# Interactive Pac-Man Calorie Counter

1st Krishna Bavana
*B.S, Dept. of Computer Science*
*University of Illinois at Chicago*
Chicago, United States
kbavan2@uic.edu

2nd Dwaragesh Sivakumar
*B.S, Dept. of Computer Science*
*University of Illinois at Chicago*
Chicago, United States
dsivak3@uic.edu

3rd Madhava Sai Yamike
*B.S, Dept. of Computer Science*
*University of Illinois at Chicago*
Chicago, United States
myamik2@uic.edu

4th Richa Bhujbal
*M.S, Dept. of Biomedical Engineering*
*University of Illinois at Chicago*
Chicago, United States
rbhujb2@uic.edu

5th Dr. Hananeh Esmailbeigi
*Clinical Associate Professor*
*University of Illinois at Chicago*
Chicago, United States
hesmai2@uic.edu

6th Dr. Joseph Hummel
*Clinical Associate Professor*
*University of Illinois at Chicago*
Chicago, United States
jhummel2@uic.edu

***Abstract*—We have been conditioned to think about exercise being the main component of any weight loss journey. But in truth, research has shown that for years exercise, while great for health,is not all that is necessary for weight loss. This is because it accounts to only a small part of total calorie loss and it is difficult to create significant calorie deficit solely through exercising. A study from 2009 has shown that people seem to increase their calorie intake through food consumption after exercise because they believe they burnt a lot of calories or because of hunger. A different review from 2012 showed that people generally fail to accurately estimate how much energy exercise burned and eat more after workout [1]. In regards to this, in this final project we have built an Interactive Calorie Counter which allows the users to know the balance between how to burn calories and how to intake the right amount of calories. This device and user interface allows the users to influence their food habits thus enabling public-health obesity to prioritize fighting the over-consumption of low-quality food and improving the food environment.**

***Index Terms*—Healthy Lifestyle, Sensors, Food Modes, Calories, Score**

## I. Introduction

People tend to struggle with finding the right balance between knowing how to burn calories and how to intake the right amount of calories. This imbalance caused an influence in acknowledging the perception of working out more than considering how to control food intake more. Therefore, Pac-Man Calorie Counter will serve as a great starting point that will influence people to eat healthy, where they are enabled to choose a healthy, junk, or mixed mode, choose what to eat and what not to eat (i.e. healthy food items vs. junk food items) within the respective mode, choose to 'pick up' items (i.e. height blocks and danger blocks) within the respective mode, and monitor calories, lives, and score at the end.

## II. Methodology

The development process of the prototype was done with the support of the following components - short flex sensor, capacitive touch sensor with MPR 121 board, and FireBeetle Board-328P with BLE4.1 [Central and Peripheral]. The short flex sensor is used to open and close the mouth of the Pac-Man, serving as user input that will be prompted once the user chooses the food mode. This sensor will be placed on the jaw of the user, and can be monitored on the Arduino Serial Monitor to receive pertinent data.

On that note, the food modes were specified by the capacitive sensor via the MPR 121 board to detect events that occur on a smart pad that communicated with the board. Specifically, the electrodes of the wires connected from the headers of the MPR 121 board will be placed onto the smart pad, masked with copper tape. The heart of the prototype revolves falls under how the Bluetooth communication is made with the two FireBeetle boards - the Peripheral board [with headers] will be worn by the user and prompt user input respectively; whereas the Central board [without headers] will be connected with a Micro-USB cable that will receive user input in order to use it on the two code bases: Arduino and Processing. Ultimately, the integration of these components had helped create the Pac-Man game by receiving pertinent to apply it onto the UI.

## III. Implementation

### A. Device Design

To explain the purpose of the device design, and its structure of construction, it is important to understand how each components work from an electrical engineering perspective. Firstly, the flex sensor is used to detect flexing and bending in one direction. When the flex sensor is bent, the conductive particles are farther apart when measured with a multi-meter; when the flex sensor is not bent, the conductive particles are closer together when measured with a multi-meter [2]. Secondly, the capacitive touch sensor is used to detect a medium with conductive or di-electric difference from atmosphere [3]. This will be used to ensure what pin the user touched, or in other words, identify the touch event that is registered. In all, this prototype will be built with the inspiration of a game interface (i.e. a game watch) that allows users to do wonders in the Pac-Man Calorie Counter UI.

### B. Hardware

Components to implement Pac-Man game:

- FireBeetle Board-328P with BLE4.1 x 2
- MPU-6050 (GY-521) Accelerometer*
- MPR 121 Board with Headers for Touch Sensors
- Short Flex Sensor
- Resistor (3.3 KOhm) x 1
- Micro USB Interface Cable
- Rechargeable Lithium Battery
- Wire and Wrapping Tool

For the device hardware, the aforementioned components have been used based on the requirement of the game design. The sensors are connected to the bluetooth enabled board wirelessly to obtain a compact and wearable device that can be worn on hand by the user.

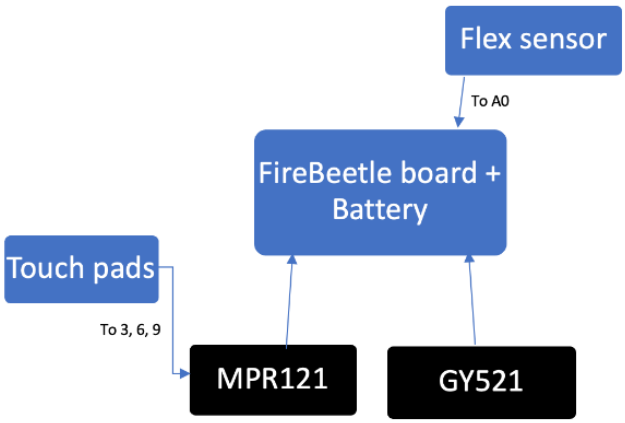


Fig. 1. Block Diagram of Hardware Connections

### C. Software and UI Implementation

The Pac-Man - Calorie Counter project, two different code-bases were used - Arduino and Processing. In order to retrieve data from the user, Arduino provided the team to gather the data for Short Flex Sensor, Capacitive Touch Sensor via MPR 121 board, and lastly, MPU-6050 (GY-521) Accelerometer*. Fast forward, the data stored in a buffer in Arduino had been put to use to establish the logical component behind an interactive user interface using Processing.

Retrospecting back to the brainstorming sessions, the two core values that were unanimously decided to use for the coding portion of the project was to follow functional decomposition and coloration. Over a period of time, this choice benefited our team significantly because there were many components defined for the Pac-Man game - and adding every minute and major component into a big function would have brought more hassle. In Processing, draw() was used to call all the functions in order to complete important actions - such as setting initial points, defining polygon size, etc. As weeks progressed, the development process of the code had brought wonders to the Pac-Man game - this included Food Modes Setup, Addition of Pac-Man Actions, Enable Pac-Man to Eat Food, Size Increase of Avatar, Size Decrease of Avatar, and Height Increase of Exit Door.

Despite the fact that there were many challenges that had to be crossed to develop this game, the ultimate challenge that our team had was how to figure out the Addition of Food Items in a row of a path followed by the Pac-Man. The root cause behind this challenge evolved from the arrangement of the food items, as seeing that it had to be random in every row of a path. To resolve this issue, our team moved away from the idea of using different functions and moved towards a tracking approach that helped identify the position to release the food items in a circle format as well as danger blocks in a circle format. This tracking approach ultimately benefited our team to keep track of each and every variable that was used to develop the game, i.e X-Coordinate, Y-Coordinate, Width, Height, Angle One, and Angle Two.

As mentioned before, the other important choice was to follow the idea of lighter coloration to decorate the user interface. The purpose of pursuing this idea was to not harm the eyes of the player, no matter how long they play; This ideology was used in a consistent manner for all four screens of the game.

## IV. Results

Hence, we have successfully designed the Pac-Man Calorie Counter. Figures (a) and (b) below show the final designed device. The UI of the game is shown below. It starts with

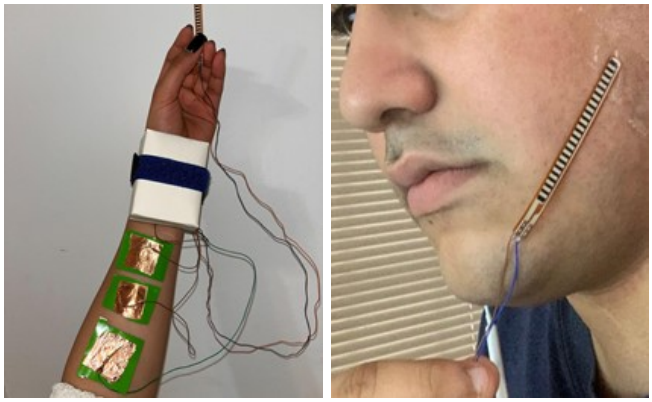

Fig. 2. (a) Pac-Man Calorie Count Device (left) (b) Flex Sensor Pos. (right)


Fig. 3. User Interface Screens for Pac-Man Calorie Counter

a welcome screen; It goes to the mode selection screen; It prompts the user to the play screen to play the game; It completes the game with end scene. This device has been developed for different requirements collected through research which has led to a low-cost design with interactive functions as well as ease of use. Above all, it enables the user to keep a track of their calorie count for applications in various dieting routines. These aforementioned specifications make Pac-Man - The Calorie Counter stand out.

## V. Conclusion

In conclusion, with the intention to serve as a torch for people who require the light to change their mindset towards their food intake, the Pac-Man - Calorie will serve as a great stepping stone to promote healthy lifestyles. The advantages for it are as follows:

- Cost-effective with basic components which are cheap and easily available in the market, Portable for more convenience
- Accurate and synchronized experience
- Can be customized according to the needs of a specific user's diet as well as various parameters like heart rate, SpO2, etc.

For further development, the system can be modified by incorporating sensors to motivate users for walking to burn the calculated calories. Music can also be included in the system which will have a positive effect on the user. Furthermore, more game modes can be added in different ways to include types of food items customized to a specific user's regular diet plan.

Ultimately, this project has helped us evolve as programmers because our team had a great understanding on how to gather data from different sensors at the same time, and more importantly, achieve a clean user-interface.

## VI. Acknowledgment

The authors would like to express their sincere gratitude towards the coordinators of BME/CS 479. We extend a special thanks to Dr. Hananeh Esmailbeigi and Dr. Joseph Hummel from The University of Illinois at Chicago. In addition to this, we would also like to extend heartfelt gratefulness towards Karla Salcedo Diaz and Kyle Coda for their continued abatement and guidance.

## References

[1] "The science is in: exercise won't help you lose much weight". Retrieved from: The Vox; J. Belluz and C. Haubursin; January 2, 2019; https://www.vox.com/2018/1/3/16845438/exercise-weight-loss-myth-burn-calories

[2] "Arduino Flex Sensor Tutorial: How to Use a Flex Sensor". Retrieved from: DFRobot; April 3, 2018; https://www.dfrobot.com/blog-889.html

[3] "Waking up a capacitive touch-sensing device with an MCU peripheral". Retrieved from: Embedded; July 27, 2011; https://www.embedded.com/waking-up-a-capacitive-touch-sensing-device-with-an-mcu-peripheral/