ARK: FPR Report

Ethan Miller (EE), Jacqueline Lagasse (CSE), Matthew Bolognese (EE), and Charles Klinefelter (CSE)

***Abstract*—Augmented reality is combining a real environment with computer generated sensory data, such as images and sounds, to create an experience that is different from pure reality. Some video games have been based off of augmented reality technology, but can only incorporate data captured by a camera into the augmented world. Augmented Reality Kick (ARK) is a gaming peripheral and Android app that allows data from a user's limb movement to be incorporated into an augmented environment without the use of a camera. A virtual reality headset holds a phone in front of a user's face while running the ARK app. The app itself displays what the rear camera can see, and overlays a virtual ball and net in a suitably flat area. Utilizing an inertial sensor on the foot, a user can then kick the virtual ball into the net to score points. Our device is intended to be used primarily for entertainment, but could also be used for athletic training, physical therapy and rehabilitation purposes.**

I. Introduction

THE core of this project is augmented reality (AR). AR is the dynamic response of a computer to its environment. Using computer vision or object recognition, a computer "augments" the world it sees, most commonly through overlaid sounds and visuals. A user requires goggles or a screen to use this technology, similarly to Virtual Reality (VR). These are not to be mistaken however, as VR completely constructs and controls the user's visual world. AR is constrained by the user's immediate surroundings and must process and react to the changes in them [1].

AR is of growing popularity among tech companies, particularly Microsoft, Google, Apple. While Microsoft has the standalone HoloLens product, Apple and Google have primarily focused on mobile phone integration. Both have developer-supported AR platforms (ARKit and ARCore, respectively) which enable developers to utilize tools within smartphones that detect and track surfaces, as well as placing models on them [2], [3], and [4].

The potential for gaming with AR has been demonstrated by the massively popular and successful *Pokemon Go!* Released by Niantic in 2016 [5]. As of July 2017 it has been downloaded almost 10 million times. The premise of this game is that as the user walks around in real life, they can find and catch virtual pokemon with their smartphone.

One major limitation of this gaming setup is the lack of a controller. Currently there are no peripherals for augmented reality on smartphones outside of the headsets users wear. When a user is wearing a smartphone on their face, they cannot interact with its touchscreen, which means that the inputs for controls have to come from onboard sensors such as an accelerometers or GPS units. It follows that since the phone is attached to the user's head, it can only follow the direction or position of the head, and cannot sense the extremities without seeing them with the camera. It is proposed that inertial sensors allow for movement of extremities to be sensed, recorded, and acted upon.

Inertial sensors have been used before in gait reconstruction and limb tracking [6], [7], and [8]. The applications of this technology have proven useful in physical training, whereby trainers and physicians examine how a person moves their body, but with more detail than a visual inspection. As shown by Thiel et al. [9], the use of inertial sensors can demonstrate a congruence between the lateral tilt of a ballet dancer's torso and lower scores from judges. Such observations can also be used in physical therapy; when an injured person demonstrates motion to a doctor, inertial sensors can provide indicators of strong or weak healing of muscles or tendons.

Our goal was to meld inertial sensing with AR technology in smartphones. In this sense, we wish to immerse the user in a game via AR, and substitute a conventional controller or controllers with inertial sensors on the foot. The sensors collected data about the foot movements of the user, and interpret the data as a kicking input to our soccer-based app.

For an inertial sensor to be used in a more active environment, some requirements must be met. The sensor package must fit on a user's shoe, without overhanging so much as to trip the user or become damaged from clipping an object. The sensor package must also be lightweight, as too much weight on a user's foot can upset their balance or become outright uncomfortable. Finally, the system must last a long enough time on a single charge for a user to have multiple sessions using the device. Table 1 includes the requirements, quantified:

TABLE 1
SENSOR PACKAGE REQUIREMENTS

Requirement	Status
App connects to Bluetooth and begins game in less than 10 seconds	Met - 2-5 seconds on average
App must determine user's kick speed and direction	Met - Straight, Angle, 90 deg kicks implemented
Ball must move with speed and direction proportional to user's foot	Met - Unity game engine maps input vectors to ball as a rigid body
System delay < 300ms, ideally <100ms	Met - 18.7 Hz from sensor to Unity, 60 Hz refresh, 5 frames = 83ms delay
Maximum dimensions of device: 4 x 3 x 2 inches	Met - 4.5 x 3 x 1.25 - overall volume is within spec
Maximum weight: 1 lb	Met - Weight 4.6oz less than 1/3 of spec)
Minimum battery life: 5 hours	Met - Sources ~65mA during operation = 46hr of battery life

II. DESIGN

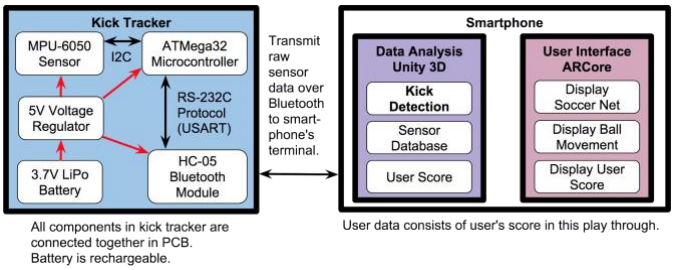

FIGURE 1. BLOCK DIAGRAM.

A. Overview

To create a sensor package as described previously, referred to from here on out as the "kick tracker," we needed a few core devices, as introduced in the block diagram in Figure 1. These include an inertial sensor, a microprocessor, a Bluetooth wireless module, a voltage regulator, and a battery. These components are described in further detail in the following blocks.

To start, it was necessary to consider what data we would need to represent the kicking motion. Acceleration data was used to determine whether the user was kicking or not. To deduce more information about a kick, angular velocity was obtained from a gyroscope. However, with 3-axis acceleration alone, we were able to determine what type of kick the user had made, and reflect this in-game. An acceleration range for each axis (X, Y, and Z) was set for three different types of kicks, as shown in table 2.

Axis	Straight		Angled (45°)		Side (90°)	
	Min	Max	Min	Max	Min	Max
X	1.50	2.00	1.40	2.00	1.50	2.00
Y	0.75	1.50	0.60	1.50	0.00	1.50
Z	-0.50	0.75	0.15	0.75	0.50	1.50

TABLE 2. MIN/MAX ACCELERATION THRESHOLDS FOR EACH KICK TYPE (IN UNITS OF G).

Acceleration data was pulled constantly from the inertial sensor. There are multiple possible protocols through which this could have been implemented. We initially tested with analog sensors, due to the simplicity of interfacing with them. However, other possibilities include Serial Peripheral Interface (SPI) and Inter-Integrated Circuit (I^2C), both of which output sensor data in a digital format [10]. We decided to use I^2C, because although both protocols are quite similar, I^2C made sense for our purposes because it uses a single data line, while SPI does not. Since we only received data from the sensor and did not transmit to it, this was a simpler setup. Our sensor [11] outputs 16-bit datagrams to be sent to the processor.

From the processor, we used Bluetooth LE (BLE) to transfer the data from the sensor to the phone [12]. While WiFi was also an option, it is important to note that this would have limited usability as it would have required a network. Bluetooth allows the user to connect the kick tracker directly to their device. Additionally, the latency of the BLE data transfer was as low as 3ms [13]. Given our need for end-to-end latency to be <100ms in order to retain user immersion, this allowed plenty of flexibility in additional processing time, while still staying below our delay cutoff.

For the purpose of our virtual game, we quantified what constitutes a kick. This included setting acceleration thresholds to determine the strength of a kick. Each piece of data was compared to these profiles and matched with one in order for a kick to happen. Once a piece of inertial data was transferred to the phone, these comparisons occurred as shown in the Data Analysis portion of Figure 1. At that point, user movement resulted in movement of the virtual ball if our system determined that a user had performed a sufficient kick.

All of this needed to be displayed on a smartphone in a generic AR headset. Given advances in AR from both the iOS and Android communities, either option would have been suitable for our end product, but we chose to work with the Android OS given our accessibility to AR-capable devices as well as its higher global market share. Figure 1 shows the information that was provided to the user in the User Interface section.

B. Sensor Network and Power Supply

The most fundamental piece of hardware was the inertial sensor. Our design included an Adafruit sensor known as the GY-521 MPU-6050, an inexpensive, thin, and low-powered chip that consists of both a triple-axis accelerometer and a triple-axis gyroscope [11]. The digital output accelerometer built into this sensor can measure a full-scale range of ±2g, ±4g, ±8g, and ±16g. The digital output gyroscope also contains a user-programmable full scale range of ±250deg/sec, ±500deg/sec, ±1000deg/sec, and ±2000deg/sec. This sensor also makes use of three 16-bit analog-to-digital converters (ADCs) for digitizing the gyroscope outputs and three 16-bit ADCs for digitizing the accelerometer outputs. These parts allowed for precise tracking of both fast and slow motion. The MPU-6050 has a communication interface of I2C, which allows it to read data from external sensors such as magnetometers. This interface communicated with our ATMega32 microcontroller in the design.

The next piece in our kick tracker was a microprocessor. We chose Atmel's low-power CMOS 8-bit microcontroller ATMega32/L due to its inexpensive cost, its versatility in interfacing, and reasonable size [14]. This processor contains several communication interfaces that were used for our design which includes I2C and ADC for collecting sensor data from the MPU-6050 sensor, and a serial programmable USART to communicate with the Bluetooth module. This processor is also capable of obtaining a high throughput of up to 16 MIPS (million instructions per second) at a clock rate of 16MHz.

C. Wireless Module

Our wireless interfacing device was the HC-05 Wireless Bluetooth Serial Transceiver, a simple Bluetooth module that provides the link between kick tracker and smartphone [15]. This part is also inexpensive with a USART interface that allowed it to communicate with the ATmega32 processor.

D. *Power System*

Lastly, we powered the system with a 3.7V 1000mAh lithium ion rechargeable battery cell [16]. These batteries were purchased inexpensively (~$5 to $9 per battery) and contain more than enough energy to power our system for extended periods of time. Some safety features included within these batteries consist of a safety vent to prevent from an excessive buildup of pressure and a built-in PTC (Pressure, Temperature, Current Switch) to protect against current surges. The battery was contained in a plastic battery storage case that is enclosed to prevent it from being damaged when attached within the "kick tracker" [17]. These batteries can also be recharged up to 1200 times. Our battery charger prevented batteries from being over-charged, over-discharged, supplied with too much voltage or current, or short circuited while charging [18].

E. *Data Analysis*

Once the data was transferred to the phone via Bluetooth, it was averaged with the values of the previous four samples. This was done to reduce the peakiness of the signal and to compensate for noise. When each new point of data entered the phone, the oldest point was discarded, forming an averaging window. The foot was modeled in the phone as a pendulum, as shown in Figure 2. The accelerometer was strapped onto the foot to coincide with the axes shown.

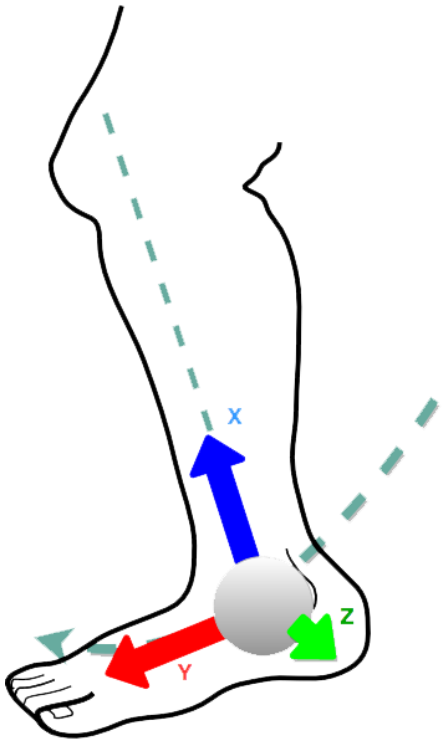

FIGURE 2. THE PHONE'S MODEL OF THE FOOT WITH ITS ASSOCIATED AXES.

The phone detects a kick when the thresholds for the X-, Y-, and Z-axes are met. These thresholds were determined empirically by recording traces from multiple users imitating a kick. An example of some recorded kicks is shown in the time series in Figure 3. Once a kick was detected, the incoming data was mapped onto the ball as a force vector. This ensured the ball moved realistically in relation to the user's foot.

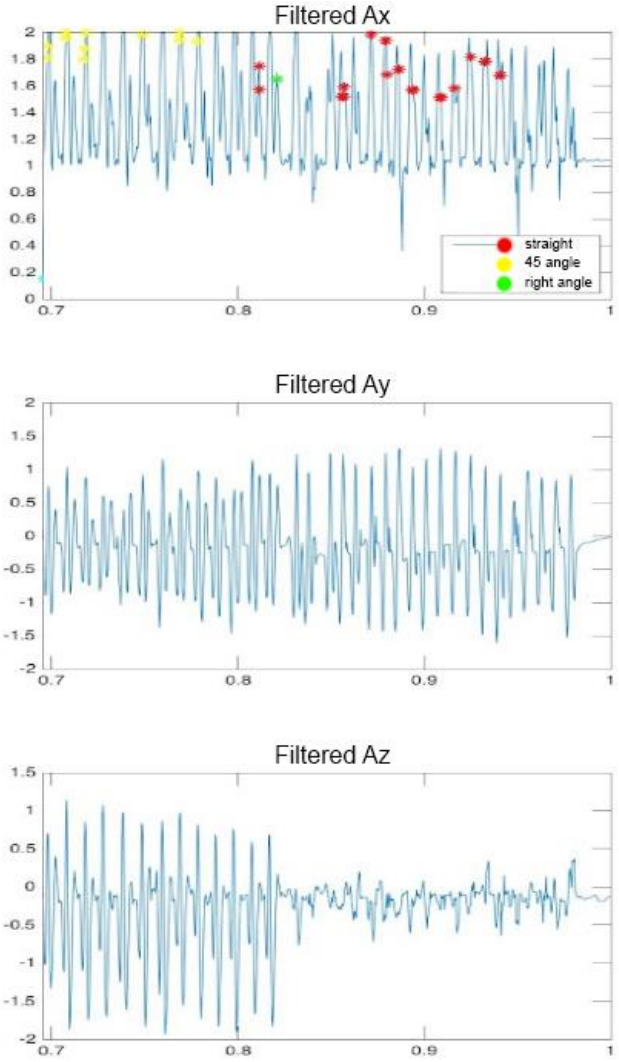

FIGURE 3. TIME SERIES OF MULTIPLE DIFFERENT KICKS.

F. *User Interface*

Using Google's ARCore API, we implemented a basic soccer ball/net environment, overlaid on the physical space that the user's smartphone camera can see from its headset mounting [3]. This system then is best used in a relatively open space, in order to provide realistic depth perception.

Once the user opens the app and selects "Play," they can then insert the phone into the headset. From here they must find themselves a suitable area for a plane, net and ball to spawn. Onec the net has spawned a countdown timer is initiated. Scores will only be tracked while the timer is running. Once completed, the user then has an option to save their score on a local leaderboard.

While the user sees both the virtual ball and net as well as their leg if it comes into view, the kicking is not vision-based. The app's display of a kick is entirely based on the input from the kick tracker.

III. PROJECT MANAGEMENT

Each member of the team had a specific role in this project. Matteo's primary responsibilities involved designing the physical device. This included designing the PCB, creating the

3D printed case, determining what material to use for the case, helping to select components, and analyzing power and battery needs. Chad led the general app development, including creating the preliminary app design and backend. He also helped with programming the microprocessor and team management. Ethan collected test data and used MATLAB to analyze it. This analysis was done to determine kick detection thresholds. Ethan also helped with selecting components, prototyping the device, and analyzing power and battery needs with Matteo. Jackie was responsible for implementing augmented reality into the app with ARCore and designing the gameplay. This included selecting models for the ball and goal, implementing proper physical behavior of the ball, creating the graphical user interface, and creating the scoreboard.

Our team communicated each week for at least one hour with our adviser Professor Holcomb. We also met as a team without our adviser at least once a week. We regularly updated our evaluators before presentations and created all of our presentations together as a team.

IV. Conclusion

Since MDR, we made enormous progress on this device and app. The majority of the app development and device construction was done over this period of time. At MDR, we had only a proof of concept with no game or data analysis performed yet as we had completely changed our project after PDR. Since MDR, we switched sensors, created several iterations of the case and PCB, implemented augmented reality, performed data analysis, and designed the game itself.

Overall, we were able to create a successful physical device and app to play augmented reality soccer. ARK was successful in that it met all of the system requirements and was able to be played by the public on demo day. There are a few aspects of ARK that could be improved but a future team, including making the final foot attachment device even smaller by surface mounting components to the PCB and increasing the complexity of the app. For example, having balls bounce off of surfaces other than the floor, introducing obstacles, and varying the distance from the net might have improved the gameplay.

References

[1] "Everything You Need to Know About Augmented Reality Now That It's Invading Your Phone," Gizmodo, 2017 [Online]. Available at: http://fieldguide.gizmodo.com/everything-you-need-to-know-about-augmented-reality-now-1809069515/.

[2] Matney, L. (2018). *Google shows off ARCore, its answer to Apple's ARKit*. [online] TechCrunch. Available at: https://techcrunch.com/2017/08/29/google-shows-off-arcore-its-answer-to-apples-arkit/.

[3] Google Developers. (2018). *ARCore - Google Developer*. [online] Available at: https://developers.google.com/ar/.

[4] Developer.apple.com. (2018). *ARKit - Apple Developer*. [online] Available at: https://developer.apple.com/arkit/.

[5] Perez, S. (2018). *Pokémon Go becomes the fastest game to ever hit $500 million in revenue*. [online] TechCrunch. Available at: https://techcrunch.com/2016/09/08/pokemon-go-becomes-the-fastest-game-to-ever-hit-500-million-in-revenue/.

[6] R. Slyper, J. K. Hodgins, "Action Capture with Accelerometers," Eurographics/ ACM SIGGRAPH Symposium on Computer Animation, 2008

[7] H. Yang and J. Ye, "A calibration process for tracking upper limb motion with inertial sensors," *2011 IEEE International Conference on Mechatronics and Automation*, Beijing, 2011, pp. 618-623

[8] A. Ahmadi *et al*., "3D Human Gait Reconstruction and Monitoring Using Body-Worn Inertial Sensors and Kinematic Modeling," in *IEEE Sensors Journal*, vol. 16, no. 24, pp. 8823-8831, Dec.15, 15 2016.

[9] David V. Thiel, Julian Quandt, Sarah J.L. Carter, Gene Moyle, Accelerometer based Performance Assessment of Basic Routines in Classical Ballet, In Procedia Engineering, Volume 72, 2014, Pages 14-19, ISSN 1877-7058, https://doi.org/10.1016/j.proeng.

[10] Introduction to I²C and SPI protocols – Byte Paradigm – Speed up embedded system verification", *Byteparadigm.com*, 2018. [Online]. Available at: https://www.byteparadigm.com/applications/introduction-to-i2c-and-spi-protocols/. [Accessed: 10- Feb- 2018].

[11] InvenSense "MPU-6000 and MPU-6050 Product Specification Revision 3.1," MPU-6050 datasheet, Nov. 2010 [Revised Oct. 2011].

[12] "Introduction | Introduction to Bluetooth Low Energy | Adafruit Learning System", Learn.adafruit.com, 2018. [Online]. Available at: https://learn.adafruit.com/introduction-to-bluetooth-low-energy/introduction. [Accessed: 13- Feb- 2018].

[13] "SIG INTRODUCES BLUETOOTH LOW ENERGY WIRELESS TECHNOLOGY THE NEXT GENERATION OF BLUETOOTH WIRELESS TECHNOLOGY | Bluetooth Technology Website", Bluetooth.com, 2018. [Online]. Available at: https://www.bluetooth.com/news/pressreleases/2009/12/17/sig-introduces-bluetooth-low-energy-wireless-technologythe-next-generation-of-bluetooth-wireless-technology. [Accessed: 10- Feb- 2018].

[14] Microchip Technology "8-bit Microcontroller with 32KBytes In-System Programmable Flash," ATmega32(L) datasheet, Feb. 2011 [Revised Jul. 2017].

[15] ITead Studio "HC-05 Bluetooth to Serial Port Module," HC-05 Bluetooth Module datasheet, June 2010 [Revised Jul. 2010].

[16] "Lithium Ion Battery - 1Ah", *Sparkfun.com*, 2018. [Online]. Available: https://www.sparkfun.com/products/13813/. [Accessed: 12- Apr- 2018].

[17] "JST Jumper 2 Wire Assembly", *Sparkfun.com*, 2018.[Online] Available: https://www.sparkfun.com/products/9914. [Accessed: 12- Apr- 2018].

[18] "SparkFun Battery Babysitter - LiPo Battery Manager", *Sparkfun.com*, 2018.[Online] Available: https://www.sparkfun.com/products/13777. [Accessed: 12- Apr- 2018].