# Scratch for Arduino: Exergaming Development

Alka Harriger
Purdue University
harrigea@purdue.edu

Mayari Serrano
Purdue University
serranm@purdue.edu

## Abstract

Currently, obesity has become one of the major health concerns in the United States. A way to relieve this problem is creating fitness activities using the technology and tools available at hand. "Punching Pad" is a prototype that utilizes Scratch for Arduino software and the Arduino Board to make this possible. This device is not only considered a viable fitness activity, but also it could help to inspire children to build other gadgets that, in fact, would facilitate the acquisition of programming skills and basic electronics concepts. The authors use free downloadable software and widely available electronic components to develop this demo, which can be viewed as an alternative to commercial gaming consoles in a learning environment.

## Introduction

Childhood obesity in the United States has reached epidemic levels, triggering physical and emotional health deterioration. Improper nutrition along with inadequate physical activity constitutes the underlying causes for obesity. In all human life stages, it is important for good health to be physically active; this is essential for children who are developing lifetime habits. Additionally cost of care and treatment of obesity and its co-morbidities has increased private and public health care expenditures [1].

Teenagers in the United States devote about 6 to 16 hours per week using computers, phones, watching television, and playing video games. For many years, the increasing availability of technology was seen as a problem instead of a solution. An approach to increase physical activity among children and teenagers is "Exergaming" which is defined by Klein & Simmers as follows: "Ability to tie video games and exercise into a single medium for the benefit of making exercise fun" [2].

The Department of Health and Human Services advises young people, ages 6 to 19, to engage in moderate to vigorous physical activity for 60 minutes per day [1]. This goal could be achieved by using interactive games to influence physical activity in young people's routines.

The present study has the objective to investigate whether Scratch for Arduino is a valid tool that can be used as an early instrument to teach programming concepts and to help foster STEM (science, technology, engineering, and mathematics) and fitness enthusiasm in children.

### *Scratch*

Scratch was developed as a programming tool that would appeal to people who did not think of themselves as programmers. Scratch was designed to facilitate program scripting to all of its users without any age or background limitations. Scratch developers pursued a programming tool that is able to produce interactive stories, animations, games, and simulations that could be shared in a community collaborative platform [3].

Scratch users are mainly between the ages of 8 and 16, most of them 12 years old. There are also a considerable number of adult users. Users learn mathematical and computational concepts while they creatively express themselves, reason systematically, and practice cooperative work [3].

There are many benefits associated with the ability of programming: the range of what users can create and learn while using a computer is increased. Programming mainly develops "computational thinking," which promotes problem solving and strategy skills that could be used in non-programming settings. In fact, since programming comprises the creation of external exemplifications of individual problem solving processes, programming delivers the opportunity to reflect individual thinking [3].

Computer programming skills are viewed as a technical activity suitable for only a small part of the population. Other tools that tried to foster programming enthusiasm in children failed because languages were too difficult to use, the activities used were not appealing to young people, and when users encountered problems, there was no guidance available [3].

Scratch uses a building block programming analogy to construct the program script. With this software, it is possible to incorporate and manipulate images, sounds, and animations. Scratch allows users to share projects using an online platform, so the Scratch community can interact and share techniques. More than 250,000 projects were shared in an 18-month period after the tool was launched [3].

Scratch developers created an infrastructure that allows the translation of programming blocks to any language with any character set [3].

Several studies state that children find Scratch fun to use and engaging. Scratch has proven to successfully introduce basic programming concepts without the distraction of syntax. Additional benefits of using this tool to introduce programming is that children associate this experience with enthusiasm and enjoyment that eases the transition to more sophisticated language such as Java and C [4].

### *Arduino*

Arduino programming language is grounded on C/C++. Kato defined physical computing as "the interaction with physical objects by controlling sensors and actuators attached to microcontrollers" [5].

Arduino is an open source electronics platform where users need neither electronics nor programming previous knowledge. Using this platform enables users to move beyond consuming technology to actually produce it [4].

First-time Arduino users learn to program by repetition of steps and contact with the physical components. Nevertheless, the process of writing a problem is a complex mental process, requiring repetition even when working on small or simple programs [5].

### *Scratch for Arduino*

Scratch for Arduino shows useful features in physical programming. This software enables the communication between an external Arduino board and Scratch. This version of Scratch uses fewer, simpler blocks in which all actuators are directly controlled by a PC without using a controller unit [5].

Visual programming characters enhance physical programming learning, because the programs and flow can be graphically understood. This feature could be very useful in classroom settings [5].

Physical programming allows users to interact with real world objects while implementing electronic circuits. A trial and error approach offers a great opportunity to learn about electronics and computational concepts. Moreover, handlers receive intellectual rewards for what they had achieved [5].

Booth & Stumpf's study proved that task type activities, such as creation and modification, contribute to successfully use a programming language. Moreover, participants of this particular study felt more confident using visual programming to modify a program [4]. K-12 schools and some universities are using Scratch to introduce basic programming concepts to students [3]. Arduino has been also used by institutions to introduce physical computing to their students [5].

## Punching Pad Development and Implementation

The goal of this project was to create a functional prototype using Scratch for Arduino and basic electrical components. Table 1 shows the components used for the implementation. The Arduino Board, Model UNO, and a computer were also used.

The device presents 2 inputs: a thermistor to register the player's skin temperature and a button to log punches on the pad.

Additionally, it has 4 outputs: a green and a red LED, a buzzer, and the computer monitor. The LEDs are used to signal in-target and off-target punches, the buzzer gives the pace, and the screen presents the instructions and workout statistics.

Table 1. Electrical components used to build "Punching Pad" device

| COMPONENT | QUANTITY |
|---|---|
| Red LED | 1 |
| Green LED | 1 |
| Button | 1 |
| Jumper Cables | 13 |
| 100 µF Capacitor | 1 |
| P2N2 Transistor | 1 |
| Potentiometer | 1 |
| Buzzer | 1 |
| Thermistor | 1 |
| 1k Ω Resistor | 2 |
| 330 Ω Resistor | 1 |
| 10k Ω Resistor | 3 |
| 560 Ω Resistor | 1 |

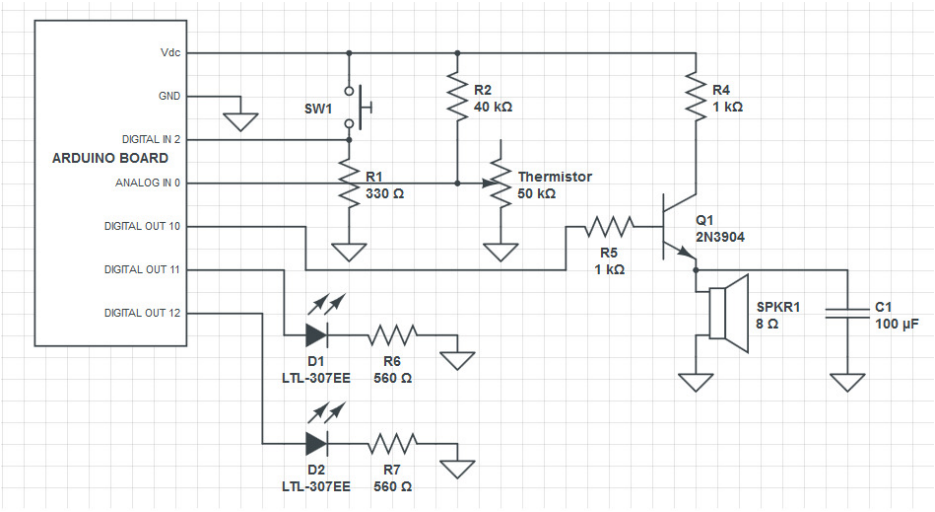


Figure 1. Electric diagram

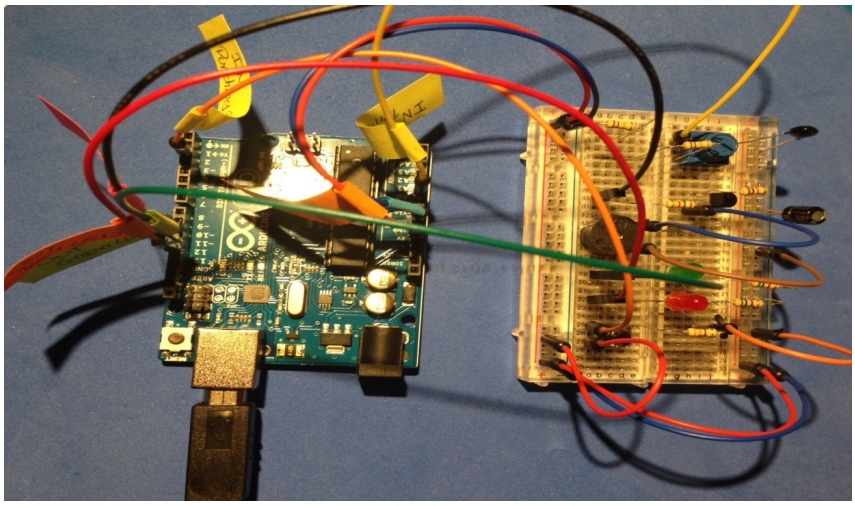

Figure 2. “Punching Pad” circuit assembly

To start the game, the user clicks on the green flag that appears on the screen and then presses the space bar to read the instructions. Now the player can hit the punching pad; the buzzer will give the punching pace. The speed increases after every 20 punches. If the punching pad is hit on time, a green LED will light up. Otherwise, a red LED will.

Initial and final temperatures were recorded to calculate temperature based on the voltage measured in the thermistor. The program will estimate the player’s heart rate based on the temperature. The workout statistics will be displayed in the screen: good punches, bad punches, initial temperature, final temperature, initial heart rate, and final heart rate.

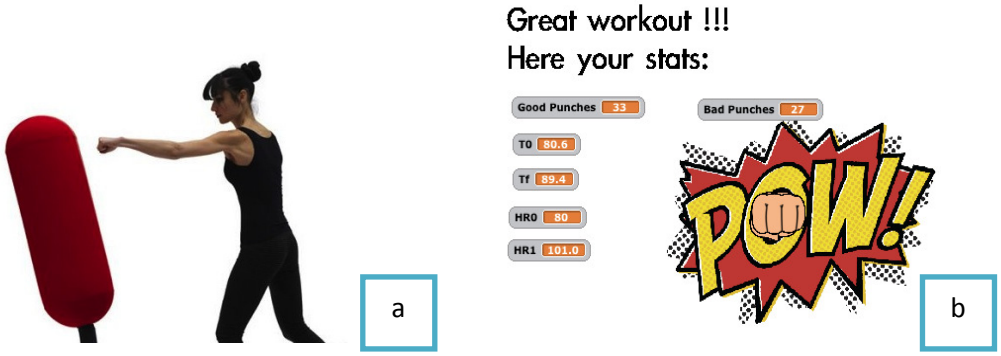


Figure 3. “Punching Pad” (a) instructions and (b) fitness statistics.

**Exergaming and Outreach**

The “Punching Pad” device was used in outreach sessions, hosted at Purdue University, to help 50 middle and 20 high school students to visualize programming applications. Attendees were first introduced to computer and information technology fields. After this short presentation, they were familiarized to Scratch and guided through the creation of a simple animation. Finally, students had the opportunity to interact with the device; they also

had the chance to explore the content of the program, made questions about the circuit and gave informal feedback about the device.

Table 2. Outreach session agenda

| **Agenda:** | |
|---|---|
| 5 minutes | Introductions & IT Background |
| 15 minutes | Dancing Girl Scratch Activity |
| 15 minutes | Group Scratch Activity |
| 10 minutes | Punching Pad Contest |
| 5 minutes | Session review; questions & answers |
| 5 minutes | Session survey |

## Conclusions

The device was well received by middle school students; they were immersed in the activity and were curious about how the program worked. On the other hand, high school students completed the "Punching Pad" activity without any kind of engagement; they were more interested in other aspects of the presentation. Nevertheless, further studies are necessary since the survey applied at the end of the sessions was not directly related to the device use.

Scratch for Arduino programming and circuit implementation do not require a great amount of previous knowledge. The device circuit implementation could be easily replicated or modified. The intuitive nature of Scratch makes novice programmers' experiences pleasant and rewarding. The Arduino board circuit implementation does not represent a source of frustration for the user; it is a versatile tool that could be used in different activities and settings.

Even though Scratch for Arduino, in collaboration with the Arduino board, are easy and intuitive to understand, to gain real programming skills students must have a prolonged interaction with these tools. The time allowed for the outreach sessions, one hour, is insufficient to create meaningful programming knowledge that could be extrapolated to other programming languages or present meaningful results at the time.

## References


[1] Dietz, W. (2009). Innovative Childhood Obesity Practices. *Innovations in Addressing Childhood Obesity. Hearing Before Subcommittee on Heath of the Committee on Energy and Commerce House of Representatives,* 7-22.

[2] Klein, M., & Simmers, C. (2009). Exergaming: Virtual Inspiration, Real Perspiration. *Young Consumers*, *10*(1), 35-45

[3] Resnick, M., Maloney, J., Monroy, A., Rusk, N., Eastmond, E., Brennan, K., et al. (2009). Scratch: Programming for Everyone. *Communications of the ACM*, *52*(11), 60-67.

[4] Booth, T., & Stumpf, S. (2011). *End-User Experiences of Visual and Textual Programming Environments for Arduino*. Berlin: Springer-Verlag.

[5] Kato, Y. (2010). Splish: A Visual Programming Environment for Arduino to Accelerate Physical Computing Experiences. *Eighth International Conference on Creating, Connecting and Collaborating through Computing*, 3-10.

**Biographies**

ALKA HARRIGER joined the faculty of the Computer and Information Technology Department (CIT) in 1982 and is currently a rofessor of CIT. She obtained her M.S. in Computer Science from Purdue University, West Lafayette, IN, 1981. Since October 2013, she has been co-leading with Prof. Brad Harriger the NSF-ITEST funded TECHFIT (Teaching Engineering Concepts to Harness Future Innovators and Technologists) project. Professor Harriger's current interests include outreach to K-12 to interest more students to pursue computing careers, and applying IT skills to innovating fitness tools. Prof. Harriger may be reached at harrigea@purdue.edu.

MAYARI SERRANO is currently a graduate research assistant in the Department of Computer and Information Technology at Purdue University. She earned her B.S. degree from the Army Polytechnic School, Quito, Ecuador. She is completing her M.S. in Computer and Information Technology at Purdue University. Mayari is currently working in the NSF-ITEST funded TECHFIT project. Her interests include foster STEM enthusiasm, and technology innovation. Mayari may be reached at serranm@purdue.edu.