EuglenaScript Remote Instructions

Thank you for testing out our first iteration of the *EuglenaScript* API and corresponding web development environment. Below is information about how to use the system. If you create any interesting biological applications (video games that interact with the cells, a scientific application, or some other application we have not yet thought of), please save it on the system. If you have comments for improving the API and/or the web interface, please email them to peterwashington@stanford.edu. This is still an early system, so the more free-form feedback we can get, the better. I am looking forward to seeing what biological programs you come up with and hearing your comments!

About EuglenaScript

The Interactive Online Biology Lab is an ongoing research project at Stanford involving the development of technologies to provide interactive access to a real biology lab over the Internet at scale for concurrent users. In this particular online lab, we provide a means to study the phototaxis of a single celled micro-organism called *Euglena gracilis* through real experimentation but over the Internet. *Euglena* go through negative phototaxis in response to a certain wavelength of light – meaning they will move away from light.

EuglenaScript is a JavaScript API that allows you to program a script that can interact with *Euglena* in various ways. The API lets you manipulate one of 4 LED lights placed on the top, bottom, left, and right sides of the microscope video view. You can also programmatically read information about the *Euglena*, such as their position, velocity, acceleration.

EuglenaScript can be used to program a variety of applications, such as video games that manipulate the cells ("biotic games") and automated scientific experiments. We have provided example code of the simplest versions of some of these applications. These should be a good start for inspiring your applications.

If you come up with an especially interesting application, we will feature it in our paper and the live deployment of the website (if you give us permission, of course).

How To Use:

(1) Navigate to **biotic.stanford.edu** and create an account.

(2) Navigate to the home screen. Under "Available Microscopes", find one of the live microscopes that has a status of "Available" (is not being used by another user) and click the **"Code" button**.

(3) On the left side of the screen, you will see 5 text boxes for code, which correspond to the following events (in order): the main event loop (called every millisecond), the start of the program, the end of the program, a keyboard press, and the movement of the blue joystick.

(4) To run your code, click the **"Run Program" button**. Note that this will trigger the *startProgram* event. You cannot edit your code while your program is running. Similarly, to you're your program, press the **"Stop Program" button**, which will trigger the *endProgram* event handler.

(5) Your resulting program can be seen of the right side of the screen. To hide the code, press the **"Hide Code" button**.

(6) Your session will expire after 5 minutes. After these 5 minutes, you will no longer be able to run your code. This is due to the limited number of online microscopes. When you run out of time, save your program (see item 7). Then, navigate back to the home page and select either the same online microscope or a new one – your program will run on all online microscopes.

(7) To save your program, click the **"Save Code"** button. Similarly, to load one of your old programs, click the **"Load Code"** button.

The complete *EuglenaScript* API is listed on the right side of the page below the live microscope view. Use the example code to get a feel for how the API is used. When you first load the page, an example program is provided. This program is a "biotic game" where the player has to guess which LED light is shining based on the swarm movement of the *Euglena*. For example, if all of the cells are moving to the right, then the left LED is most likely shining (since *Euglena* move *away* from light). The following code examples demonstrate the following base functionalities:

avoidEuglenaGame.peter: This is a program for a biotic game where the player uses the W, A, S, and D keys to move the green circle from the bottom to the top of the screen. If the green circle hits any Euglena, then the player loses. Otherwise, the player wins.
euglenaCountExperiment.peter: This program counts the number of Euglena on the screen every 10 seconds and records it in a text file. When the program is finished, the contents of the text file (containing a map of times to organism count) are displayed onto the screen.
euglenaHeatmaps.peter: This program displays the count, average velocity, and average acceleration of all the Euglena on the screen. The text for each is more red when the count/average velocity/average acceleration is higher.

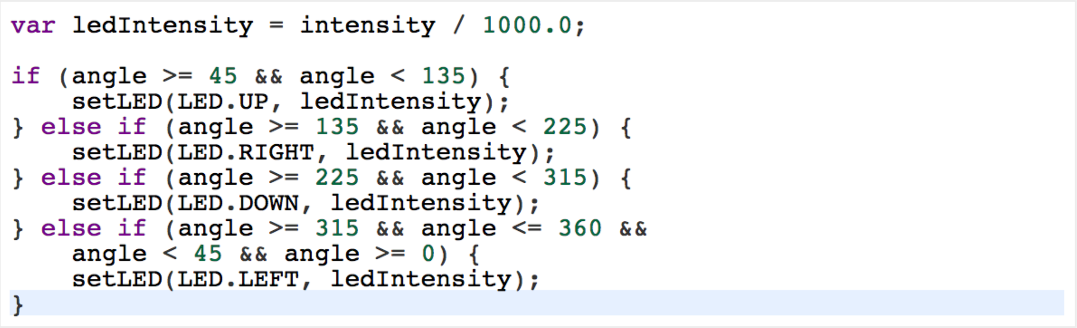
I am looking forward to seeing the biological applications you come up with! Have fun!

After you have created one or more applications, please fill out the following feedback form:

goo.gl/jSXymM

As an example, if you wanted to write a program that would allow you to control *Euglena* movement with the joystick, you could include the following code in the onJoystickChange event handler:

```
this.onJoystickChange = function(angle, intensity) {
```



```
var ledIntensity = intensity / 1000.0;

if (angle >= 45 && angle < 135) {
    setLED(LED.UP, ledIntensity);
} else if (angle >= 135 && angle < 225) {
    setLED(LED.RIGHT, ledIntensity);
} else if (angle >= 225 && angle < 315) {
    setLED(LED.DOWN, ledIntensity);
} else if (angle >= 315 && angle <= 360 &&
    angle < 45 && angle >= 0) {
    setLED(LED.LEFT, ledIntensity);
}
```

```
};
```

The angle (from 0 to 360, 0 degrees starting on west axis and moving clockwise) and intensity (from 0 to MAX_LED_INTENSITY) are passed into the onJoystickChange event. Based on the angle of the joystick, the appropriate LED is shined at the intensity passed into the event handler.

The above program demonstrates the basics of how one can program for user interaction with *Euglena*.