No Pain No Game: A Mobile Approach to Childhood Obesity Prevention
Sean Bromage, Navid Amini, Majid Sarrafzadeh
Computer Science Department
University of California, Los Angeles
{sbromage, amini, majid}@cs.ucla.edu

ABSTRACT
This paper details the development of an end-to-end wireless system that provides children with reward based incentives in exchange for increased physical activity. An Android application provides the children with dynamic information provided by an expert in nutrition or behavior and tracks how many steps they have taken throughout the day via software that utilizes the phone’s accelerometer. Based on the amount of information the child has digested and the distance they have walked throughout the day, the child is awarded screen time and is educated on the subject of childhood obesity. The screen (television, video game console, computer, etc.) is controlled via a home automation hardware system connected to a web server. By using data from accelerometers exclusively, as opposed to tracking movement with GPS, the application can track the user’s activity with relatively small amount of computation and drain on the battery.

General Terms
Management, Measurement, Design, Human Factors

Keywords
Mobile Health, Childhood Obesity, Home Automation, Behavior Management

1. INTRODUCTION
Childhood obesity’s detrimental effects on the health and lifespan of an individual coupled with its far reaching grip on today’s youth have caused concern that approaches the level of a nationwide pandemic [1][22]. The battle against childhood obesity is one that can be fought with education and personalized feedback on how to live a healthier lifestyle [21][23]. This involves educating the children about nutrition and normal exercise routines [2], using television and video games as incentives [37], and also decreasing the overall time spent in front of the television [3]. Smart phones are quickly becoming a ubiquitous device that provides internet connectivity and mobility. It was found that 42% of the U.S. was already using smartphones by the end of 2009, with large growth forecasted for 2011 [24]. By utilizing these devices, information can easily reach a large user base. Personalized feedback can be provided by adapting pedometer software to phones, taking advantage of the fact that millions of people already carry accelerometers in their pocket on a daily basis.

2. IMPLEMENTATION
This project is implemented through an application developed for Android. The application incorporates open source software that has been written to detect steps with the phone’s accelerometer. The number of steps the child has taken and the number of calories he/she has burned are converted into “screen time”. More screen time can be accumulated by reading through information fetched by the Android application from a database of professional tips and advice. The application is able to send a message to a local machine, commanding it to enable a specific device, such as a television. The computer, using a wireless home automation hardware, enables the outlet to which the television or other entertainment appliance is connected. A timer is then started on the mobile phone, allowing the user to see how much time they have remaining. If the user’s time runs out, the application will automatically shut off the outlet.

The data flow of the system is shown in Figure 1. Accelerometer data is fed into the pedometer software in order to calculate the number of steps. Data is dynamically fetched from a database containing snippets of text relevant to health and obesity. When the child wishes to watch television or play a video game, he sends a message to his PC from the application. A web server on the local computer receives the message and wirelessly enable’s the television’s outlet, accordingly.
2.1 Android Application
The mobile application has five distinct utilities that are split into tabs. They consist of the Home, Pedometer, Learn, Watch television, and Progress tabs, which are described in more detail below. Each tab is implemented as an Android Activity, allowing them to perform different tasks concurrently since each Activity runs as its own process.

2.1.1 Login
The login screen is displayed when the application is first started. It briefly describes the purpose of the system and explains how the users can earn more television time. It then prompts them for their username which is used in a variety of situations throughout the user’s experience in the application. Because of the user-dependent nature of data storage procedure and the way the application communicates with the web server, the username for each individual should never change. The Login Screen is an Android Activity that is called with the expectation of a response. This means that the user will not be able to access any other parts of the application until they have given their username.

2.1.2 Home Tab
The Home Tab is a useful way to welcome the users and allow them to check how much screen time they have at any given time. It displays the username and the amount of time associated with it. The user can go to this tab at any time during their in-application experience to view their most up-to-date amount of screen time.

2.1.3 Pedometer
The Pedometer Tab is used to display useful information pertaining to the open source pedometer software that is utilized in this project [36]. When the user navigates to the Pedometer Tab, a pedometer service is started on the phone in a separate process that actively listens to incoming data from the device’s accelerometers. The accelerometer data is run through an algorithm that detects individual steps. By launching the pedometer service in a separate process, the phone can detect steps even after the user navigates away from the Pedometer Tab, closes the No Pain No Game application, or locks the device’s screen. By using data from accelerometers exclusively, as opposed to tracking movement with GPS, the application can track the user’s activity with relatively little computation and drain on the battery. The application can currently track a user’s activity for an entire day without depleting the device’s battery. This is of course dependent on the type of device being used and will improve as processors and batteries become more efficient.

The user is able to change multiple settings for the pedometer software that increases the accuracy of the information displayed. They can change the sensitivity of the accelerometers in the range of very low, low, medium, high, and very high. This allows the phone to more accurately recognize steps, allowing for precision among a wide range of devices and placements of the device on the body, e.g. a device in a purse may need to be more sensitive than one in a front pocket. The user is also able to set the operational level of the pedometer service; they can decide if they would like the pedometer service to only run when the phone’s screen is on in order to watch results and save battery, always run the service in the background (the default), or only start the service when the device’s screen is locked to prevent it from running when the device is in use.

The Pedometer Tab can display information in both imperial and metric units, allowing for a seamless back and forth between users of different nationalities or for users with differing preferences.

During the setup of the application, the users must input their step length, body weight, and exercise type. The step length is the distance between their feet in an average step. This is used to calculate their total distance [6,7,8,9], walking or running speed [10,11,12,13], and calories burned [14,15,16]. Their body weight and exercise type, either running or walking, are also used in the calorie calculation. When the user submits his steps, the application logs how many steps were taken and how many calories were burned on a specific day/date. This information is used to create bar graphs in the Progress Tab that allow the user to track their activity over a certain period of time. In the experiments performed on the system thus far, converting the number of calories burned by the user directly into minutes of screen time has provided a reasonable amount. This conversion rate can be modified, however, should the owner, i.e. the parents, wish to award more or less screen time per calorie to the user.

2.1.4 Educational Content
The Learn Tab provides information on topics such as healthy diets and normal exercise routines. Snippets of text are dynamically fetched and displayed in a web interface. This allows the information to be updated and changed by an expert in the field of nutrition, exercise, or behavior without having to update the application itself. Also, since it is loaded in a web view, a child can easily click a link that will take him to a mobile website containing more information for them to benefit from. When the child fetches new information, a new snippet will be displayed and he is awarded a small amount of screen time.

During the setup of the application, the users must input their step length, body weight, and exercise type. The step length is the distance between their feet in an average step. This is used to calculate their total distance [6,7,8,9], walking or running speed [10,11,12,13], and calories burned [14,15,16]. Their body weight and exercise type, either running or walking, are also used in the calorie calculation. When the user submits his steps, the application logs how many steps were taken and how many calories were burned on a specific day/date. This information is used to create bar graphs in the Progress Tab that allow the user to track their activity over a certain period of time. In the experiments performed on the system thus far, converting the number of calories burned by the user directly into minutes of screen time has provided a reasonable amount. This conversion rate can be modified, however, should the owner, i.e. the parents, wish to award more or less screen time per calorie to the user.
2.1.5 Screen Time
The Watch Television Tab allows the users to enable and disable an appliance directly from their mobile device. It displays a “Start” and “Stop” button along with a timer indicating how much screen time they have remaining. When one of the buttons is pressed, an HTTP POST message is sent to the URL “\{user\}npng.dyndns-server.com/{command}”, where \{user\} is the username and \{command\} is either ‘start’ or ‘stop’, depending on which button was pressed. The static domain is mapped to the user’s local machine using a dynamic DNS service. This service detects the IP address of the machine it is running on and updates a server with the latest changes so the application will always be able to reach the user’s local machine. The dynamic DNS service and web server used to listen for commands and control the home automation hardware are described below in Section 2.2. Once the user runs out of screen time, the outlet is automatically shut off and users are informed that they are out of time. Each tab in the application runs as its own process. This allows the Watch television Tab to continuously update its countdown timer when the user navigates away from the tab.

2.1.6 User Progress Display
The Progress Tab is used to provide a meaningful representation of information to the user. By providing personalized feedback based on what they do, the application can encourage the user to continue with their active lifestyle. Everything that happens within the application is logged, included the number of steps taken, the number of calories burned, and the number of minutes of screen time used. The data is stored with the date on which they occurred. The Progress Tab organizes this data into bar graphs that display the aggregate of entries per day. The user can choose between seeing the number of steps they have taken per day, the number of calories they have burned per day, or the amount of screen time they have used per day. They can also choose how many days worth of data they would like to see, ranging from 7 days to a month in week-long increments. The data is stored in CSV files on the mobile device’s memory and is dynamically fetched depending on which chart the user would like to see and for how many days in order to improve efficiency.

2.2 Home Automation
Controlling the outlet that the screen is plugged into is achieved by using wireless home automation hardware. The hardware is controlled by a web server that runs constantly on the user’s local machine.

2.2.1 Hardware
The hardware used in this project is produced by Insteon, a division of SmartLabs Inc. The following is a list of hardware needed for the project:

<table>
<thead>
<tr>
<th>Device</th>
<th>Manufacturer</th>
<th>Quantity</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>PowerLinc Modem</td>
<td>Insteon</td>
<td>1</td>
<td>$80</td>
</tr>
<tr>
<td>LampLinc</td>
<td>Insteon</td>
<td>1</td>
<td>$50</td>
</tr>
<tr>
<td>USB-Serial Adapter</td>
<td>Many</td>
<td>1</td>
<td>~$15</td>
</tr>
</tbody>
</table>

The PowerLine Modem attaches to the local machine via the USB to serial adapter. The adapter is not needed if the user’s machine has a serial interface, a feature that is increasingly uncommon among modern PCs. The PowerLine Modem is used to control the LampLinc. The desired home appliance is plugged into the LampLinc, which controls the flow of electricity through it depending on the commands it receives from the PowerLine Modem. The PowerLine Modem receives serial commands from the computer and then communicates with the LampLinc over a mesh network consisting of wireless communications sent over RF and messages sent over the house’s power infrastructure using...
the X10 protocol. This system was able to be realized relatively cheaply with a price less than $150, making it accessible to a wide user base.

2.2.2 Web Server

The web server was written in Python. It implements a BaseHTTPServer and overrides the do_POST and do_GET functions in order to provide the desired functionality for No Pain No Game’s purposes. When the server is started, it listens for incoming requests on port 80 until the user kills it. When a request is received, it passes the URL that was requested to the appropriate function. In the case of turning a device on or off from the mobile device, the URL will be the static domain related to the user’s machine via a dynamic DNS service with a start or stop command appended to the end. This URL will be sent to the do_POST function where the command will be stripped from the URL.

Depending on the command, specific functions are called from a separate Insteon Python module developed for communicating directly with the serial interface. This module contains the Insteon specific commands that enable or disable the LampLinc controller. It uses PySerial, an open source utility supported by Python, to communicate with the machine’s serial interface via the Python script. When a command is sent to the PowerLinc Modem, it translates the desired action into the X10 and proprietary RF protocols and sends them to the LampLinc controller. The LampLinc then sends back an acknowledgement that it has received the command that includes its address. Since serial communication is often unreliable, the turn_on function is called in a loop until the appropriate acknowledgement is received. If after 50 iterations of the loop (these happen quite rapidly) the acknowledgement has not been received, then the server deduces that the home automation hardware is unreachable and sends a response to the mobile device. The mobile device responds to the message by prompting the user with helpful feedback about what is unreachable so the user can better troubleshoot any problems that might have occurred. It is important to ensure that the LampLinc is not plugged into an outlet controlled by a switch, since there will be two devices controlling the flow of electricity through the outlet and this can lead to confusion. The Insteon module attempts to open and communicate with specific COM ports on the user’s machine. The ports used must be identified during the set-up phase.

The do_GET function is implemented in the web server in order to add browser accessibility. When a URL is entered in a browser, it automatically sends a GET request to the domain. This request is received by the web server and sent to the do_GET function where it strips the command from the URL in the same manner as the do_POST function. Browser accessibility is built into the system in order to provide tools for testing the home automation hardware and dynamic DNS service without having to access them from the application itself. It also provides parents or the owners of the system to manually override the home automation hardware should they wish to turn the outlet on or off at any time.

The dynamic DNS is a free service provided by DynDNS.com. It allows the user to set up the appropriate domain name, given in Section 2.1.5, and download a program that detects the machine’s current IP address and updates DynDNS’ servers. When an HTTP request is sent to the domain, it is received by DynDNS and re-routed to the IP address associated with it, i.e. the user’s local machine. The request is then handled by the web server. A dynamic DNS service is necessary since the web server must reside on the user’s local machine in order for it to communicate directly with the serial interface via the Insteon module.

3. RELATED WORK

Other solutions for encouraging activity among children have taken the form of interactive video game systems. One such project uses accelerometers and pressure sensors to control soccer video games [4]. The system detects movements corresponding to running, passing, and shooting the ball. It also detects when the user is no longer standing and pauses the game until they once again become active rather than sedentary. This project rewards the user with video games for their activity and prevents them from being able to sit in front of the television for hours on end. This project is related to new video game systems that have recently been released, such as the Wii Fit and Xbox Kinect, two systems that use accelerometers and cameras, respectively, to control video games via the player’s movements.

Numerous applications in wireless health are emerging [26,27,28,29,30,31,32,33,34]. Systems that provide reward based incentives for children’s activity also exist. A wireless Body Area Network has been developed for wearable monitoring applications that are meant to avoid pediatric obesity. The network incorporates accelerometers and heart rate sensors in order to automatically recognize physical activity and the behavior states through evaluating multimodal sensing and interpretation [25]. Another system places a home automation interrupter between a video game console and a television. The interrupter is only disabled once it detects the operation of an exercise machine. The parents can set the system to only allow the use of the video game console once an adequate amount of exercise has been performed [35].

4. ONGOING AND FUTURE WORK

This project provides a useful application for encouraging activity and education for obese children. There are many areas of focus that can be improved upon in order to increase its effectiveness in the future.

4.1 Cheating Prevention

Many applications depend on the fact that the user’s intent is to use it properly in order to provide the most utility for them. This is certainly not the case with No Pain No Game. As a matter of fact, it is assumed that the user will make attempts to maliciously game the system and search for gaps that can be exploited since gaining more screen time is a large incentive. Cheating threat mitigation can take the form of anything from a lock box that prevents the child from unplugging the appliance from the home automation hardware to algorithms that can detect distinct users based on an accelerometer signature that is specific to their way of walking or running [17,18,19].
One type of cheating that the system is particularly susceptible to can be reduced to cases of false identity. This refers to scenarios in which the child gives their mobile device to another person in order to gain screen time for their activity, or when one child uses another’s device to enable a television in the same household without earning the screen time themselves. The threats to the system caused by these scenarios can be mitigated by designing a more intelligent step detection algorithm. If the application is able to determine the unique signature based on the movement of the intended user through machine learning techniques, referred to as Biometric Gait Authentication, it can distinguish between genuine activity committed by the user or false activity committed by a third party [19].

Since the pedometer software only takes accelerometer data into account, false positives have not been completely eliminated. The false steps can be caused by unintentional circumstances, such as the user riding in a bumpy vehicle, or maliciously, such as giving the device to another person in order to gain screen time for their activity, shaking the device instead of walking with it, or placing the device on an active pet. Work has been done to incorporate a shoe that contains an accelerometer and pressure sensor into the system. This will ensure that accelerometer data was only valid when the user was walking, as detected by the pressure sensor, thus eliminating false positives for shaking the device and riding in a vehicle. Validating accelerometer data can also be done via the device’s GPS module. By determining if the user is static or moving at too fast of a pace for walking or running, false positives for shaking in place and riding in vehicles can be reduced. However, this approach vastly increases the amount of power necessary for the application to function. This is why this approach was not pursued in the current implementation of this project. Other types of hardware and sensors can be incorporated into the system in order to improve its accuracy. By incorporating an unobtrusive heart rate monitor, the user’s activity levels and the amount of calories they have burned can be more accurately measured and calculated.

Many different media can be explored to convey information to the user. Currently, text and websites are used, but pictures, videos, and audio are all viable options for educational material. Videos can be used to instruct children on proper exercise techniques or teach them new ways of becoming active. Pictures could be used to depict meals with healthy ingredients and suggested serving sizes. Making the educational aspect of the application more fun for a child will increase the amount of information they consume and retain. By using a game-like format to expose children to new information, the process will be more enjoyable and feel less forced. One possibility is to incorporate short quizzes on information that is presented to them. Depending on the number of questions they correctly answer, they can earn varying amounts of extra screen time. Future iterations could even utilize speech recognition to recognize responses to audio questions asked of the user. By ensuring that the system dynamically fetches educational information, the administrators of the application can ensure that it is personalized to individual users. It also ensures that the information is always fresh so the user will constantly learn new things. By collecting the results of the quizzes, the quality of questions can be determined, leading to the design quizzes that are very effective. There are several pieces of information that can be collected from the user, such as quiz results, steps per day, calories burned per day, minutes of screen time used per day, their weight, and their body mass index. The more information on the user that is collected, the more personalized the feedback can be. For example, if one user is answering quizzes on nutrition correctly but is lagging in the number of steps they take per day, the information can shift to the importance of physical exercise or to suggestions for outdoor exercises.

A website that incorporates the user’s activity information can be used to offer further incentives that go beyond screen time. Coupons generated by the child’s parents can be added to the system to allow the child to spend the points they earn through being active on rewards such as weekly allowances, new video games, etc. A web interface can also be used to provide an interactive learning environment and progress feedback that goes beyond what mobile platforms are currently capable of.

The proposed system provides a platform on which novel ways to encourage exercise for obese children can be tested and measured. One such example that goes beyond simply using screen time as an incentive involves tracking when the user turns on the television and for how long. This data can be used to prompt the user to increase their exercise so they will have enough time to watch a particular show at the regular time. This system is used to modify the user’s behavior through feedback that is specific to their habits and to educate them about nutrition and normal exercise routines. It uses television and video games as incentives to become active and also decreases the overall time the child spends in front of the television.


[36] Levente Bagi. Pedometer Android Application. Copyright 2009, GNU General Public License