Using Flowchart Programming to Create Exergames

Alka Harriger
Purdue University
harrigea@purdue.edu

Mayari Serrano
Purdue University
serranm@purdue.edu

Abstract

Physical, mental, and social health benefits by the regular practice of physical activity. In the United States, obesity has reached epidemic levels; the excessive use of technology has contributed to the increasingly sedentary lifestyles of people. Exergaming activities demonstrate how technology could be used as an instrument to reduce the impact of this disease. One can purchase commercial, technology-based exergames such as Nintendo Wii Fit or Xbox Kinect games; however, the authors developed a custom exergame using Phoenix Contact's Nanoline microcontroller and nanoNavigator software flowchart. The beauty of the flowcharting software is that people who have no or little programming experience can easily understand its structure. The software is downloadable for free and includes a simulator that allows the developer to determine the accuracy of the functionality of the exergame.

Introduction

In less than 30 years, childhood obesity in the United States has more than doubled [1]. Sedentary activities have gradually replaced physical ones, and childhood obesity has been steadily increasing [2]. Raising physical activity in children's lives has become a challenge because of today's excessive use of media, including television, phones, video games, computers, and tablets. Incorporating exergaming activities into children's lives could be the answer to this crisis. Exergames provide the capability to link exercise and video games into a single activity [1]. The authors are taking advantage of this concept to promote exercise and tie it to learning programming to foster fitness and STEM enthusiasm.

Technology could enable teachers' capabilities to design proper education environments in which the students could reach autonomy during skill and fitness learning [2]. In the right environment, children could be able to not just program but also to acquire planning and problem-solving skills. Acquiring these skills will depend on the amount of exposure children have in the field of programming [3].

Practicing programming concepts and strategies in a game setting has been shown to encourage novice programmers to accomplish more than the bare minimum; in fact, they are willing to work on the enhancement of their scores [4].

Developing a toolkit that could be easily assembled and programmed in a school setting could greatly benefit the meaningful learning of programming basics by children. Providing a tool that could minimize novice programmers' deficiencies and difficulties will contribute to this goal.

Exergame Design Using nanoNavigator Software

The exergame described in this paper was designed using nanoNavigator, a free flowchart programming tool downloadable from the Phoenix Contact website. It uses flowchart programming to describe the program logic.

Figure 1 shows the "Programing Blocks" that are available in nanoNavigator:

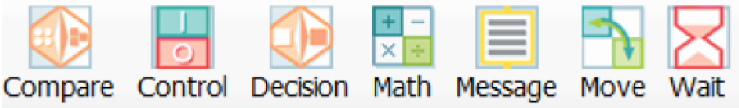


Figure 1. Programing blocks menu

1. *Compare Block:* can make a numerical comparison (<,=,>,>=,<=, and IsEven). The flow follows one of the two exit pathways based on the result of the comparison.
2. *Control Block:* is capable of setting bits On or Off, and doing internal timer or counter operations.
3. *Decision Block:* evaluates a Boolean expression as *true/false*. This block has one entry point and two exit points (Yes/No), based on the Boolean decision. The AND and OR "Operator" in the dialog box permits adding a second Boolean condition to the expression to create compound decisions. Expressions are evaluated from top to bottom.
4. *Math Block:* provides the ability to perform math functions on data within the flowchart. It performs basic math operations (addition, subtraction, multiplication, and division) on similar and different data types.
5. *Message Block:* nanoLC can display messages to the user. The messages can contain static text, dynamic values from any data item, and data entry requests.
6. *Move Block:* moves data from one location to another. Doing a move from one data item to another moves the data of the first data item into the second data item. The value of the first data item is preserved, so the move actually performs a copy operation.
7. *Wait Block:* inserts a wait period in the execution of the flow chart.

This tool also allows the user to simulate the script without having the electrical components assembled or connected. The simulator allows the user to watch the movement of the flow of control through the flowchart's logic as well as see the values of all data elements referenced in the program.

Exergame Hardware

The Phoenix Contact Nanoline microcontroller was used as the processing hardware for the pushup exergame described in this paper (see Figure 2.) It includes six digital input and four

PNP digital output channels. Additional channels could be incorporated using extension modules if needed. The pushup exergame incorporated the communication module, NLC-COM-GSM–2701344, which allows GSM communication to enable the user to control part of the game play via text messaging from a mobile device.

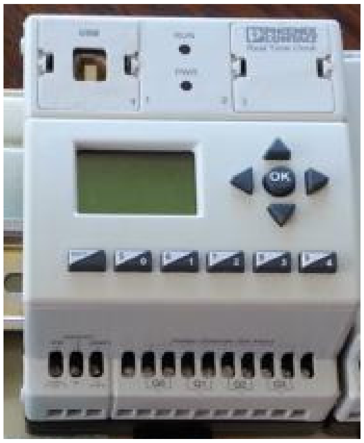


Figure 2. Nanoline microcontroller base unit

While the microcontroller does the processing work for the exergame, several additional electronic components are needed to handle the input and output functions.

- Input
  - Proximity sensors
  - SMS messages
- Output
  - Green light
  - Red light
  - Yellow light
  - Buzzer
  - nanoNAVIGATOR screen
  - SMS messages

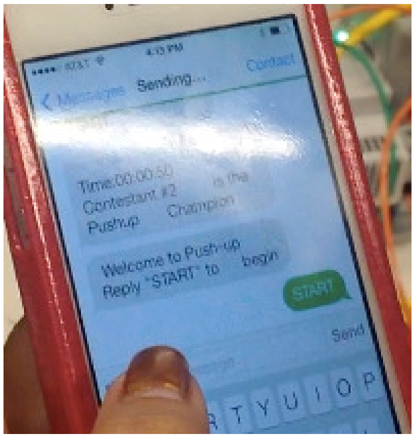
Lights provide visual feedback to the user, an alarm provides auditory feedback, and a sensor detects accuracy of performance. Table 1 identifies all of the hardware required to create the pushup exergame.

Table 1. Electrical components used to build "Push-Up Competition" device

Component	Quantity
Nanoline microcontroller Base	1
Red light	1
Yellow light	1
Green light	1
Communication module - NLC-COM-GSM – 2701344	1
Antenna - PSI-GSM/UMTS-QB-ANT – 2313371	1
SIM card	1
Proximity sensor	2
Buzzer/alarm	1
Cellphone	1
Wires (blue and orange)	3 meters

Exergame Implementation

The "Push-up Competition" exergame was designed using Phoenix Contact's Nanoline microcontroller and nanoNavigator software. An SMS message is sent by the microcontroller to all enabled cellphones with instructions asking the prospective player to reply with the command "START" to begin the game (see Figure 3.)



Time:00:00:50 Contestant #2 is the Pushup Champion

Welcome to Push-up Reply "START" to begin

START

Figure 3. Sending start message to the Nanoline GSM component

The two contestants then place themselves in a "push-up" position and wait for a buzzer to set the exercise pace. The lights will turn on when the push-ups are done correctly (see Figures 4 and 5).

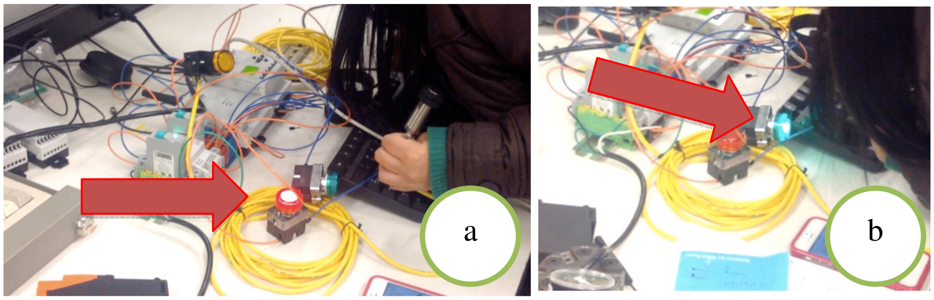


Figure 4. Light signals correct push-up for Player 1 (a) and Player 2 (b)

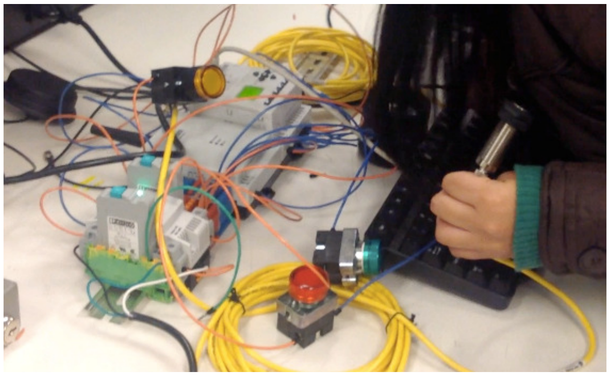
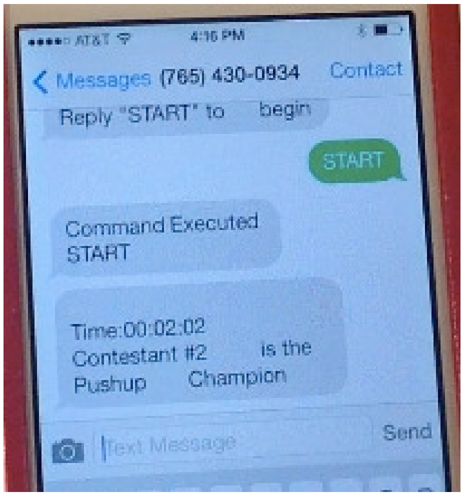


Figure 5. No light signals incorrect push-ups

If the contestant keeps the pace, one "push-up point" will be added to the total score. If the contestant fails to keep the pace, one "push-up point" will be subtracted. The first contestant to complete 10 well-performed push-ups wins. An SMS message will be sent to all enabled cellphones identifying the winner and the time it took to complete the 10 well-performed push-ups (see Figure 6.)



(765) 430-0934

Reply "START" to begin

START

Command Executed
START

Time:00:02:02
Contestant #2 is the
Pushup Champion

Figure 6. Winner message sent by the Nanoline microcontroller

The “Push-up Competition” device, along with other exergames, has been used in many outreach sessions. Figure 7 shows two middle school students competing with each other as their classmates watch. The players had the chance to interact with the device and the software. It gave the players and observers a concrete and fun example of the value that programming brings to real life.

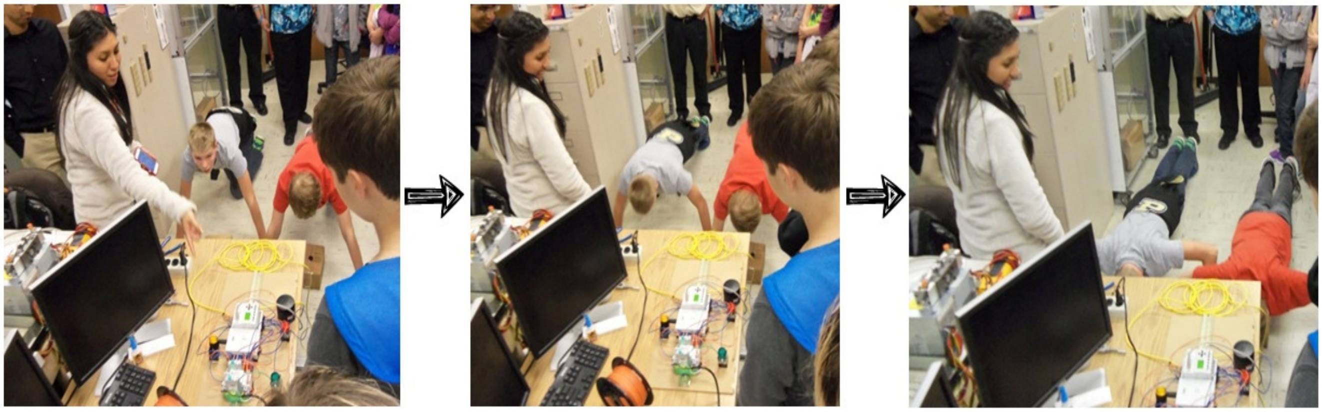


Figure 7. Middle school students playing with “Push-up Competition” Exergame

Exergame Flowchart Program

When developing a flowchart program with nanoNavigator, thinking through the process, identifying the input and output data elements, and even spelling out the messages before the user starts coding will significantly speed up the coding. The Edit Data Items option allows the user to customize the input terminals, output terminals, flags, counters, timers, registers, and message items to be used by the program. When using the simulator, these are the data elements that will appear in the watch window. The Edit Message Content option is where the user formats the messages that the program may display at some point during program execution. After these preliminary steps have been completed, the user is ready to begin programming.

Just like in traditional languages, any counter or timer variables should be initialized at the start. The push-up exergame uses a timer and two counters, so these data elements, along with the GSM module are initialized. (See Figure 7.)

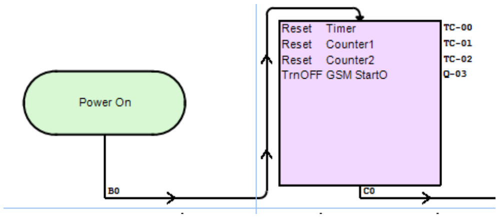


Figure 7. Initializing timer, counters, and GSM

Because the GSM module initiates the start of the game, the next code block checks connectivity to the GSM network, sends a message to the approved cell phones, receives the “START” message, and displays the instructions message. See Figure 8.

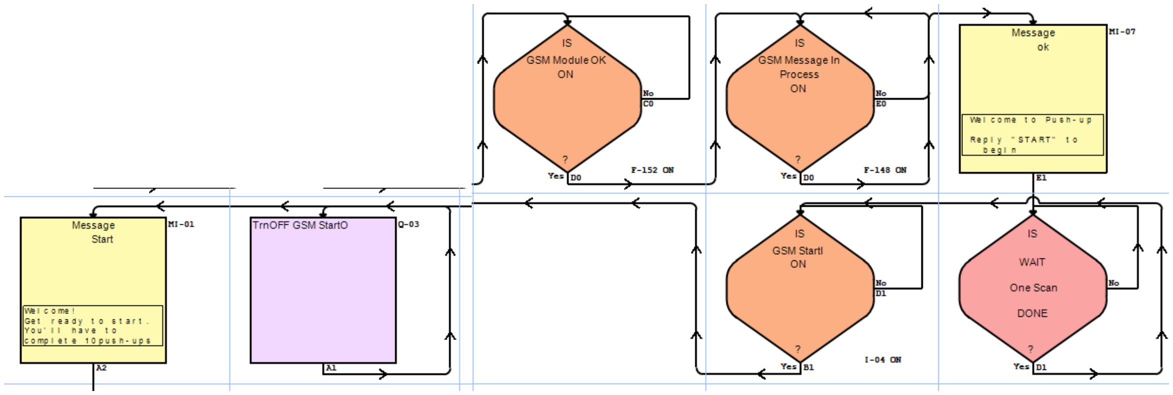


Figure 8. GSM-related code block

Once a player texts START to begin the game, the next block of code sets up the timer, starts the buzzer (for audio pacing), and turns off the signaling lights. Then the sensors are used to determine whether the player performed the push-up correctly at the set pace, and the appropriate counters are incremented or decremented. See Figure 9.

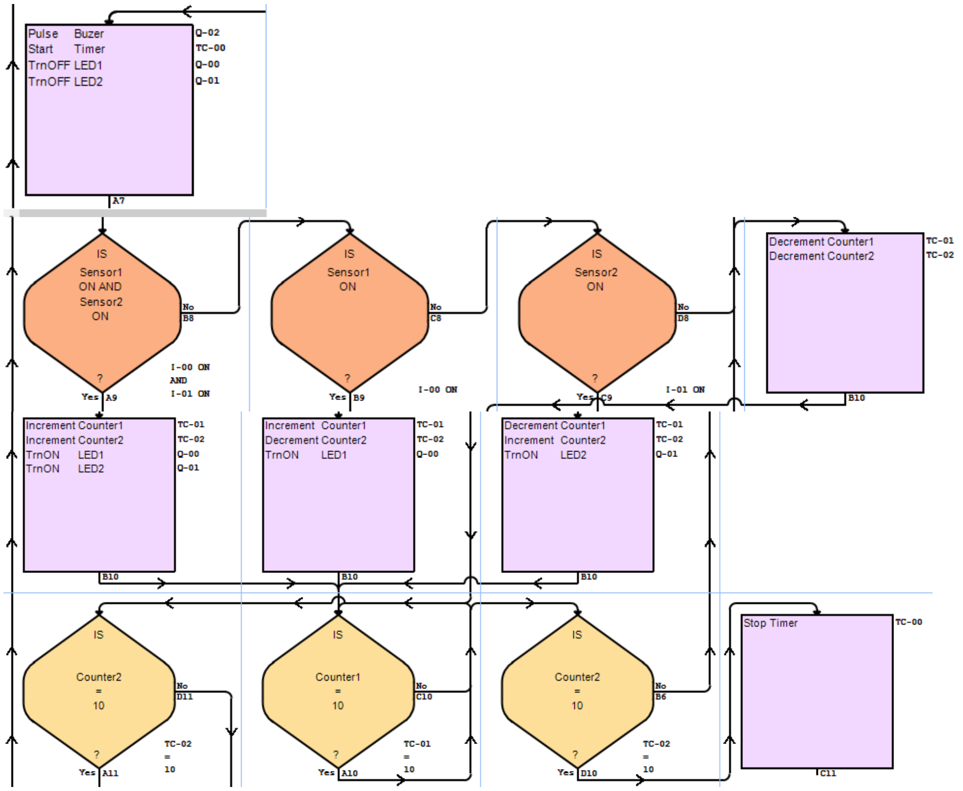


Figure 9. Code to handle player performance

Finally, when a winner is detected, the next code block stops the timer, sends the message about the winner, checks the GSM network, sends a restart message to the assigned cellphones, and restarts the game or finishes it depending on the response message received. See Figure 10.

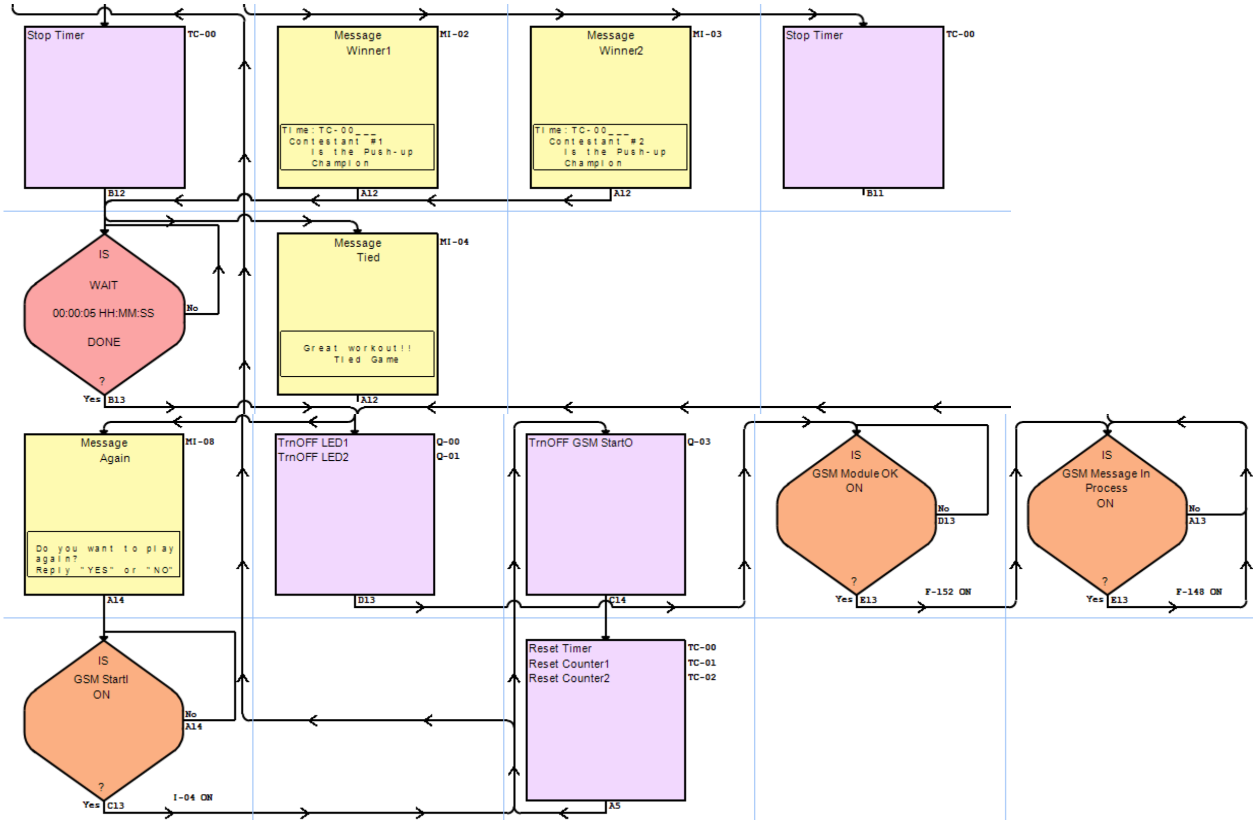


Figure 10. End of game processing code block

Conclusions

Based on observations and informal survey results, the push-up exergame succeeds in creating STEM interest in children, along with fitness awareness. The prototype could be used to inspire young people to exercise and also to help spark interest in engineering innovation.

The "Push-up Competition" device was designed as a fitness activity for middle school students through adults. The two-player feature made the competitive nature of the activity more appealing for the participants. This activity is seen as an entertaining game and not just as a boring, solitary fitness activity. It should be noted that in order to win the game, it is just as important to keep pace with the buzzer, encouraging better physical fitness. The components used for this project could be easily rearranged to create different fitness activities, depending only on the placement of the proximity sensors.

From the context of computing, flowchart programming has been an intuitive tool to work with in outreach sessions to middle school and high school students. The added benefit of providing a tangible product, the exergame, makes it an appropriate tool for introducing novices to programming in a way to maintain their interest in programming.

Future Direction

Further studies are necessary to determine if flowchart programming will help novice programmers migrate to more complicated programming languages. Another Purdue graduate student is pursuing this area for his master's thesis.

The exergame described in this paper will continue to be used to demonstrate simplicity of programming; however, further enhancements are recommended to improve the usefulness of the device:

- Modify the program to provide workout statistics to users so that they can actually be stored and organized to maintain reliable fitness records.
- Modify the device and related software for mobility and transportability. In its current state, the device does not lend itself to being moved across a large campus easily. However, making modifications to support mobility would enable its use in different locations, further extending its impact.

References

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

[2] Thornburg, R., & Hill, K. (2004). Using Internet Assessment Tools for Health and Physical Education Instruction. *TechTrend, 48*(6), 53-70.

[3] Hattie, J. (2013). *Visible Learning: A Synthesis of over 800 Meta-Analyses Relating to Achievement*. London: Routledge.

[4] Bromwich, K., Masoodian, M., & Rogers, B. (2012). Crossing the Game Threshold: A System for Teaching Basic Programming Constructs. *CHINZ '12*, 56-62.

Biographies

ALKA HARRIGER joined the faculty of the Computer and Information Technology Department (CIT) in 1982 and is currently a professor 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. Alka's current interests include outreach to K-12 to interest more students to pursue computing careers and applying IT skills to innovating fitness tools. She 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 an active team member of the

NSF-ITEST TECHFIT project. Her interests include fostering STEM enthusiasm and technology innovation. Mayari may be reached at serranm@purdue.edu.