

# WebGL Spider Model



Arvo Sulakatko

[jsc-solutions.net](http://jsc-solutions.net)

March 15, 2012



# Contents

<b>1</b>	<b>The Why</b>	<b>7</b>
1.1	Intro . . . . .	8
1.2	Goals . . . . .	8
<b>2</b>	<b>The What - Create a WebGL Spider Model</b>	<b>11</b>
2.1	3D Visualization . . . . .	12
2.1.1	Program 23 - Calibration . . . . .	12
2.2	Arduino . . . . .	12
2.2.1	Lessons learned . . . . .	12
<b>3</b>	<b>The How - Install JSC</b>	<b>17</b>
3.1	Browser . . . . .	18
<b>4</b>	<b>Connecting model with a data cable to Arduino</b>	<b>19</b>
<b>5</b>	<b>References</b>	<b>21</b>
5.1	Document source . . . . .	21
5.2	Project Source . . . . .	21
5.3	Video . . . . .	21
5.4	JSC Web Installer . . . . .	21
5.5	Website . . . . .	21
5.6	Blog . . . . .	22



# List of Figures

1.1	Physical Spider To Be Programmed . . . . .	7
1.2	What can we see on the spider . . . . .	8
2.1	Visual Studio Web Developer Express - New Project . . . . .	11
2.2	Solution Explorer . . . . .	13
2.3	Program 23 . . . . .	14
2.4	Arduino - spiderwalk2.ino . . . . .	15
3.1	Download JSC at <a href="http://download.jsc-solutions.net">http://download.jsc-solutions.net</a> . .	17
3.2	Make sure your device is supprting WebGL . . . . .	18



# Chapter 1

## The Why

[...] people don't  
buy what you do.  
people buy why  
you do it!

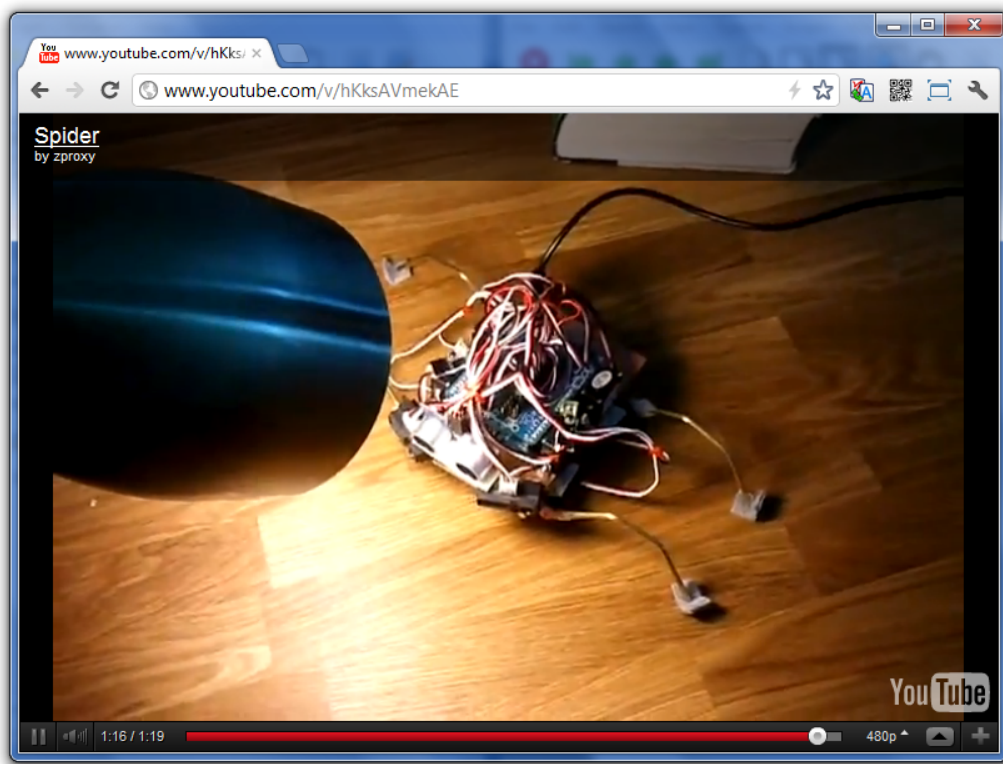


Figure 1.1: Physical Spider To Be Programmed

## 1.1 Intro

In 2011 I took a course. It was the **Advanced Topics in Biomechanics** course by **Adriano Cavalcanti, Ph.D.** During this course we had to come up with various 3D visualizations of different models. I chose to do that within WebGL. As a final task we had to come up with a mechanic spider. My part was to make it move. I had never programmed a robot before.

I was given a piece of hardware which had a few sensors and four legs.

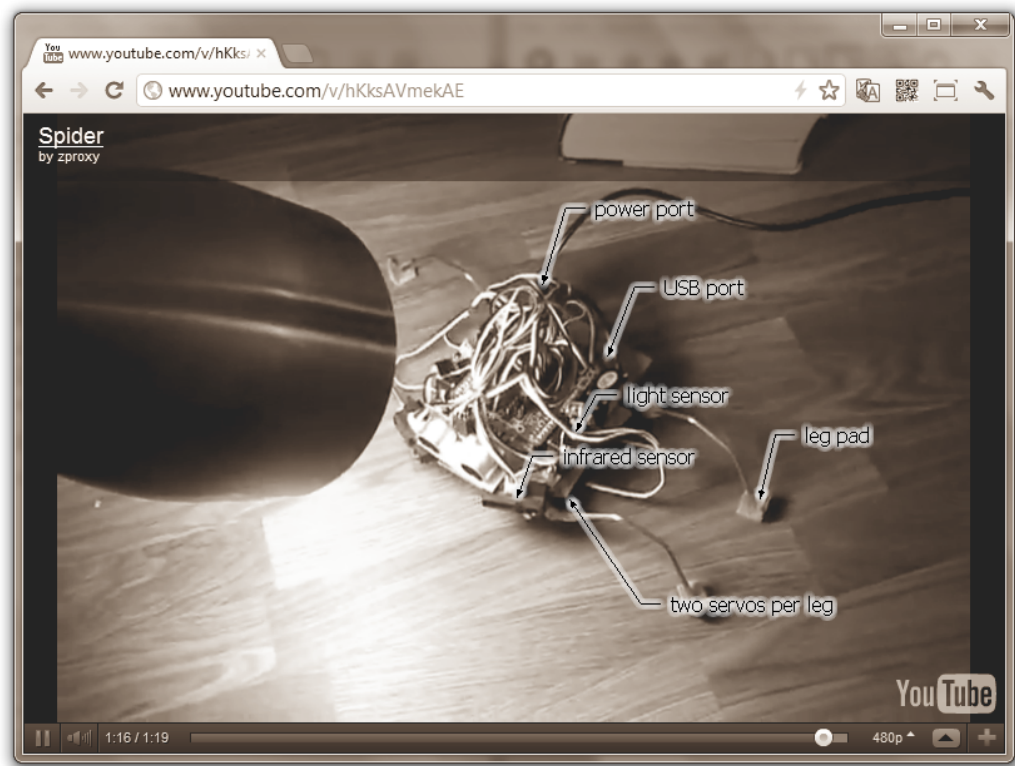


Figure 1.2: What can we see on the spider

## 1.2 Goals

For a project to be succesful goals needs to be set.

- Avoid obstacles



- Go to the light
- Stop when there

With the current setup we are able to sense light on both sides, sense distance and move servo motors to move the legs. Additionally the spider has to move without a data cable. It can have a power cable attached but cannot have the data cable.

The movement of the legs shall be time dependant. Legs can have different types of movement. We will name them as programs.

- Program 23 - Calibration
- Program 43 - Stand
- Program 53 - Mayday
- Program 13 - Turn Left
- Program 14 - Turn Right
- Program 15 - Go Backwards
- Program 16 - Go Forwards
- Program 17 - Go Left
- Program 18 - Go Right

Program code was selected at random. Keyboard codes were also considered.

While within our model we can visualize any movement we want to we will be limited by the physical version of the spider. For example movements to the left and right will not have the expected outcome.

Next, lets have a look what was built to visualize spider movement.



## Chapter 2

# The What - Create a WebGL Spider Model

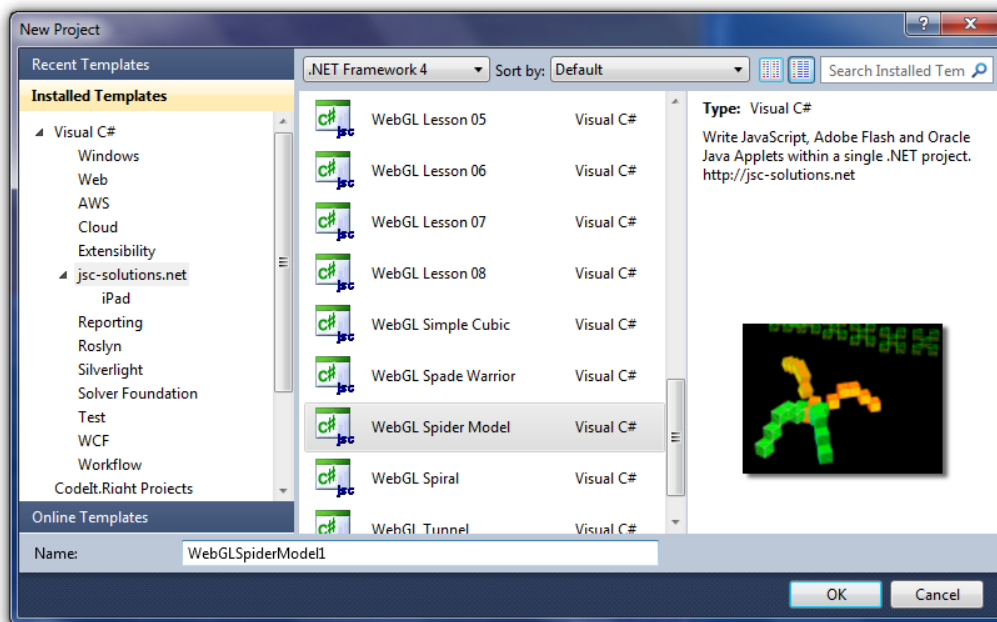


Figure 2.1: Visual Studio Web Developer Express - New Project

## 2.1 3D Visualization

While taking this project I wanted to make use of the WebGL technology. During the course itself I had learned how to make a few simple models. This time I had to do four legs and also add sensor visualization. The project itself is written in CSharp. The JSC compiler will translate it JavaScript to run in a WebGL capable web browser.

Refer to the next section to install jsc!

In this chapter we shall have a look at how to build on this example on your machine.

The project template is part of the JSC experience and as such you will be able to create a new project and go from there.

### 2.1.1 Program 23 - Calibration

## 2.2 Arduino

Although this document briefly describes Arduino related development it is considered out of scope and is not part of the default **jsc eXperience**.

At this time jsc does not support any languages that target Arduino platform. As such I had to make use of Arduino programming language. Otherwise I could of had my CSharp code compiled to Arduino. This would of had allowed me to use the same code in the visualization and on the chip.

### 2.2.1 Lessons learned

While programming for Arduino I had to manually port my code I had written for the visualizer to the Arduino platform. In doing so I discovered that the int is considered to be 16 bits and that I cannot make use of function pointers. To overcome that I had to divide before I did my multiplication. Yes I had to track down an overflow bug before I realized this. The callbacks I used were simple. They only had a few parameters. This allowed me to replace the function pointer with pointer to variable and have the same behaviour of code.

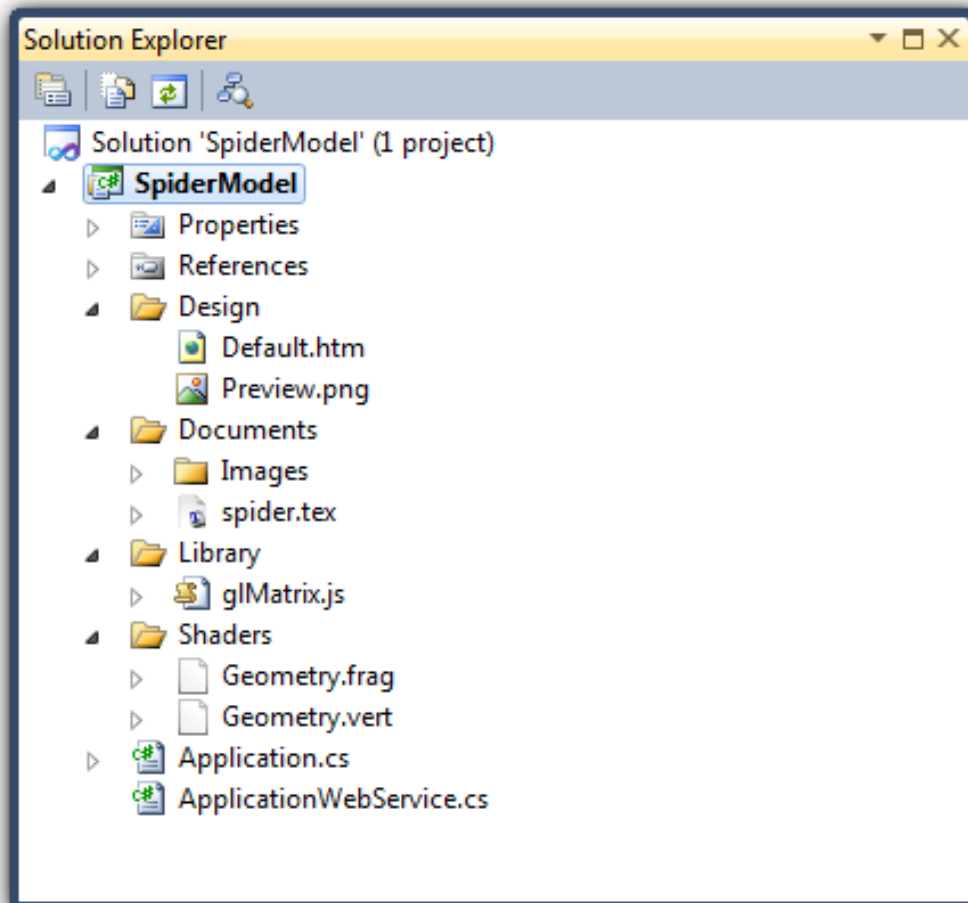


Figure 2.2: Solution Explorer

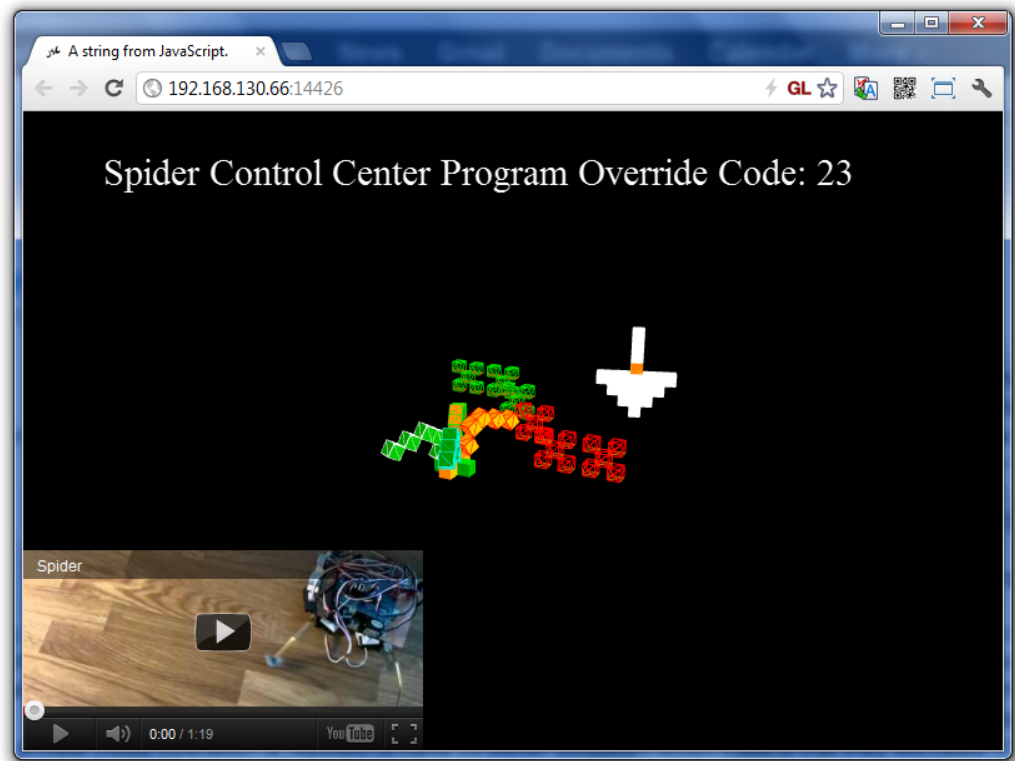


Figure 2.3: Program 23

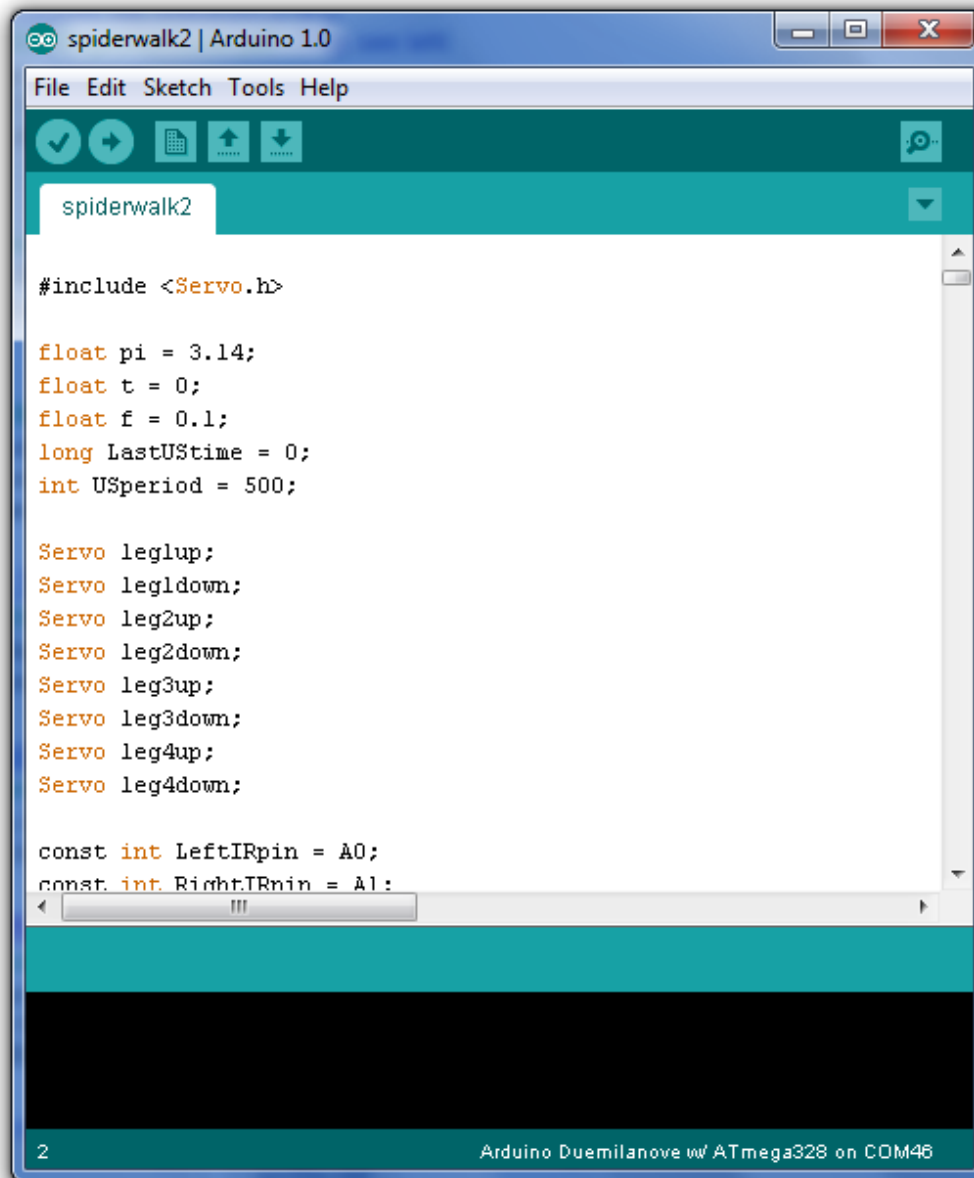


Figure 2.4: Arduino - spiderwalk2.ino





## Chapter 3

### The How - Install JSC

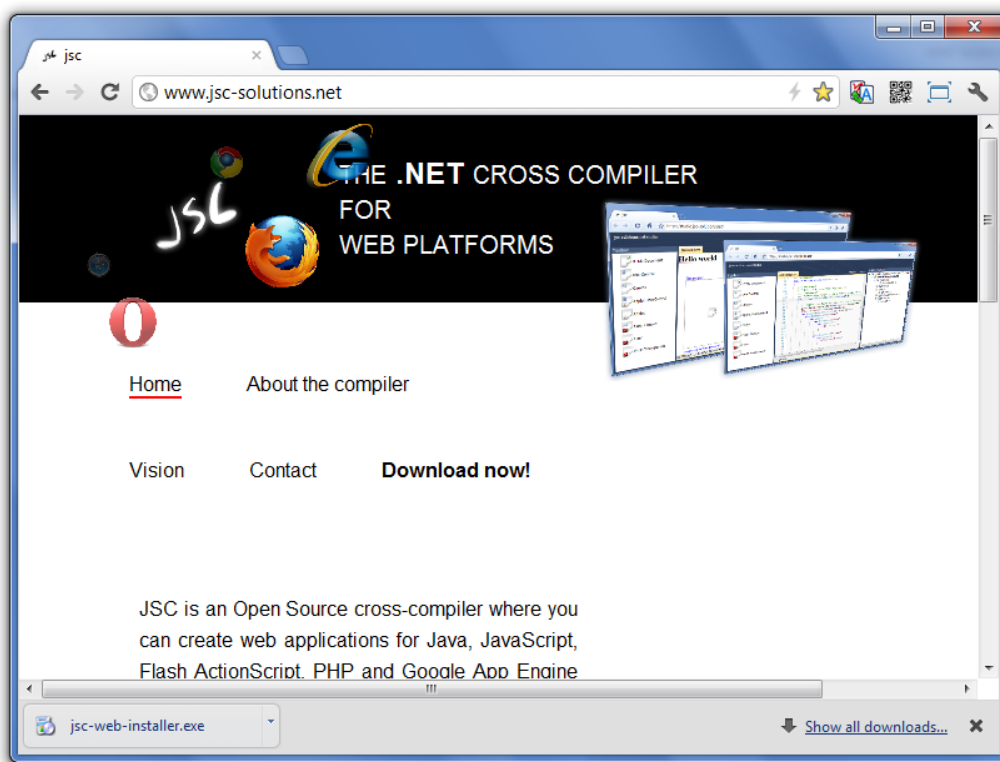


Figure 3.1: Download JSC at <http://download.jsc-solutions.net>

Installing JSC is easy. Before you do make sure you have installed Visual

Studio 2010 Web Developer Express.

### 3.1 Browser

For older machines WebGL might need additional manual configuration.

```
chrome.exe -enable-webgl -enable-apps -ignore-gpu-blacklist
```

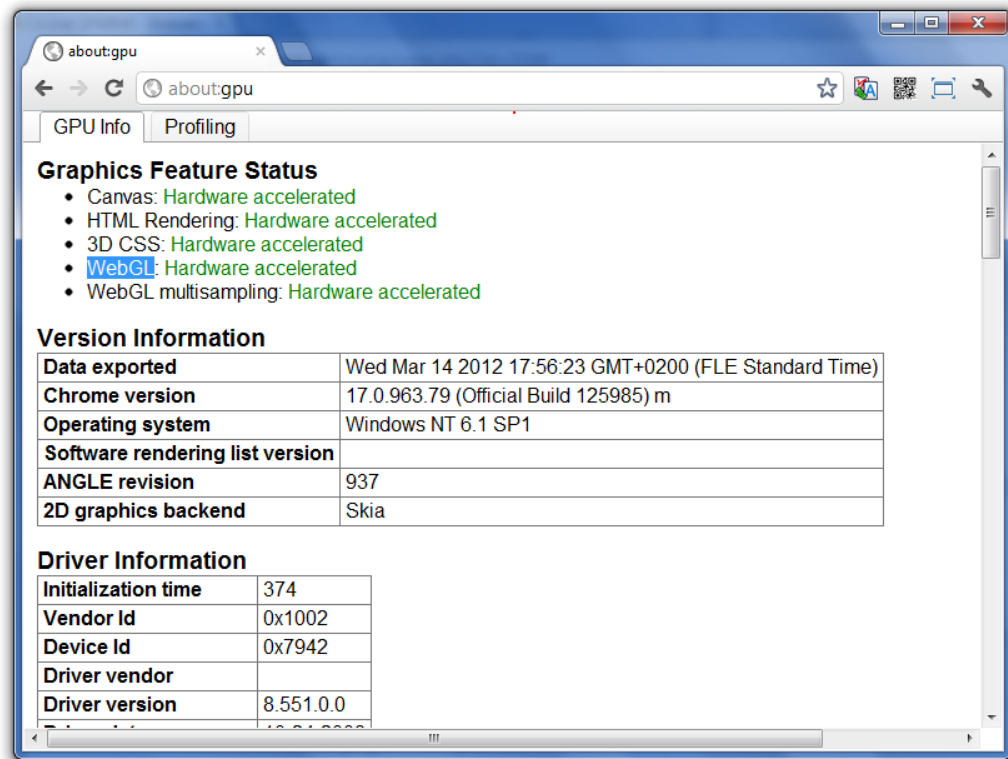


Figure 3.2: Make sure your device is supprting WebGL

## Chapter 4

# Connecting model with a data cable to Arduino

This model was extended in a related project to connect to the Arduino via USB Serial Port. The spider is listening for Program Override code. This allows to issue specific commands to the spider and test out new ideas.



# Chapter 5

## References

### 5.1 Document source

<https://jsc.svn.sourceforge.net/svnroot/jsc/examples/javascript/ArduinoSpiderControlCenter/SpiderModel/Documents/spider.tex>

### 5.2 Project Source

<https://jsc.svn.sourceforge.net/svnroot/jsc/examples/javascript/ArduinoSpiderControlCenter/SpiderModel/>

### 5.3 Video

<http://www.youtube.com/v/hKksAVmekAE>

### 5.4 JSC Web Installer

<http://download.jsc-solutions.net>

### 5.5 Website

<http://www.jsc-solutions.net>

## 5.6 Blog

`http://zproxy.wordpress.com`