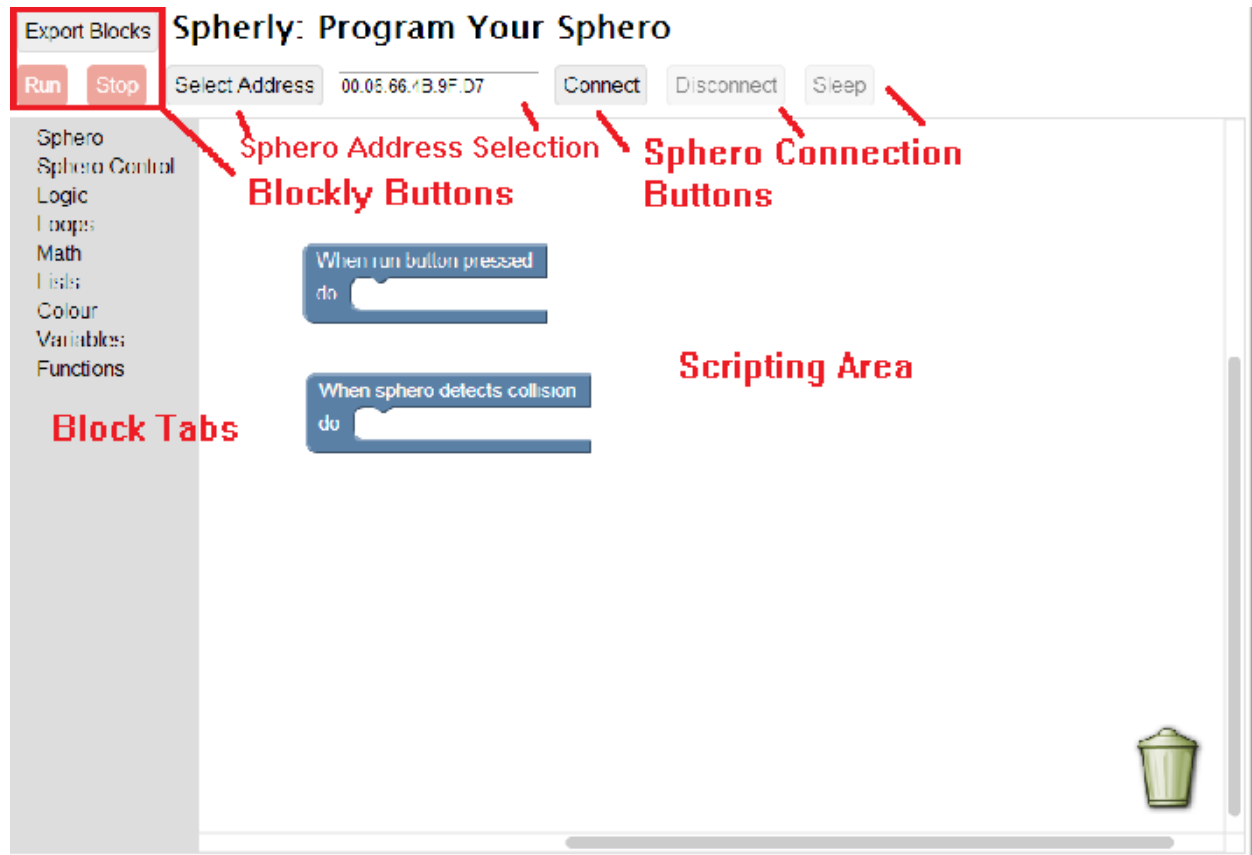


Spherly: User Manual

Spherly is a custom extension of the **Blockly** web-based, graphical programming editor. Spherly comes in two components: the front end web application that runs from a browser, and the server component which runs on your computer and will use your computer's bluetooth to communicate with the Sphero.

I. Layout of Spherly Web Application



After connecting to the Sphero (which will be discussed in the next two sections), the majority of the work will be done with the **Blockly Buttons**, **Scripting Area**, and the **Block Tabs**. Essentially, what you'll want to be doing is dragging Blocks over from the various *Block Tabs* to the *Scripting Area* into a sequential program to be executed either when the **Run Button** is pressed or when Sphero *detects a collision*.

When you have your blocks set up in the program you want, simply press the **Run** button to send the instructions from the web client, to the server, to the Sphero ball.

II. Starting the Server

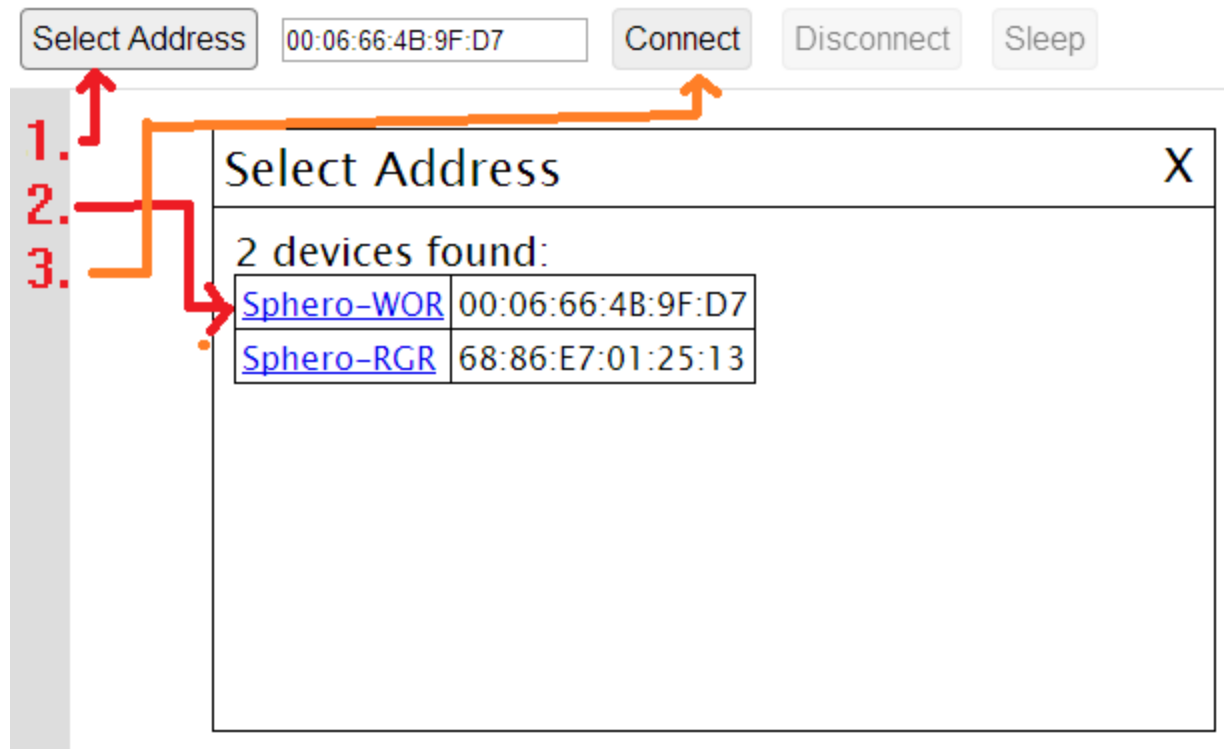
The server can be run by starting SpherlyServer.exe, which can be found from the landing page website. The server currently only works on Windows machines (tested on Windows 7 and Windows 8), but future versions will look to create a distribution for tablets and other OSs.

Also, the server will run on any Windows machine, but needs some sort of Bluetooth on the machine to be able to communicate with the Sphero. This can be achieved with built in Bluetooth on laptops, or a Bluetooth dongle for desktops.

III. Connecting to Sphero

Firstly, make sure the SpherlyServer.exe is running on the computer you want to run Spherly on. Secondly, connect up to the Sphero through the Windows bluetooth manager.

Spherly: Program Your Sphero



After your Server and bluetooth is set up, navigate to the Spherly Application and:

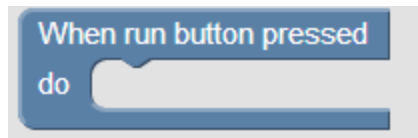
1. Click the Select Address Button
2. Select the appropriate Sphero you want to connect to through the popup window
3. Click Connect

You might have to try connecting more than once if the Server just started, but the app should notify you when you're connected to Sphero successfully, and Sphero should also light up a blueish white color.

IV. Spherly Specific Blocks

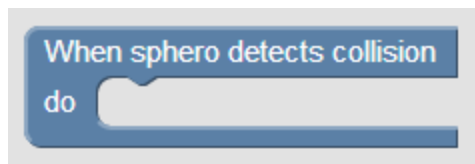
Now we'll go through all of the blocks custom created for the Spherly application.

A. Blocks under the “Sphero Control” Tab



This is one of the two blocks that are present in the Scripting Area when Spherly starts. This wrapping block will hold all blocks that you want to command the Sphero when the big **Run Button** is pressed.

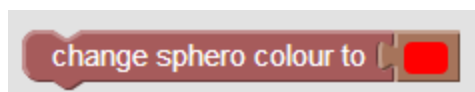
Any blocks that are outside of this run button block are considered scratch and will not have any direct effect on the canvases. (**NOTE:** an exception is any **function** created outside of the *run* block. These will not be executed directly, but can be referenced from inside of the *run* block and will function properly).



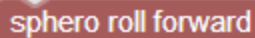
This is the other block that is present in the Scripting Area when Spherly starts. Similarly to the *when run button pressed* block, this wrapping block will hold all blocks that you want to command the Sphero when it detects a collision (i.e. running into a wall or being lightly hit)

Collision detection in this version of Spherly is somewhat sensitive, and so it is common for Sphero to detect a collision when it begins rolling or turning on its own. This is a lot more present on certain floor surfaces than others. If you don't put any blocks in this wrapping block, or delete this wrapping block, Sphero will not do anything when it collides with something.

B. Blocks under the “Sphero” Tab

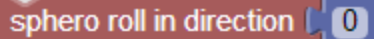


Simply enough, this block will make Sphero's LED light up to the specified color. The color portion on the right can be changed using the built in color picker (click on the color to bring up the picker), or switched out with any block from the **Colour** tab.

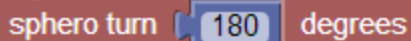
A red block with the text "sphero roll forward" in white.

This block will instruct Sphero to begin rolling forward. Sphero calibrates its positioning upon starting, so it may be helpful to use this command when you first start the application to find out what direction Sphero thinks forward is.

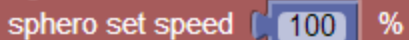
(**Note:** the direction Sphero thinks is “forward” will be changed by using the “**turn**” command below)

A red block with the text "sphero roll in direction" in white, followed by a blue input field containing the number "0".

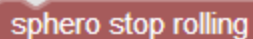
This block will instruct Sphero to roll in the specified direction. The direction is in units of degrees, and corresponds to the classic 0 to 359 degrees of a circle. Rolling in the direction of 0 degrees corresponds to “roll forward” if you have not yet used the “**turn**” command.

A red block with the text "sphero turn" in white, followed by a blue input field containing the number "180", and the word "degrees" in white.

Instructs Sphero to change its **forward** direction by the set amount. If you instruct Sphero to turn 180 degrees, it will be facing the opposite direction it was just facing, and any further “**roll forward**” commands will use this new facing.

A red block with the text "sphero set speed" in white, followed by a blue input field containing the number "100", and a percentage symbol "%" in white.

Changes the speed of Sphero to a percentage of its max speed when executed. Sphero’s initial speed is set at 100%. Max speed varies between Sphero versions 1 and 2.

A red block with the text "sphero stop rolling" in white.

If Sphero is currently rolling, this will make it stop where it is. If Sphero is not rolling, this command does nothing.

A red block with the text "wait" in white, followed by a blue input field containing the number "1", and the word "seconds" in white.

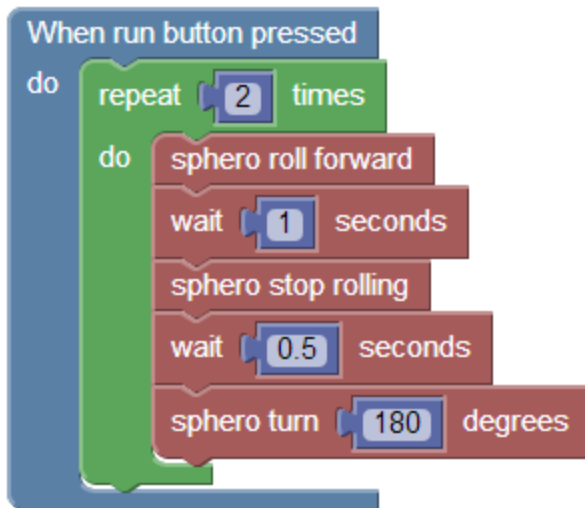
While this will not send a specific command to Sphero, it can be used to pause before you send the next command to Sphero.

For instance, because Sphero takes half a second or so to turn, it could be helpful to put a “**wait**” command after a “**turn**” command, in order to give the Sphero enough time to position itself correctly. If this is not done, Sphero may not roll in the correct direction immediately.

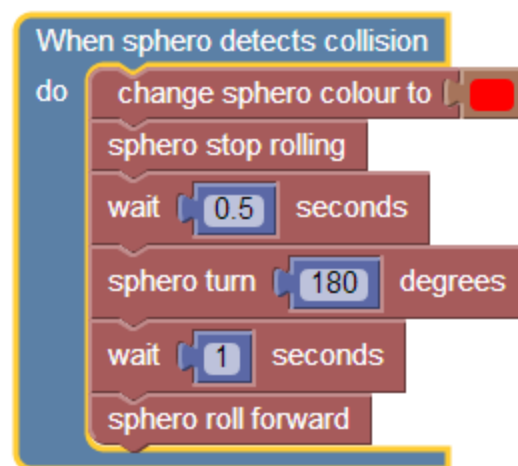
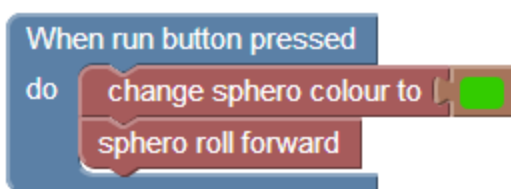
V. Code Examples

(**Note:** When working with Sphero, it can sometimes get confused when trying to right itself upward, and so it is not guaranteed that every execution of the code run the same every time. Allowing Sphero more time to execute *turn* commands by using *wait* commands is suggested, but not guaranteed to work with 100% efficiency every time)

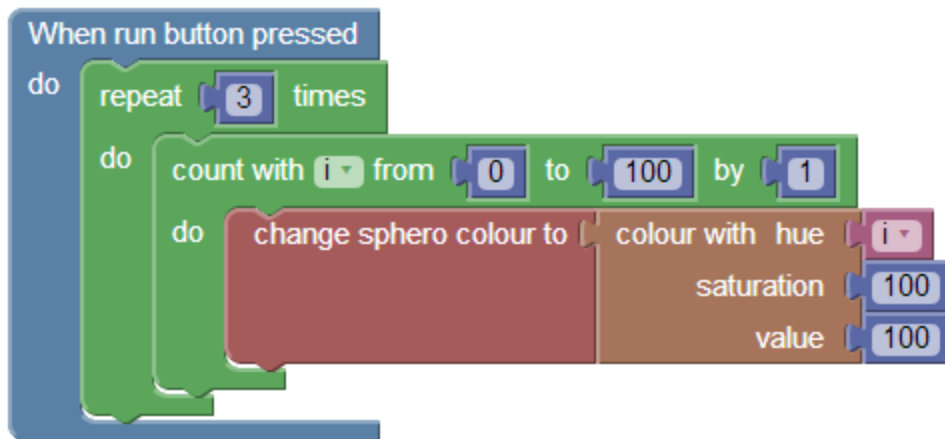
A. About Face



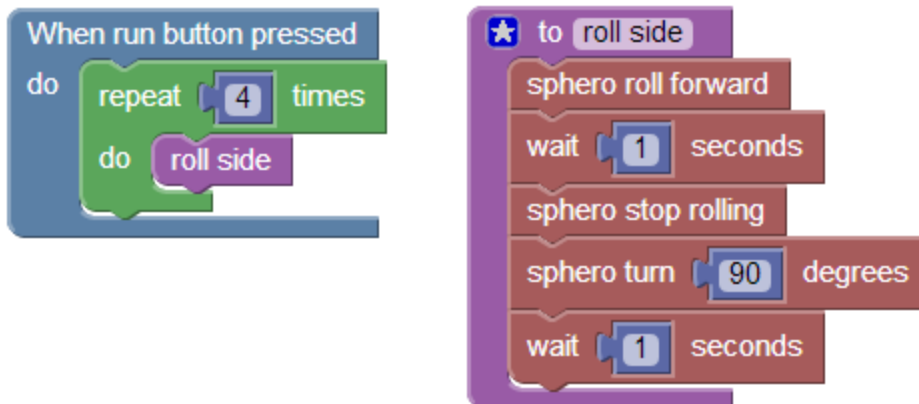
B. About Face: Collision Edition



C. Color Cycling



D. Rolling in a Square



E. Rolling in a Spiraling Square

