

ZERO ROBOTICS

ISS PROGRAMING CHALLENGE

Getting to Know the ZR IDE (Text Editor)



Goals



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

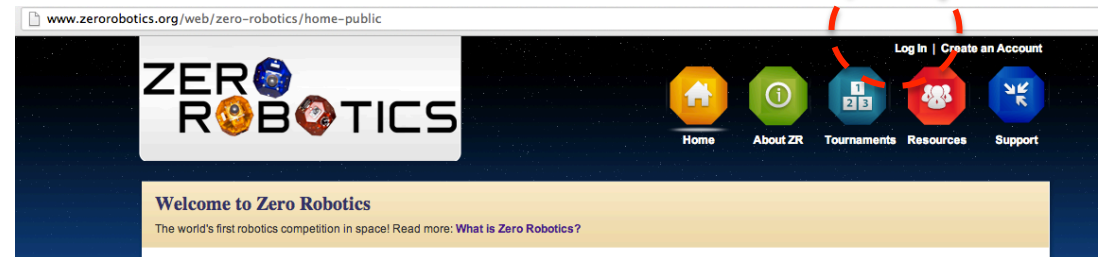
- Create a project in the C++ programming language
- Compile code (check it for errors and translate it from C++ to binary code a SPHERES can understand)
- Simulate (run the code in a simulation of the real SPHERES hardware on the ISS)

Log In



- Go to the Zero Robotics website:

<http://www.zerorobotics.mit.edu>



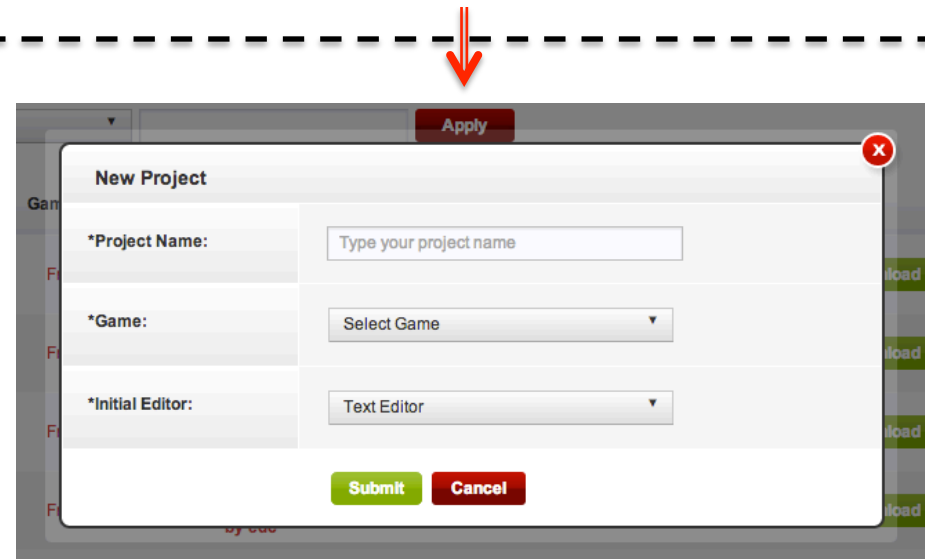
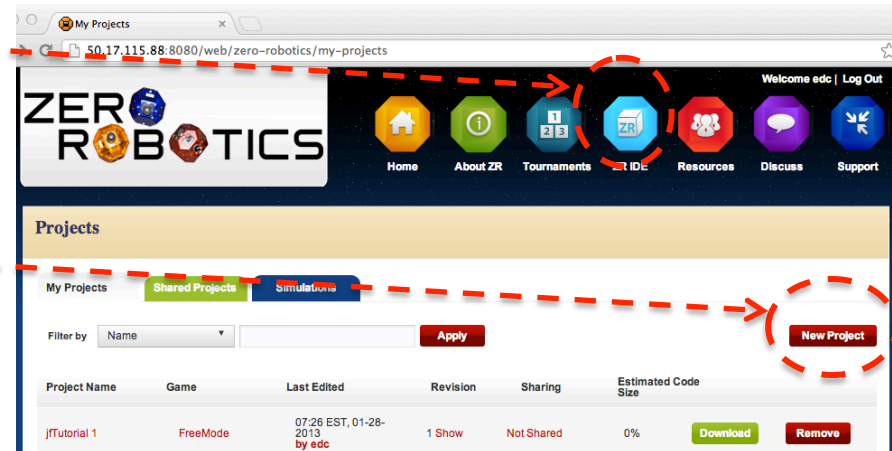
- Log in with your username and password

A screenshot of the "Log In" form on the Zero Robotics website. The form has a title "Log In" and a red "X" icon in the top right corner. It contains two input fields: "Username" and "Password". To the right of the "Password" field is a red "Log In" button. Below the "Password" field is a checkbox labeled "Remember Me". To the right of the "Log In" button is a link that says "Forgot Password?". A red dashed arrow points from the "Log In" link in the top right corner of the homepage to the "Log In" form.

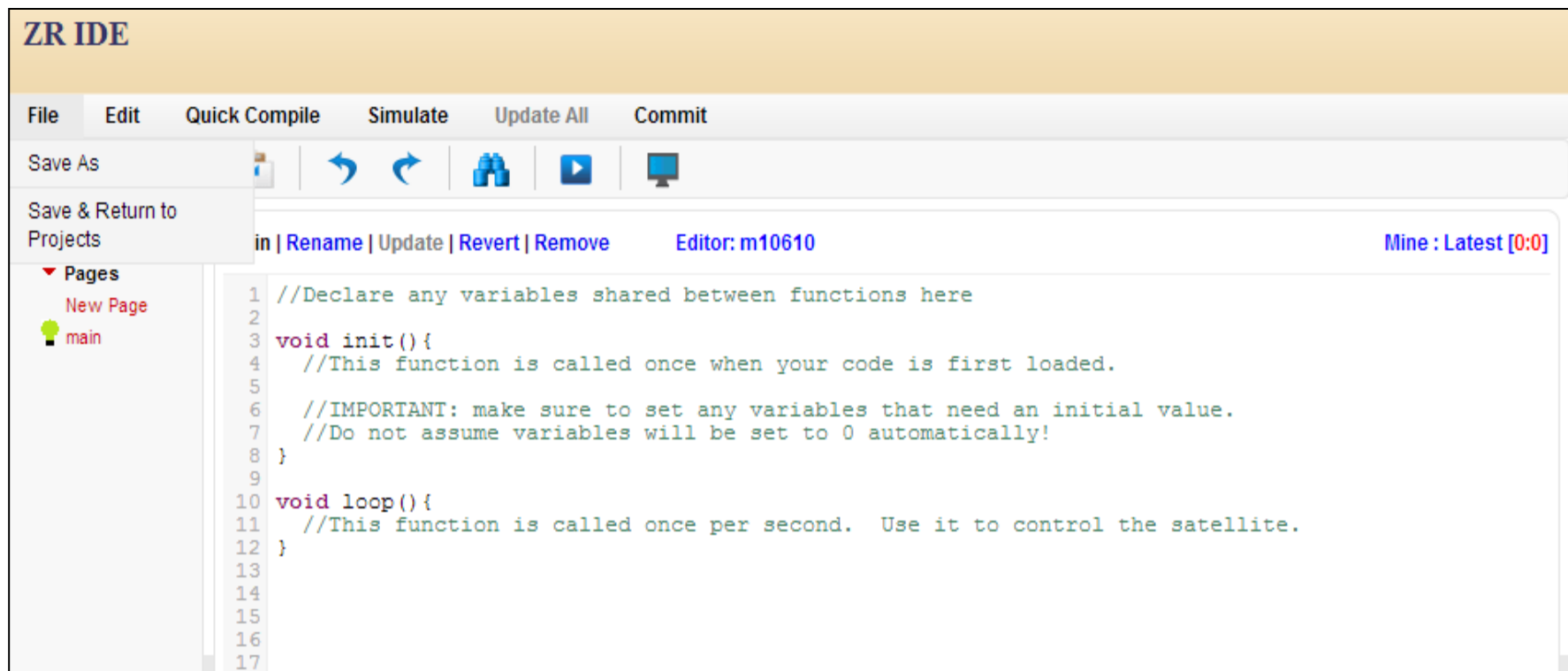
Create a New Project



- 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 “Text Editor”
- Click “Submit”



- The Text Editor version of the ZR IDE is shown here with the basic template code for a ZR project



ZR IDE

File Edit Quick Compile Simulate Update All Commit

Save As

Save & Return to Projects

in | Rename | Update | Revert | Remove Editor: m10610 Mine : Latest [0:0]

```

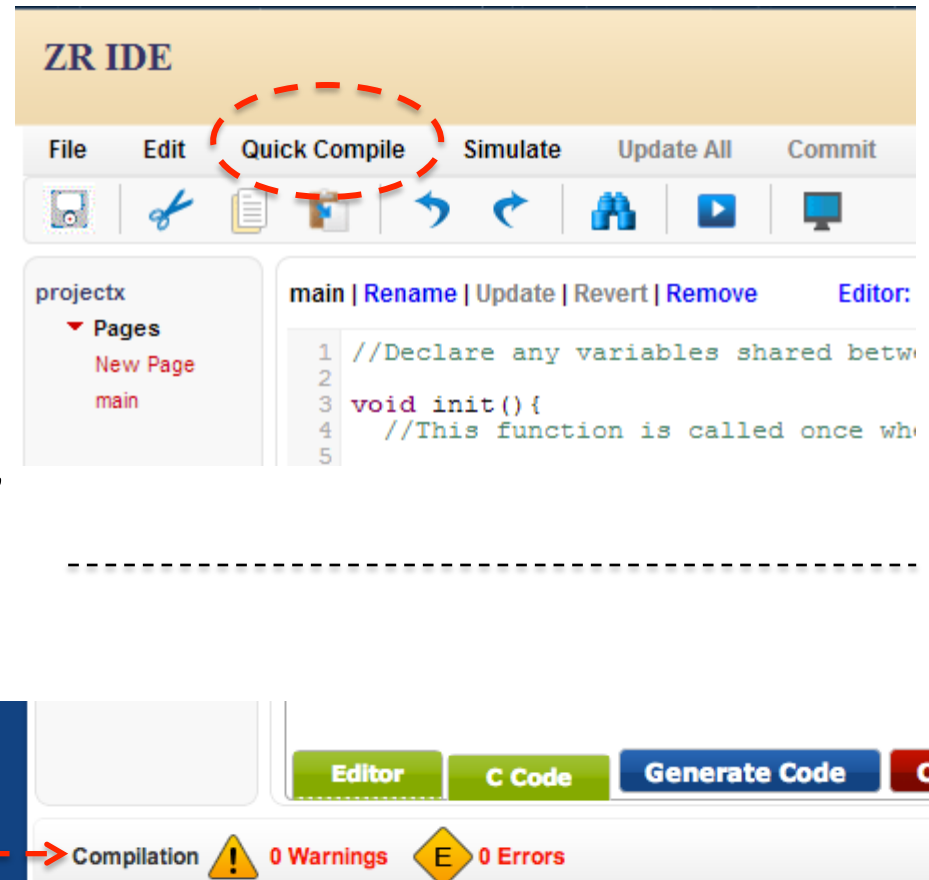
1 //Declare any variables shared between functions here
2
3 void init(){
4     //This function is called once when your code is first loaded.
5
6     //IMPORTANT: make sure to set any variables that need an initial value.
7     //Do not assume variables will be set to 0 automatically!
8 }
9
10 void loop(){
11     //This function is called once per second. Use it to control the satellite.
12 }
13
14
15
16
17

```

Quick Compile



- Now let's see a program in action!
- Click on “Quick Compile” (top menu, third from the left) and wait a moment.
 - 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:
 - Select “Create new...”
 - Type a settings name: “Tutorials_90”
 - This simulation will run for 90 seconds
- Leave everything else as is

The screenshot shows the ZR IDE interface. The top menu bar includes 'File', 'Edit', 'Quick Compile', 'Simulate', 'Update All', and 'Commit'. The 'Simulate' menu is highlighted with a red dashed circle. Below the menu bar, the 'main' file is open in the editor, showing C code. A red arrow points from the 'Simulate' menu to the 'Simulation Settings' dialog box. The dialog box has the following fields and options:

- *Load Settings: A dropdown menu set to 'Create new ...' and a text input field containing 'Tutorial_60'.
- *Simulate As: Radio buttons for 'SPH1 (Blue)' (selected) and 'SPH2 (Red)'.
- *Maximum Time: A text input field set to '60' with the unit 'Seconds'.
- *Game Variables: A section with a 'Set from Game Rules' button and a table of variables.
- *Positioning and Attitude: A section with a table of variables.
- Opponent (optional): A section with 'Empty Opponent', 'Select Opponent', and 'Clear Opponent' buttons.
- Buttons: 'Run' (green) and 'Cancel' (red) at the bottom.

	X	Y	Z	nX	nY	nZ
SPH1	0	0.5	0	0	1	0
SPH2	0	-0.5	0	0	-1	0

Simulate (cont.)



- Click on green “Run” button at the bottom
 - This will take a minute
 - You will see messages while you wait
- Click on “View simulation”
- A new browser window or tab should pop up.

Simulation Settings

*Load Settings: Create new ... Tutorial_60

*Simulate As: ☒ SPH1 (Blue) ☐ SPH2 (Red)

*Maximum Time: 60 Seconds

*Game Variables:

*Positioning and Attitude: **Set from Game Rules**

	X	Y	Z	nX	nY	nZ
SPH1	0	0.5	0	0	1	0
SPH2	0	-0.5	0	0	-1	0

Opponent (optional): Empty Opponent

Select Opponent **Clear Opponent**

Run **Cancel**

Simulation Complete

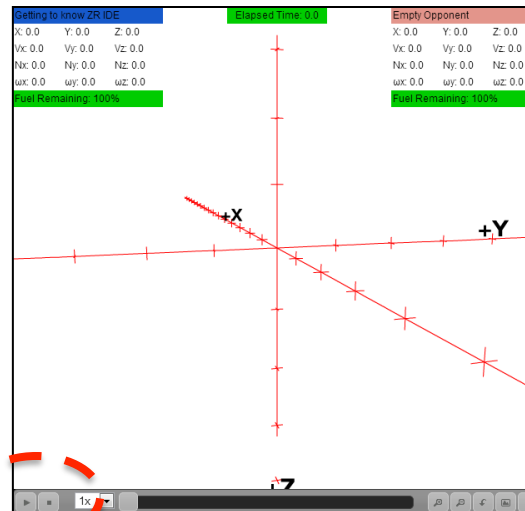
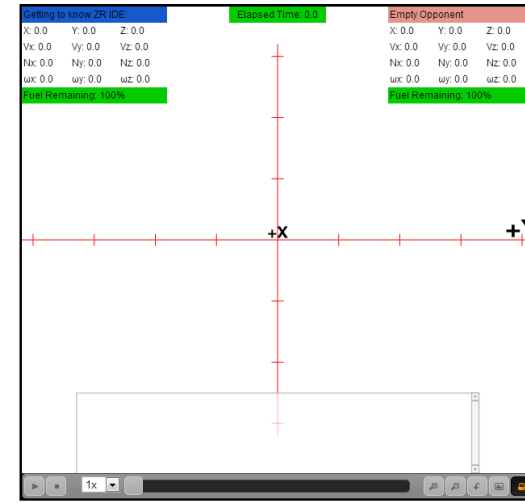
Results [SPH1, SPH2]: [0, 0]

View simulation **View Report** **Close**

View Simulation



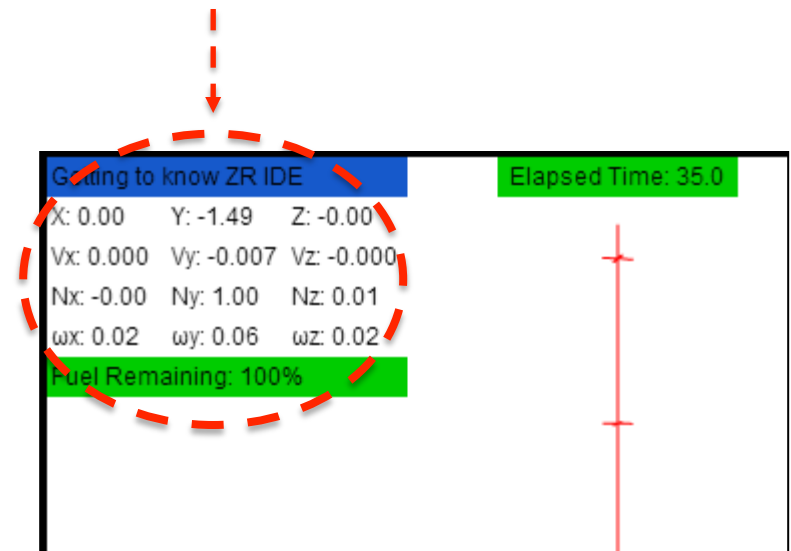
- The initial view shows y and z axis
 - horizontal line (the y-axis)
 - vertical line (the z-axis)
- To see the x axis:
 - Click and hold the left mouse button anywhere on the background and move the mouse until x, y and z axis are visible
- Click the “Play” arrow at the bottom left of the screen and 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 satellites will not do much because you have not programmed them to do anything yet!



View Simulation (cont.)



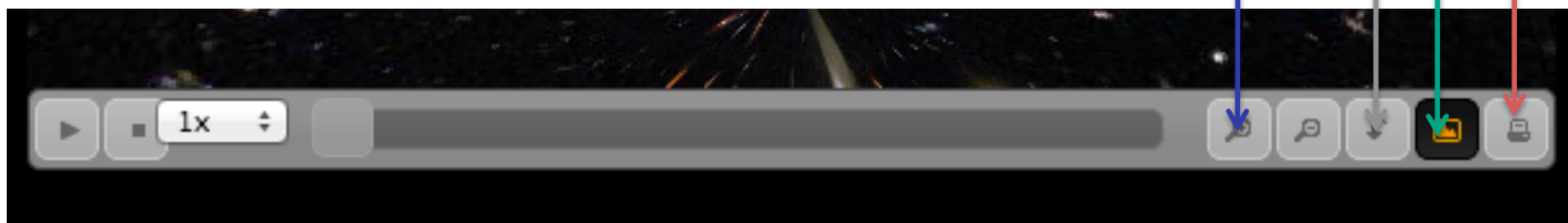
- Replay the simulation by clicking the play arrow again
- Experiment with your views by clicking on and moving the screen
- Watch the scoring box (top-left corner of the screen with blue label) which provides information about the blue SPHERES satellite:
 - where the satellite is (X, Y, Z)
 - how fast 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).
- Notice that the satellites "jiggle" a little even when they are not programmed to do anything. There is noise (random movement) built into the simulation to represent the imperfections in the actual SPHERES.



View Simulation (cont.)



- Experiment with the simulation buttons and views at the bottom to:
 - change simulation speeds (dropdown menu on the left)
 - zoom in/out
 - reset view
 - change background
 - show/hide the console





- Congratulations!
- You learned how to use the ZR IDE
- You compiled code
- You ran the code in a simulation