Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnline CONFERENCE PAPER ARCHIVE

IFAC-PapersOnLine 49-6 (2016) 203–206

The Use of 3D LED Cube for Basic Programming Teaching

Katarína Žáková

Slovak University of Technology
Faculty of Electrical Engineering and Information Technology
Bratislava, Slovakia, (e-mail:katarina.zakova@stuba.sk)

Abstract: The paper demonstrates how 3D LED cube model can be used in education process and help to learn basic concepts of programming. It is realized in two ways -in the form of WebGL model and as a real device, as well. The application is available online and therefore it can be used anytime and from anywhere.

Keywords: computer based education, LED cube, 3D model, WebGL.

1. INTRODUCTION

Nowadays computers are used everywhere. Concepts of computer based design, computer based manufacturing, computer based production, computer based assessment, computer based education and many others are not rare.

In control area computers are very often used as controllers that compute a control variable to control a considered system. To define the control algorithm in computer based environment one must have at least a basic knowledge of programming. It does not includes only calculation of mathematical expressions but also control flow statements (i.e. loops or conditional commands), as well. Therefore learning of programming techniques is inseparable part of university study also in fields of Automation, Cybernetics, Process Control, Robotics or Industrial Informatics.

Of course, learning of programming brings its difficulties. Loops, conditionals, arrays and recursion have all been identified as language features that are especially problematic, and could benefit from particular attention (Robins et al., 2003). Du Boulay (1989) notes that "for" loops are problematic because novices often fail to understand that "behind the scenes" the loop control. Robins et al. (2003) mention that some novices expect, based on a natural language interpretation, that the condition in a "while" loop applies continuously rather than being tested once per iteration. Hristova et al. (2003) tried to identify common Java errors that are accomplished by students. They reported 20 errors that they considered to be essential from the educational aspect of their project. Three of them were connected with the syntax errors in loops and if commands.

Our aim was to find a tool that could contribute to this part of programming and facilitate its learning in a little bit more attractive way. For this purpose we decided to use visualization of the resulting code by means of 3D LED cube. Similar approach of learning via various forms of visualizations can also be found e.g. in (Yu and Abarca, 2013) or (Rudder, 2007).

2. LITERATURE OVERVIEW

Using Google search and looking for the term "LED cube" we got 5 210 000 results. In spite of the fact that in reality this number will be smaller, it shows that the topic is quite popular. It is also evidenced by the fact that similar search in YouTube portal gives approximately 195 000 results. In both cases many of contributions provide guidance on how to construct such cube.

Following Google Trends statistics (Fig.1) it is evident that the topic started to be interesting at the end of 2010 and the most often realized cube has 8x8x8, i.e. 512 LEDs.

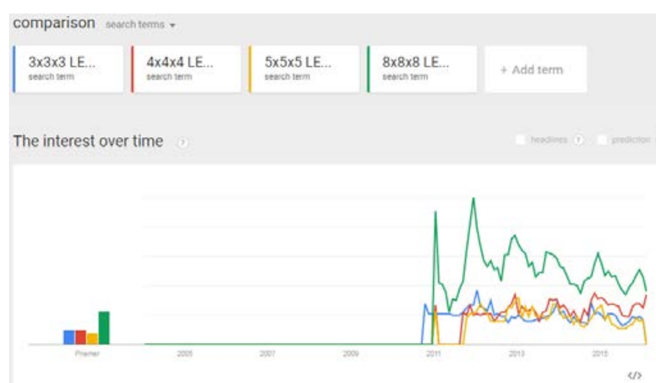

Fig. 1. Google Trends statistics.

We can find many realizations that use various microcontrollers, e.g. on 8-bit PIC microcontroller (Images SI), 16 bit PIC24 Microchip microcontroller (Ausley et al., 2013), Atmel ATmega16 (Brinkman, 2010), ATmega32 (LostRite, 2013). Several solutions use FPGA. In the last period Arduino based solutions are very popular. Such example can be found e.g. in (Sauer, 2012) where Arduino Mega microcontroller was used.

An alternative to the methods based on microcontrollers is presented in (Rob et al, 2012). It shows how the LabView

10.1016/j.ifacol.2016.07.177

software environment can be used for controlling a real 8x8x8 LED cube using a data acquisition board NI-6221.

The majority of presented cubes use single color LEDs, but there also exist solutions using RGB LEDs that enable to change color of the light.

Some realizations are even able to produce visual effects on the base of audio input. Such visualisations are described e.g. in (Daigle et al., 2011) or (Jackowski and Stepniewicz, 2012).

In spite of the fact that the most of solutions taken from Internet describes own design and realisation, there also exist commercial products that offer either the fully assembled cube or kits with pre-made schematics requiring the user to assemble the cube (see e.g. http://www.imagesco.com/led/cube.html or http://hypnocube.com).

3. REALIZATION

Our solution or better to say a tool was realized in two versions – in the form of an animated model and as a real laboratory device, as well. The first one can be used in parallel by several online users. The second one is in each moment available only to one user. It is guaranteed by the user registration and booking the pre-reserved time interval.

3.1 WebGL Model

The animated LED cube model was prepared using WebGL. WebGL (Web Graphics Library) enables to render interactive 3D computer graphics within any compatible web browser. It enables to create very realistic bodies that user can rotate and enlarge to all details that he or she would like to follow. We tried to exploit WebGL features for demonstrating behavior of the system that is very often used in the educational process.

For preparation of WebGL graphics one can use several software graphical environments, e.g. WebGL-Publisher, CopperCube, Clara.io, 3DTin, Blender, SketchUp.

Our WebGL LED cube model was quite simple and therefore we used only Three.js library. It enables easy preparation of 3D objects that can be visualized in the browser. While a simple cube in raw WebGL would turn out hundreds of lines of JavaScript and additional extra code, a Three.js equivalent is only a fraction of that. It runs in all browsers supported by WebGL and is available under the MIT license.

The visualization of the model can be seen in Fig.3.

After the model realization it was embedded to the final web page that was prepared by standard technologies: html (web page structure), cascade styles (graphical template) and JavaScript (interaction). The whole application is running just on the client side.

The user interaction with the model is realized via JavaScript that is entered to the application via a text area input field.

3.2 Real Device

Our realization of the real LED cube laboratory model (Fig.2) is based on Arduino motherboard. It enables to interact with the control unit of the cube and in this way to influence the behavior of LEDs (their switching on and off).

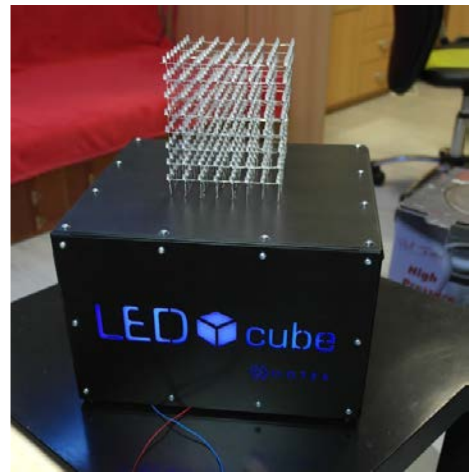

Fig. 2. 3D LED cube laboratory model.

The Arduino Uno is a microcontroller board based on the ATmega328 (ATmega8 and ATmega168 can also be used). It has 14 digital input/output pins (6 of them can be used as PWM outputs) and 6 analog inputs. The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The board can operate on an external supply of 6 to 20 volts, however, the recommended range is 7 to 12 volts.

Our aim was to control the LED cube remotely. Therefore it is connected to the web server where the supporting web application is placed. The interface for the user is similar as in the case of the animated model – one text area serves the user for entering the control code.

4. FUNCTIONALITY

For influencing LED cube diode behavior two functions were prepared. Function `on(x,y,z)` enables to switch on the diode whose position is determined by coordinates that are entered as input arguments of the function. On the other side the function `off(x,y,x)` enables to switch off the specified diode. Since we consider 8x8x8 LED cube, parameters x, y and z are integers from the interval <1, 8>.

Each of three input parameters can also be set to 0. In this way we can influence not only one diode but several ones. Setting $x = 0$ defines the full range of all available indexes for x (i.e. it substitutes 8 various commands where $x = 1, 2, 3, \ldots, 8$). This approach simplifies the diode manipulation when in one moment we can switch on not only one diode but several ones. As a consequence the command `on(0,0,0)` enables to switch on the whole cube at once.