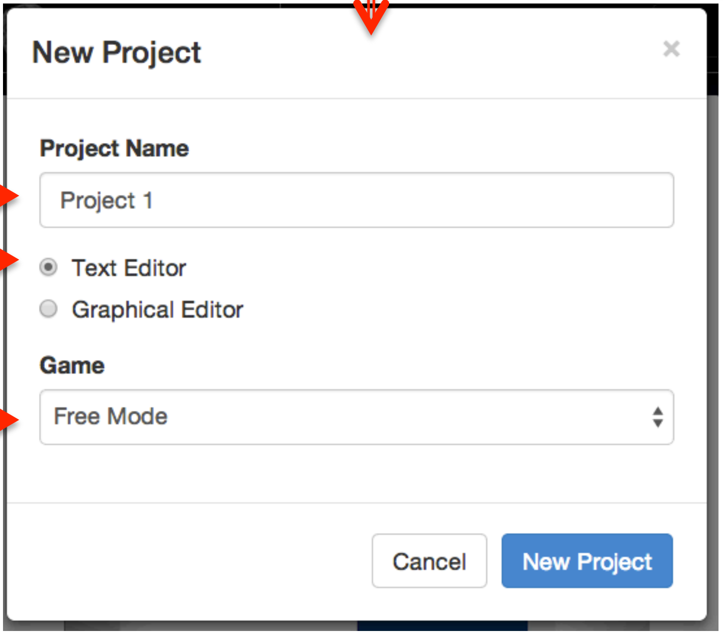


Sign in with your Google Account

Create a New Project

- Select light blue "ZR IDE" SPHERES icon on top ribbon
- Select " New Project"
- Enter
  - Project Name
    - Type: Project 1
  - Select "Text Editor"
  - Select "FreeMode"
- Click "New Project"




New Project

Project Name

Project 1

● Text Editor

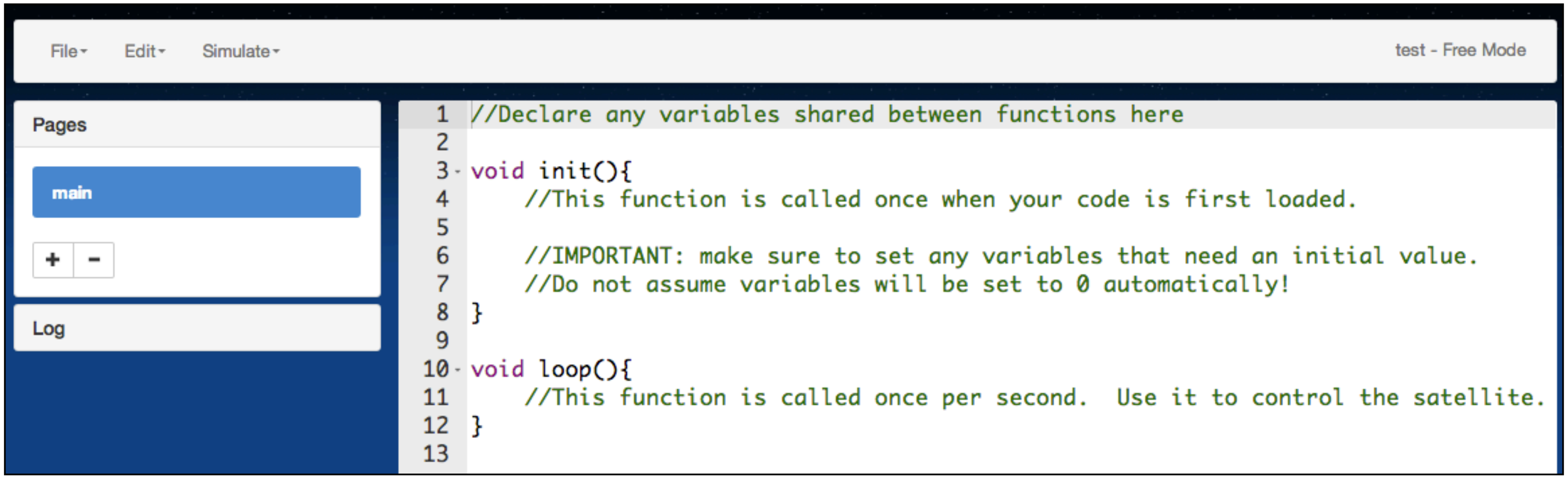
○ Graphical Editor

Game

Free Mode

Text Editor IDE

- The Text Editor version of the ZR IDE is shown here



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

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!
}

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

- 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 (for example, a number)
  - The **name** of the variable — like a label on the container so you can find it and use it (for example, Y)
- This is called **declaring** the variable.

```
Y

Y=0;
```

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 greater precision
    - Floats should end with f to show that they are single-precision float values (the type used by SPHERES)
- 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 have different types
  - C++ is **case-sensitive**, so capitalization matters: myVariable is not the same as MYvAriabLe
- Do not give a variable a name that 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 %)

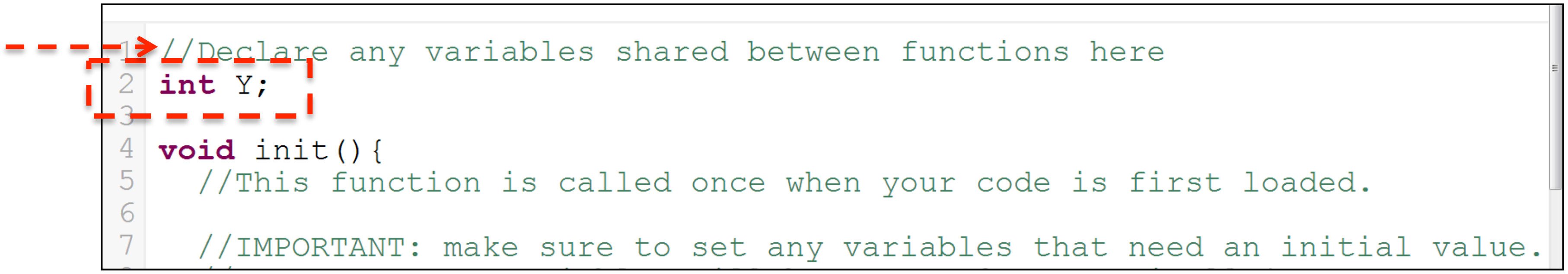
ZERO ROBOTICS ISS PROGRAMING CHALLENGE

Declaring a New Variable

- Declare a variable (called "Y") to set the position of the SPHERES satellite
  1. Declare what **type** of variable the variable will be. The two types you know so far are **int** and **float**.
  2. Put a space and then type the **name** of the variable, followed by a semicolon.

  Format: **type name;** For example: **int Y;**

- Put the declaration statement at the beginning of your code, where the template says to declare variables shared between functions (don't worry about functions yet)

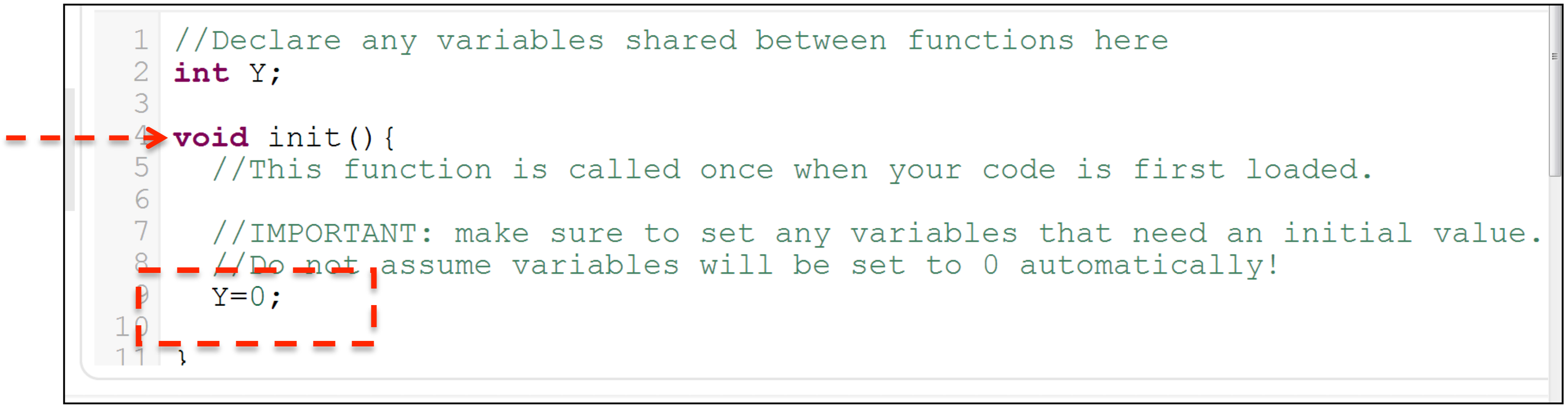


```
//Declare any variables shared between functions here
int Y;

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.
```

Assign a Value to Your Variable

Now that we've created the variable Y, we need to actually put a value into it.

- Assign a value in the section **void init()** between the curly brackets **{}**. The code here will run once at the start of the game.
- Type in the variable name (without the type), followed by an equals sign.
- Type the desired value and end the line with a semicolon.
  For example: **Y=0;**
- C++ ignores whitespace (spaces, tabs, and line breaks) between "words." Note that the picture shows this line indented by several spaces. Using whitespace can make the code easier to read but has no effect on its function.



```
//Declare any variables shared between functions here
int Y;

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!
  Y=0;

}
```

Arrays

- An array is a list of data of the same type.

  - Examples:

    {1.2, 3.0, -2.5}

    This is an array of 3 floats (could be an x, y, z coordinate point)

    {99, 95, 82, 90, 76, 91, 93, 85, 100, 65}

    This is an array of 10 integers (could be a set of test scores)

ZERO ROBOTICS
ISS PROGRAMING CHALLENGE

Declaring Arrays in C

- When you declare an array, you are actually declaring a lot of variables at once.
- An array is declared by assigning:
  - variable **type** (e.g. int)
  - the array's **name** (e.g. myArray)
  - **number of variables** (e.g. 4) in [square brackets]
  - End the line with a semicolon
- The variables in the array are named with the name of the array plus a number in square brackets. **The numbering starts with 0.**
- For example, in the array at the right the members are four int variables called myArray[0], myArray[1], myArray[2], and myArray[3]. The value of each variable is different.

Array declaration:

```
int myArray[4];
```

Member variables:

```
myArray[0] = 5
myArray[1] = -7
myArray[2] = 18
myArray[3] = 100
```

Array Declaration Question

Suppose the numbers below represent the test scores in a class.

{99, 95, 82, 90, 76, 91, 93, 85, 100, 65}

How would you declare an array to hold the scores? (What are the three things you must include in the declaration?)

What would be the names and values of the variables in the array?

ZERO ROBOTICS
ISS PROGRAMING CHALLENGE

Answer

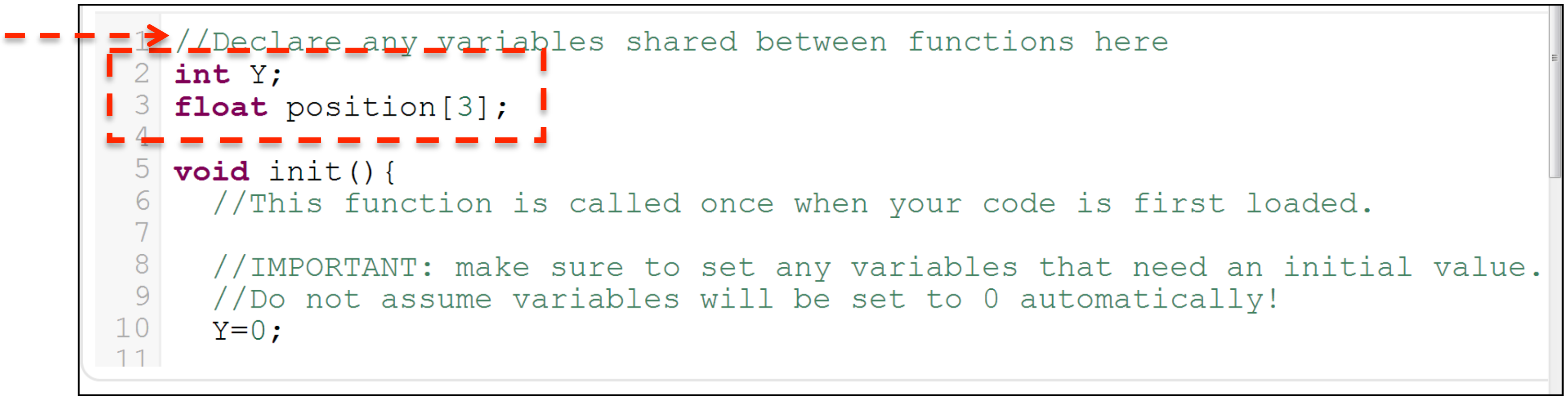
The data consists of **ten integers**, so we want ten int variables.

Declaration:	**int classScores[10];**
Members:	classScores[0] = 99
	classScores[1] = 95
	classScores[2] = 82
	classScores[3] = 90
	classScores[4] = 76
	classScores[5] = 91
	classScores[6] = 93
	classScores[7] = 85
	classScores[8] = 100
	classScores[9] = 65

Declare a New Array

- Start your program by declaring an array to hold x, y, z position coordinates for the SPHERES satellite
- Go to the space above void init() and void loop(), right below the integer Y that we declared earlier. Remember, any new variables and arrays we want to declare will be declared in this area.
- The array consists of three floating-point numbers, and we want to name it position. Insert the declaration line:

float position[3];



```
//Declare any variables shared between functions here
int Y;
float position[3];

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!
  Y=0;
```

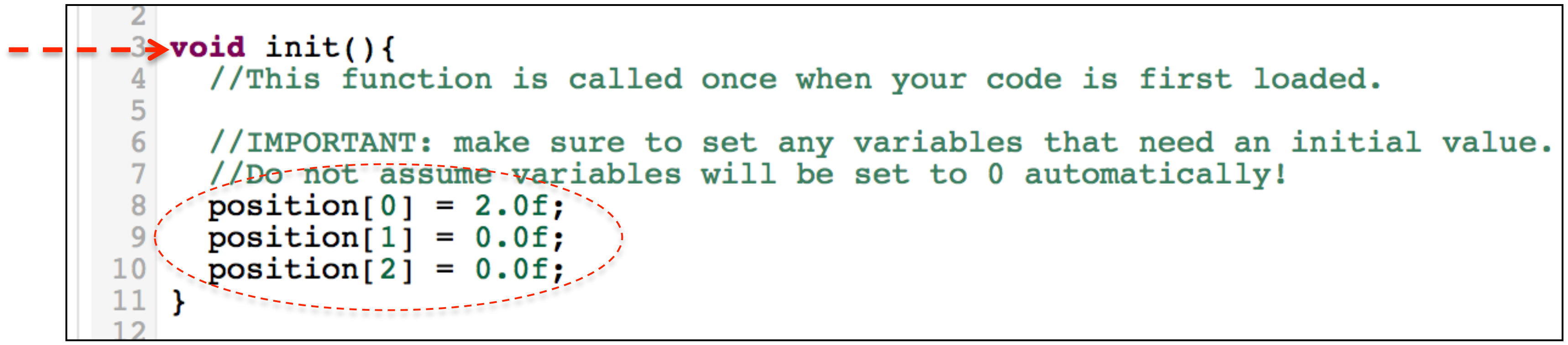
Assign Values to Your Array

- Now we assign values to the array in the void init() area, just as before.
- You must assign a value to each variable in the array. The name of each array member is the name of the array plus the member's number in square brackets.
- Let's assign the values 2.0 to the first element (x coordinate), 0.0 to the second (y), and 0.0 to the third (z.) Type:

position[0] = 2.0f;

position[1] = 0.0f;

position[2] = 0.0f;



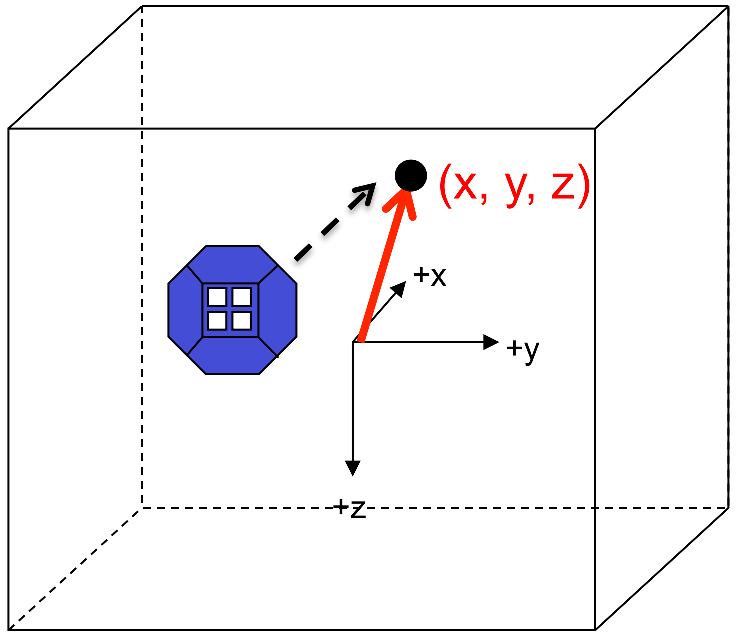
```
void init(){
   //This function is called once when your code is first loaded.

   //IMPORTANT: make sure to set any variables that need an initial value.
   //Do not assume variables will be set to 0 automatically!
   position[0] = 2.0f;
   position[1] = 0.0f;
   position[2] = 0.0f;
}
```

setPositionTarget

- The SPHERES control function **"setPositionTarget"** allows you to move the satellite to a target position

- The target point is input as an array of three floats that represent its x, y, z coordinates in meters. The first three elements of an array will correspond to the x, y, and z coordinates

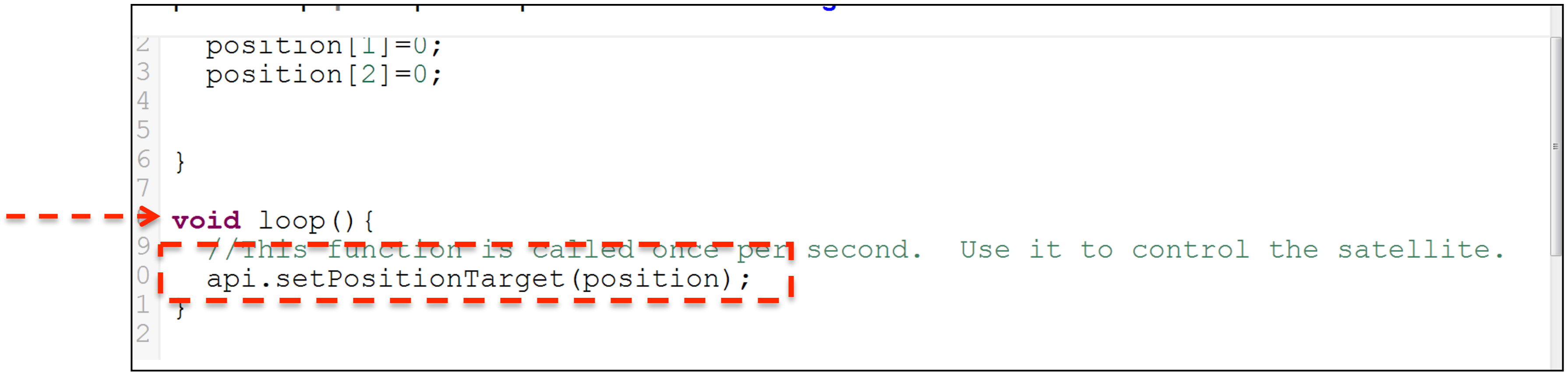
- When a position is commanded, the satellite will fire thrusters to move to the target point, then stop



Add setPositionTarget Function

- Add this command inside void loop(). This section contains the main ZR program, which runs once per second.
- Whenever you want to use one of the SPHERES controls functions, you must put **api.** before the name of the function.
- In order to tell the SPHERES which coordinates to move to, append the array name in parenthesis.
- End the line with a semicolon. The line should look like this:

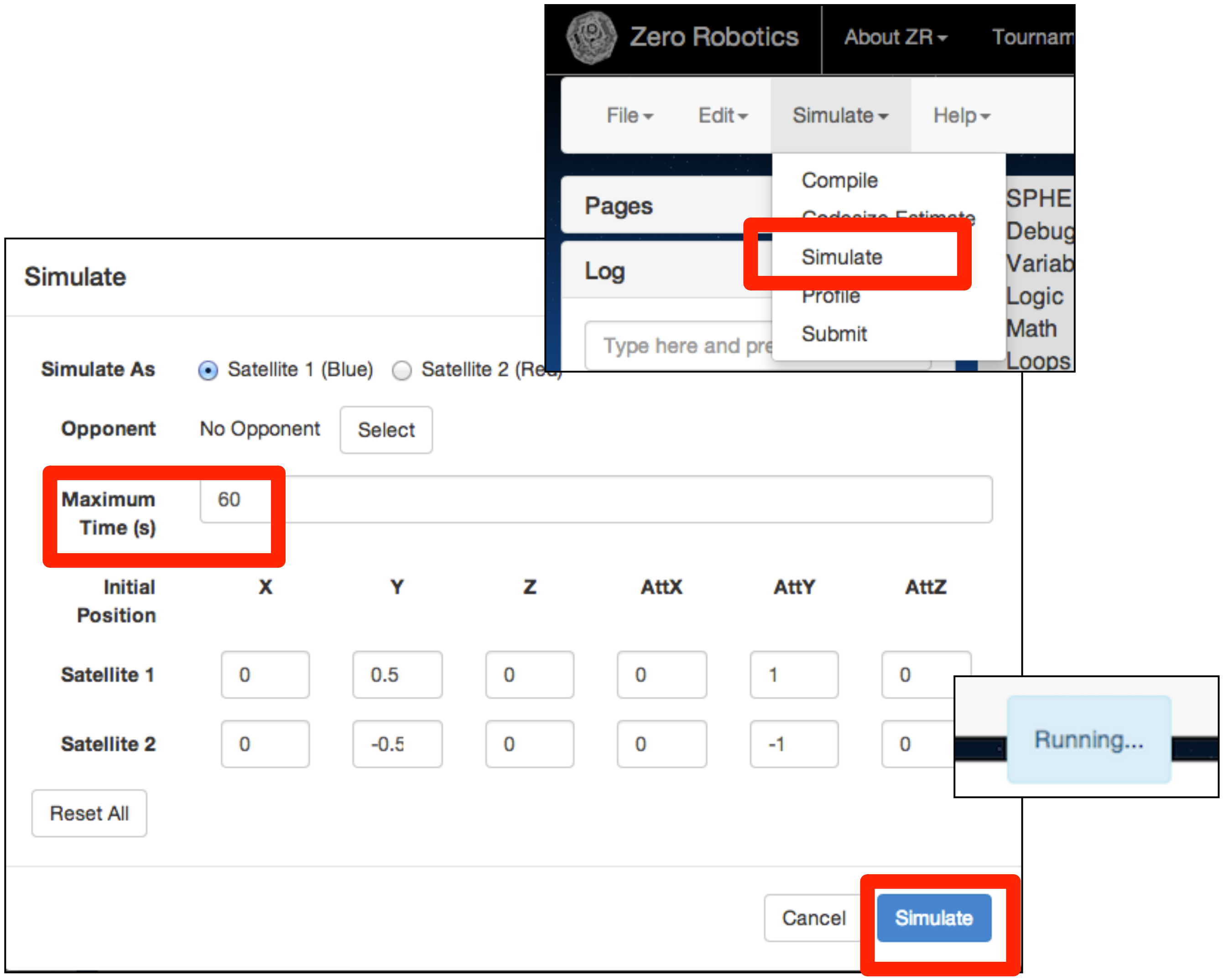
api.setPositionTarget(position);



```
  position[1]=0;
  position[2]=0;


}

void loop(){
  //This function is called once per second.  Use it to control the satellite.
  api.setPositionTarget(position);
}
```

Compile and Simulate

- Now let's see your program in action!
- Click "Simulate"
- 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

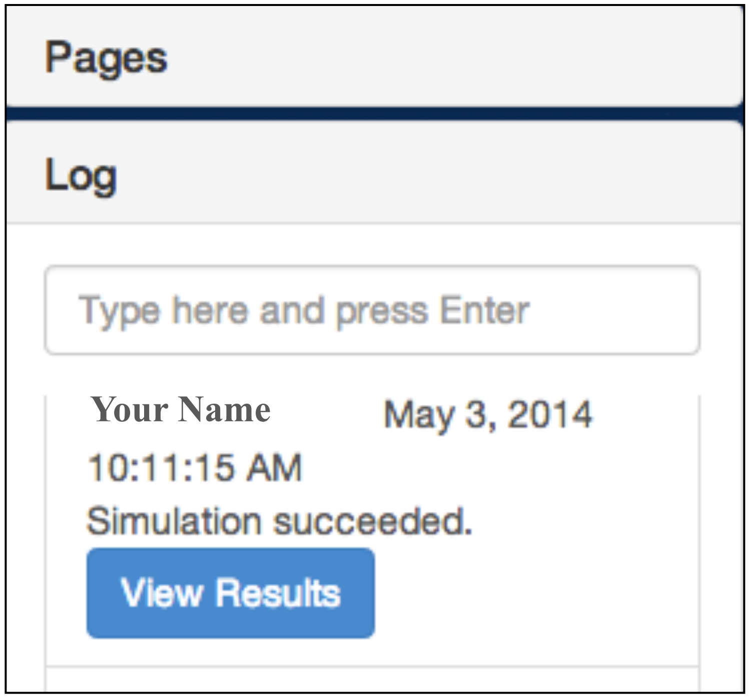
Initial Position	X	Y	Z	AttX	AttY	AttZ
Satellite 1	0	0.5	0	0	1	0
Satellite 2	0	-0.5	0	0	-1	0

Reset All

Cancel Simulate

Simulate (cont.)

- When complete:
  - The log will open with a simulation succeeded or failed message.
  - Click on “View Results”
  - Click the Play button. The satellites should appear and the blue SPHERES should move to coordinates (2.0,0.0,0.0), just as you told it to.

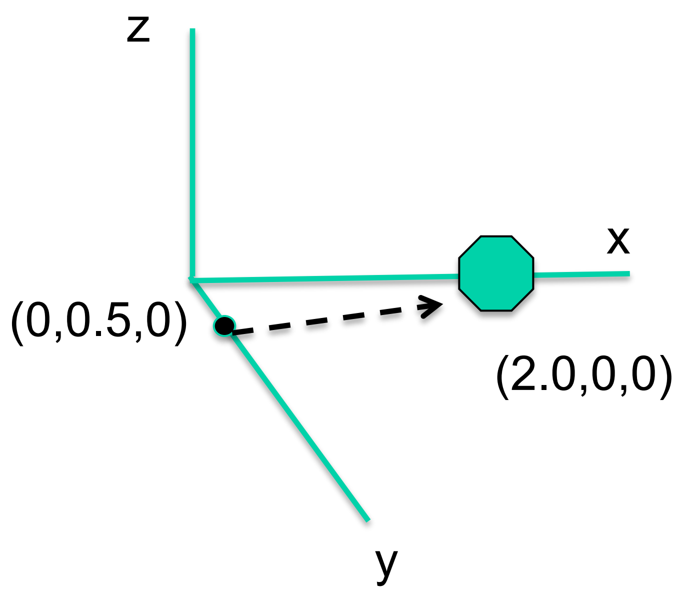


Your Name May 3, 2014
10:11:15 AM
Simulation succeeded.

Review

Congratulations!

- You have successfully created and run a program in the ZR IDE.
- You have used an array to program a SPHERES satellite.
- You programmed the satellite to move to a point in 3 dimensions!


z
x
y
(0,0.5,0)
(2.0,0,0)