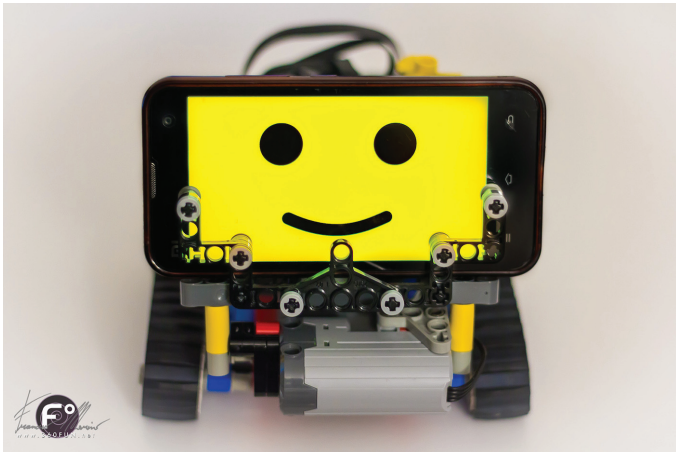
LEGO® (SBRICK) + WEB BLUETOOTH = ROBOTS!

By Francesco Marino

Images by Francesco Marino

What I wanted to pursue with this project was connecting my web knowledge with the physical world in the easiest way possible (without using Arduino or similar) and provide easier access to robotics for everybody. I have been a LEGO® lover since I was a child and I always dreamed of buying the MINDSTORMS set, yet for various reasons I never did. Instead I bought some LEGO® sets with Power Functions elements (battery packs, IR remotes/receivers and motors), which provided an easy way to control my MOCs remotely, though with the limitations of the IR connection. In 2014 a project called SBrick (by Vengit Ltd.) appeared on Kickstarter and it simply blew my mind because it promised exactly what I needed: a Bluetooth RC Receiver compatible with LEGO® motors and any Bluetooth Low Energy device. Google's later release of their Chrome browser with integrated Web Bluetooth API is what gave me the final push in the direction of using web technologies to control LEGO® robots!



BUILDING AN INTERFACE

I wrote two basic libraries; the first being Bluetooth.js [1]. This is designed to help people start playing with Web Bluetooth and perform basic operations like connect, disconnect, read, write and cache services and characteristics. The second library is Sbrick.js [2]. This is the heart of the project, and I've spent a lot of time building it since I had no previous experience with a lot of the technology involved, and I have learned much while developing it. The library depends on Bluetooth.js and can easily be included in any web page, it also supports SBrick Plus and it's able to get data from LEGO® compatible sensors. Both libraries are open-source and are available on GitHub. I then used my favourite WYSIWYG web editor, Tumult Hype[3]. With just an easy drag and drop it is possible to add Bluetooth.js and SBrick.js (in this order) inside Tumult Hype and immediately start sending commands to the SBrick with just few lines of code!

BASIC SBRICK UI CONTROLS

I think the Profile Designer made by the SBrick team is a great tool for basic users – it's really easy to create a customised controller and use it in the SBrick mobile app. At the same time I felt there was a hidden potential waiting to be exploited, so I decided to build similar controls using Tumult Hype, allowing advanced users to play with their SBricks at another level and without having to deal with compilers or tons of code. All these controls work with mouse or touch events, so you can use them from both computers and mobile devices! I used those controls to create two generic joypads and manage the two most common driving situations: car and tank.

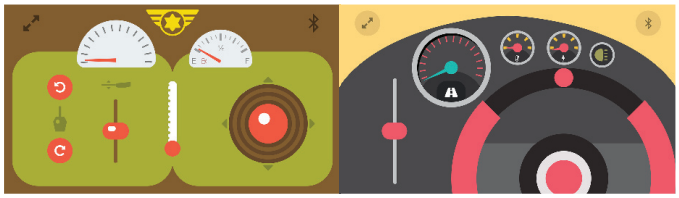
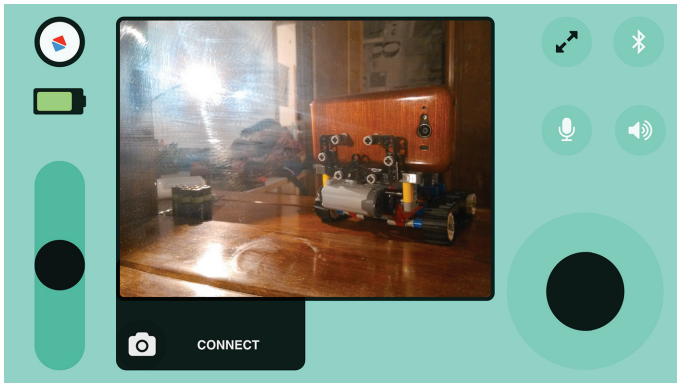


Image caption: A classic controller for your RC vehicle! The UI components are made with free elements found on the web (Font Awesome, The Noun Project…). The battery and the temperature gauges are nicely integrated into the dashboards.

SMART BRICK + SMART PHONE

Since our smartphones are equipped with cameras and microphones, it seemed only logical to me to put my mobile phone on the LEGO® robot and use it to see and hear from my LEGO® creation's point of view. In this case two devices are needed: one is installed inside the robot and the other one is used as a controller.



I used WebRTC to stream audio and video directly inside the SBrick Web Controller, creating a node-to-node connection between the smartphone and the remote control device through the local WiFi. Following this path I also started to play with the JavaScript Device APIs to get access to the mobile

sensors (accelerometer, gyroscope, compass, etc). In this way it is possible to see and hear from the robot's perspective, get sensor data (such as its orientation) and operate from greater distances (through the internet).

PROGRAMMING THE ROBOT

Thanks to the Hype 'Timeline Actions' it is really easy to program actions without writing code, but if you are a web developer the possibilities are endless: you can plan a set of actions or develop complex programs to create robots capable of interacting with their environment using your smart devices!

OVER TO YOU

Well, now it's time for you to have fun with all of this and explore new possibilities!
If you prefer to create autonomous robots and you have more advanced skills be aware that modern mobile devices are powerful enough to manage complex algorithms even if written in JavaScript, so you can use your smartphone camera to track objects, distinguish colors, respond to voice commands and much much more! You can combine the data from the mobile sensors with the data from LEGO® sensors or other Bluetooth devices and make really complex creations! I hope people enjoy this project because I made it to have fun and to help creative people and artists to express themselves using accessible technologies.

MAKING ART

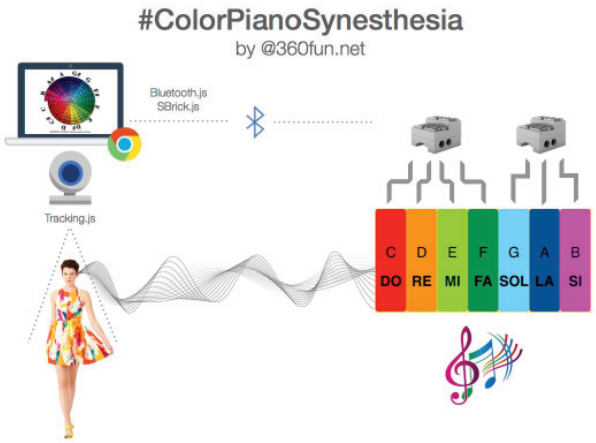
After writing SBrick.js I decided it was time to show how this little JavaScript library could be used in a creative and fun way! At the same time some friends asked me if I was interested in making an art installation for the event 'Absurd! An Exhibition on Human Absurdity'[4], organised by UMANO COLECTIVO in Studio P52 (Barcelona).

THE IDEA

I wanted to create something that would feel magical and that could be played with naturally. In my personal process of learning and exploring possibilities, I felt that it was time to create an interactive interface that didn't require a perfect correlation between action and reaction. I already knew of the existence of Tracking.js (a computer vision algorithm for browser environments) and I was looking for a way to use the color tracking feature. Moreover, I'm not a musician but generative music got my attention – so these two things were an obvious match. The last piece of the puzzle was already there in the studio: a vintage vertical piano, untuned but perfect for my needs!

THEORY

The relationship between sound and color is nothing new, which is why I didn't spend too much time on it. I instead trust those people who have dedicated their lives to this specific topic. I found some conversion schemes and just applied the color/note relationship in my code.
What I found interesting and what moved me in this direction is the condition of some people called synesthesia – I wondered how it would be to 'listen through the eyes' and that's the main reason why I made the ColorPiano[5]. And of course I thought it would be really cool to play music just by moving in front of a camera!



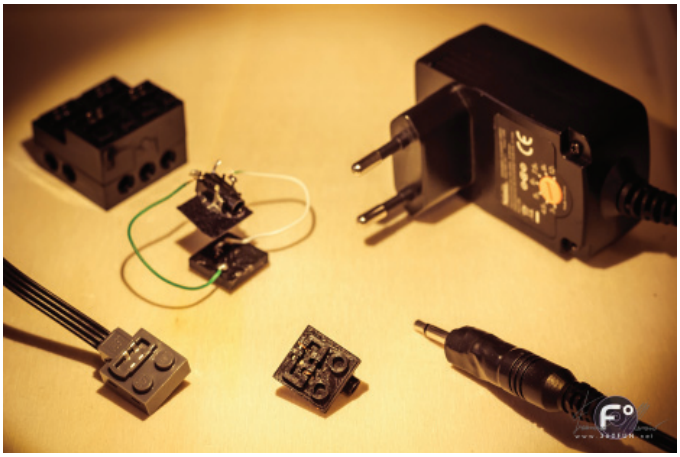
HARDWARE

I used LEGO® Technic to build the robotic fingers, and Power Functions to animate them.
The mechanism of the finger is pretty simple: it attaches to the wood of the piano and uses a wheel as counter balance to bring it back to the original position using gravity. Since the installation had to run for quite a few hours I had to put wires in to prevent them from changing position or falling.

To control the motors I used two SBricks controlled remotely by a MacBook Pro Retina. The color tracking operations are pretty heavy so I decided to avoid using a tablet/phone, though technically the code can run on any compatible device!

A LITTLE LEGO® HACK

For the power supply I couldn't use batteries since the ColorPiano had to run for several hours, so I soldered a mini-jack with a 3D printed LEGO® Power Functions-compatible socket base. In this way I could connect both SBricks to a single DC 9V 1A AC Power Adapter.



SOFTWARE

To build the UI I used my beloved Tumult Hype. I kept it really minimal since I didn't have much time to spend on it. If no SBrick is connected it plays the notes from mp3 files.
To manage the color tracking I used Tracking.js [6], assisted by a function to convert RGB values into HSV, making it easier to define a range for each color.
The entire project relies on Bluetooth.js and SBrick.js to communicate to the SBricks through Web Bluetooth. Everything runs inside Google Chrome without any external dependency.

COLORPIANO LITE

Thanks to a collaboration with Maker Convent and Ro-Botica, the ColorPiano also made an appearance at the Barcelona Maker Faire 2017 in a Lite and more transportable version. In this case I used a MIDI piano – the mechanism is similar but definitely simpler. Thanks to the spring inside the piano keys it was no longer necessary to use a counterweight to return the robotic finger back into place. The motor array is easily detachable and stays in place thanks to the weight of the piano itself.

Official page of sbrick.js and related projects: sbrick.360fun.net
You can support this project on patreon.com/360fun

[1] https://360fun.github.io/bluetooth.js
[2] https://360fun.github.io/sbrick.js
[3] http://tumult.com/hype
[4] http://umanocolectivo.com/absurd
[5] http://www.360fun.net/colorpiano
[6] https://trackingjs.com
#