Smart Diagnostic Toy

Authors: Jordan Cox, Naina Sharma, Ameer Khan, Jacques Villaflor, Tolu Familoni

Index Terms—childhood development delays, diagnosis, sensors, tests.

1 Product Summary

The Smart Toy is an interactive toy cube integrated with a sensor that has the ability to observe a child's psychological as well as physiological development by monitoring receptive communication skills, expressive communication skills, and fine motor skills.

Receptive communication is the process of recognizing or understanding spoken words and directions. The examiner verbally prompts the child to do a particular task (in this case to press the button corresponding to a particular shape and color combination). The child is expected to understand the directions given and show that understanding by completing the task.

Expressive communication is the process of sending a message across or making something happen. The act of pressing any button on the cube or having any kind of response to the prompts given is monitored.

Fine motor skills refer to the movement or coordination of smaller muscles such as hands and fingers. The act of holding the cube, moving it around, and pressing its buttons are all activities that help give an insight into a child's fine motor skills.

2 Introduction/Motivation

Every child grows/develops at a different rate. It is important to monitor that growth especially until age six to ensure that a potential development delay is not taking place. Conditions such as attention-deficit/hyperactivity disorder (ADHD) and autism spectrum disorder (ASD) begin to show signs at an early age. Thus accurate and early detection in young children is crucial in order to receive proper attention and proper treatment on time.

Currently development delays are diagnosed through a series of tests that take place in a doctor's clinic only when a parent actually brings in their child. It can be difficult to gather pertinent and impartial information just through ordinary doctor visits and simply waiting for a parent or teacher to notice a red flag or symptom is also not the most efficient. That is why having a product that could enable such monitoring in a continuous manner and make it so that it could take place in any kind of setting (school, home, or clinic) would be very valuable.

And that is exactly where the motivation/inspiration for our project came from. The intention was to produce a proof of concept device that would demonstrate that it is possible to monitor the physiological and psychological development of young children using a sensor driven interactive device that could collect a wide range of useful information over a period of time that established testing techniques/methods may not. The device would need to fulfill the requirements of traditional childhood development tests, be accessible and appealing to young children, and be safe and robust enough to be handled by them.

3 Prior Work

Our project is heavily influenced by the Bayley Scales of Infant and Toddler Development tests. These are a series of tests designed for children ranging from the age 1-42 months[3]. The tests are conducted using various items to test a childs cognitive development, motor development, and behavioral development. A video clip [2] shows the Bayley assessments in action. One of the tests performed in the video involves a child requiring to match several colored disks with crayons. As the child verbally identifies each disks color, the person performing the test stacks the disks based on the crayons spatial arrangement. The examiner lets the child know if her answer is wrong, and then asks the child why her answer was incorrect.

Another example test that inspired aspects of our own test makes use of toy ducks. The child is presented with ducks of multiple different colors, two different sizes for each color (one small and one big), and two different weights for each color (one significantly heavier than the other). The child would then be instructed to pick out a particular color duck or ducks, then to pick out a particular size duck or ducks, and then to pick out which duck is heavier. As this test makes used of three different components, size, color, and weight, it can be used to generate a lot of tests and therefore a wide range of data. This is something we aspired to accomplish as well; however instead of making use of multiple items we wanted to be able to accomplish the same results using a single item. Most of the child diagnostic tests such as the Bayley, Merrill-Palmer, and Denver tests make use of multiple items and require a proctor to note down the results and data generated. There dont seem to be readily available devices on the market that can automate that process and conducts the tests using just one, or at least far fewer, devices.

Our Smart Toy's concept is also heavily influenced from Diego Rivera and his group's design. The research group developed a prototype cube that senses its orientation, as well as its alignment with other cubes [1]. The primary purpose of the design is to detect common signs of developmental disorders such as slow movement, long deceleration time, and high amplitude speed peak [1].

Our group planned on integrating the color-matching section of the Bailey Tests with Rivera and his groups version of a smart toy design to develop an inexpensive multi-purpose cognitive and physical diagnostic toy.

4 Planned Solution

Our team proposed to design a smart toy in the form of an interactive cube that would make use of sensors to monitor the development of young children. Our solution to this problem entails the following deliverables: an interactive experience with buttons and reactive RGB LEDs, a motion sensing method with a multi-axis Inertial Measurement Unit in order to provide diagnostic data regarding fine motor skills and a real-time data link that would be maintained through wireless communication from the device to a "base station" to stream the sensor and game data. The "base station" would be able to monitor the data being generated in real time, collect the data for analysis, generate a technical report and enter the report into a database.

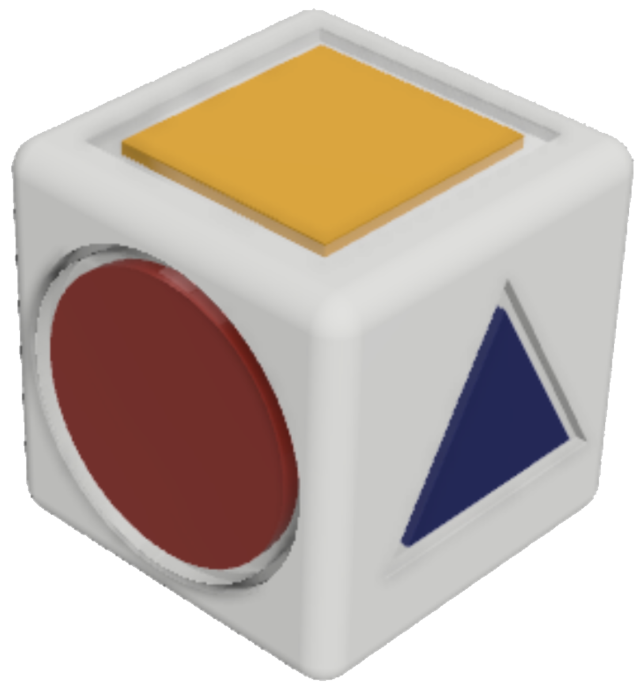

Fig. 1. Initial concept for the Smart Toy.

The cube would have a button and LED on each face monitored and controlled by a microcontroller. Each face would also have a shape on it: circle, square, or triangle. As there are six faces on a cube each shape would be repeated on the opposite side with the LEDs always guaranteed to be a different color. The child would be expected to press the button present on the side that contains a particular shape-color combination. The accuracy and speed of this task would be monitored. Another factor that would be monitored using the IMU sensor would be the fine motor skills of the child. This way more data could be collected regarding the smoothness and quickness with which a child responds to, and interacts with, the device. With the motion sensor we could also potentially introduce another test where a child would be expected to, instead of press a button, place the cube so that a particular shape-color combination is facing up.

5 Preliminary Game Design

Having had some experience with the chipKIT UNO32 microcontroller our team decided to use it as the core of our toy design. In order to get familiar with the board and the MPLABX platform, as well as to take advantage of the buttons and LEDs already present on the Digilent I/O shield that came with it, we decided to make a simple four button precursor of our game. The I/O shield has four buttons and eight LEDs on it, which informed the game concept that we developed; each of the four buttons would be assigned a shape, a circle or a triangle, so that there would be two of each shape and each shape would be assigned a color, red or yellow, based on which one of the two LEDs directly above it was lit. The game would designate the LEDs to be lit randomly keeping in mind that the same two shapes cannot be the same color.

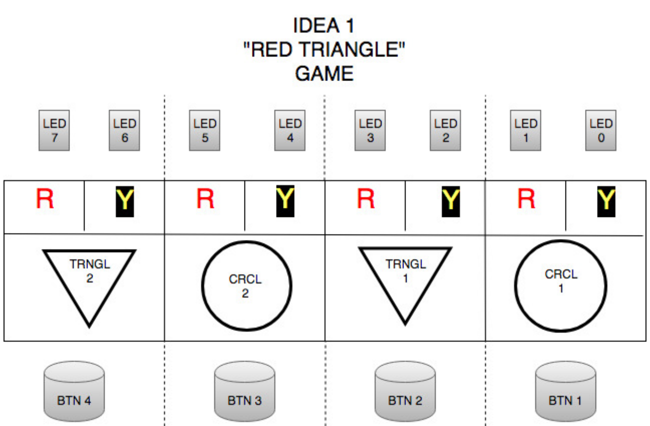

Fig. 2. This figure displays the outline created for the 4 button design of the game. It utilized the buttons and LEDs present on the I/O shield.

To create the software of the game we made various libraries and functions. We made a BUTTONS library that dealt with all of the interaction with the buttons (as well as factors like debouncing). We created a LEDS library that managed the LEDs on the I/O shield. We created a GAME library that was responsible for game generation (such as the randomization of the color and shape combinations) and we also created a RESULTS library that contained functions that were responsible for storing the results of the game (information such as timing and accuracy) as well as printing them to the screen in a comprehensible manner.

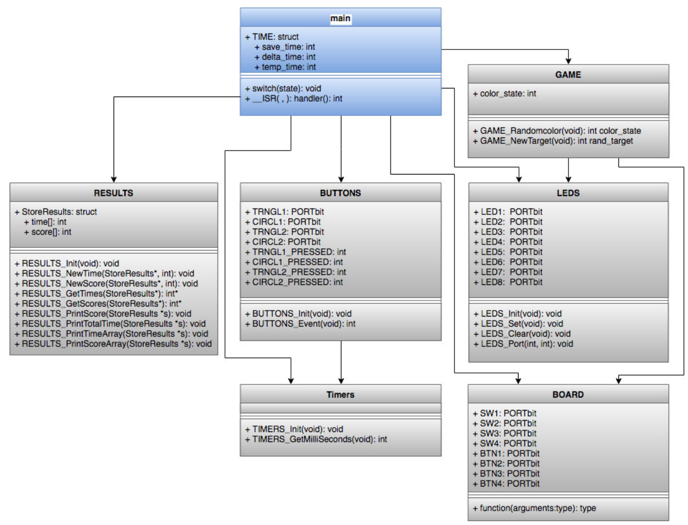

Fig. 3. This figure displays the UML chart that maps out all of the files and function created to create the software system of our design.

In the main file the game was run using a state machine implemented using a switch statement. Initially the game would start in an idle state and it would only get out of the idle state upon pressing the button representing the second triangle, at which point the game would officially start. The LEDs in the shield would be randomly lit and the user would see a prompt to press the button representing a particular shape and color on the ds30-loader's serial display. The user would be expected to respond and this would repeat for a total of ten rounds after which the game would enter the game over state. In the game over state the results such as how long it took the user to answer each prompt, how long it took the user to play the entire game and how accurate the users responses were would be printed to the screen. The user would be asked if they wanted to

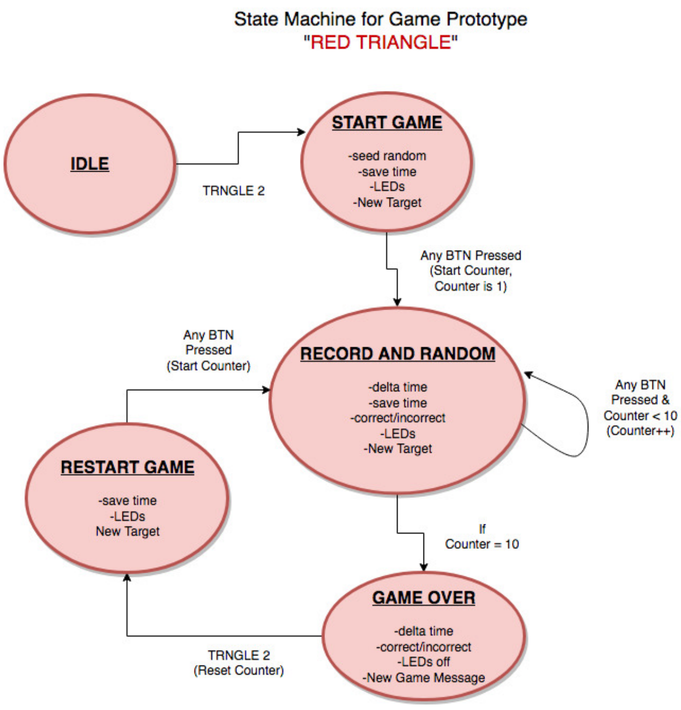

Fig. 4. This figure displays the state machine for the 4 button game.

restart the game and upon pressing the button that represents triangle two the game would restart and loop through the state machine in the same manner again. As a result through creating this preliminary game design we were able to get a feel for what components would be needed to proceed to the six button breadboard design.

6 Breadboard Design

After the four button game we needed to move the game to a breadboard, allowing us to start using components that would be used in the final product. Having the physical components on a breadboard prototype allows us to systematically add components and possibly change them if there is a problem. We expanded the game to have six buttons with six corresponding LEDs. Each button is associated with one of three possible shapes and one LED; the game currently uses three colors while ensuring that each pair of shapes have different colors.

We decided to use momentary push buttons, which are a very common, and the

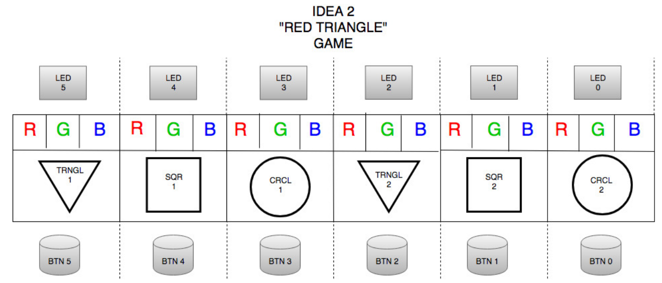

Fig. 5. This figure displays the outline created for the six button design of the game. It utilized the momentary buttons and NeoPixel LEDs.

same as the buttons on the I/O shield. The normal setup for momentary push buttons will be used, which is where the buttons will only generate a high signal while the button is being held down. Smaller buttons are being used for the breadboard prototype, they are designed to fit on a breadboard, and testing shows they dont bounce as much as the I/O shield buttons. Larger, arcade style, buttons will be used on the cube and will provide an easier target for people taking the test. In testing the larger buttons also produced a clean signal with less bounce than the I/O shield buttons.

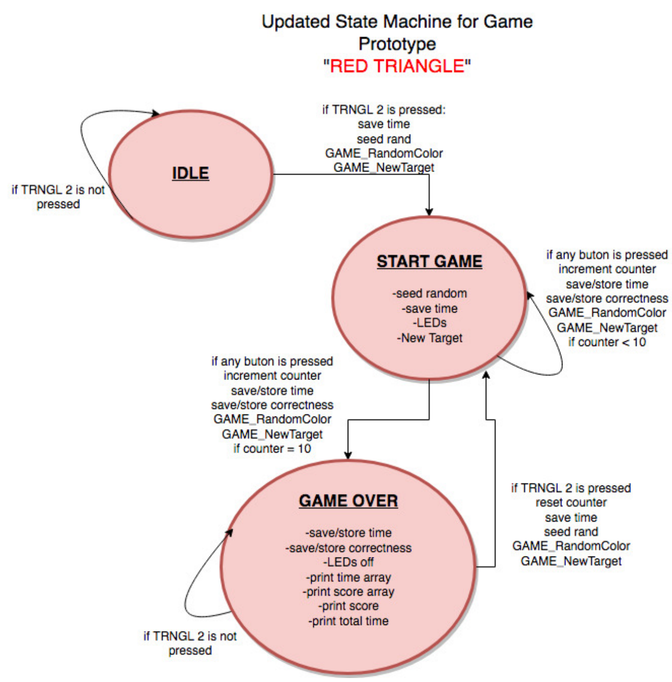

Fig. 6. This figure displays the updated state machine for the 6 button game. It has fewer states than the previous state machine. Also, the actions take place upon transition rather than being tied to the state.

While we worked to complete the breadboard prototype a UML chart was maintained representing the modules of our code. This gave all team members a reference to see the overall structure of the program and which functions where available from each module. The main file ensures all of the

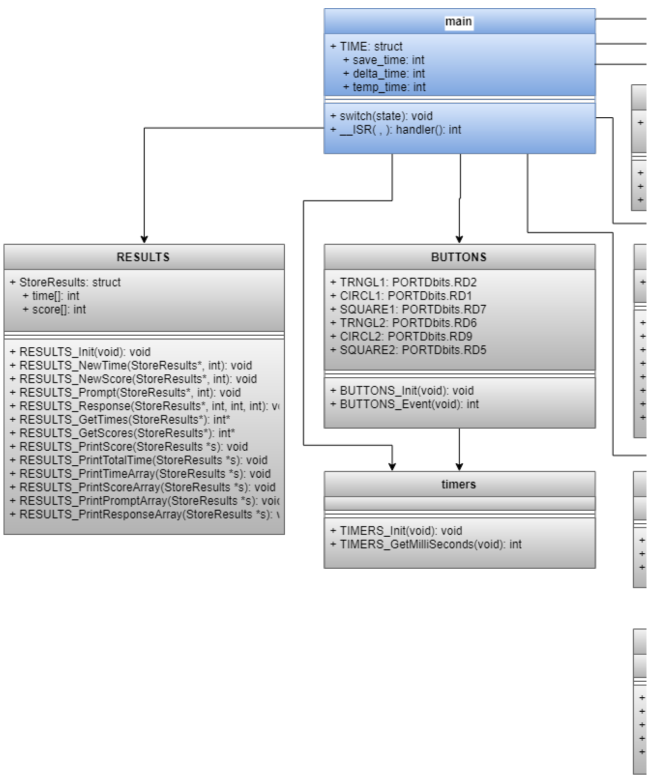

Fig. 7. Left side of UML chart of all modules used in final product.

game, when this module is also used to retrieve the data when it needs to be sent to the base station. The IMU module provides functions that take readings from the IMU, since the IMU uses the I2C communication protocol the I2C module is used. The Bluetooth_uart module

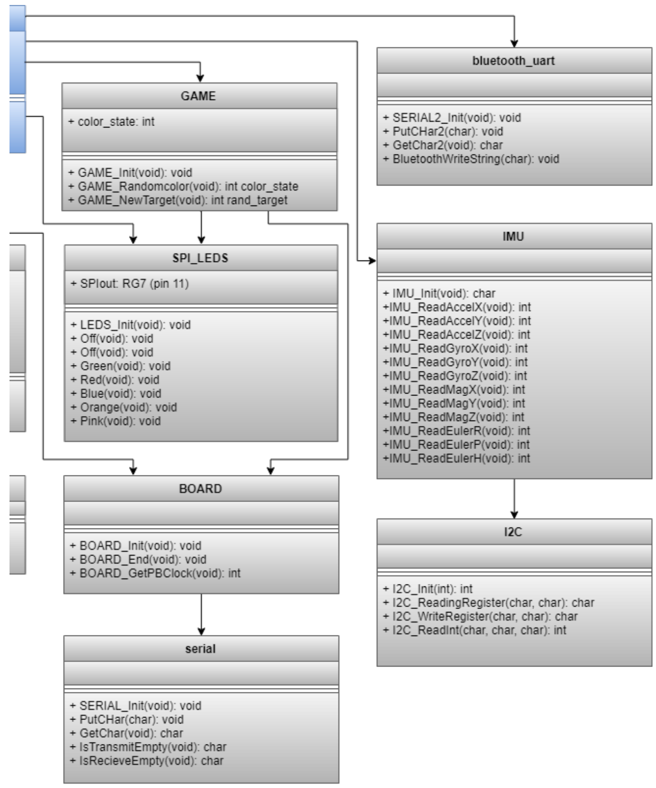

Fig. 8. Right side of UML chart of all modules used in final product.

modules initialise, along with any hardware that they control. It also runs the game by using a switch statement to implement a state machine, which will be described in a moment. The buttons module provides software debouncing of the buttons and function to check the current status of the buttons. The Game module randomly assigns which colour each LED will be whenever they need to change colour, while ensuring that each pair of shapes are not displaying the same colour. It also randomly selects one of the existing colour/shape combinations to be the next target. The SPI_LEDs module provides functions that are used to change the LEDs colours. The Results module allows us to save all of the game data that does not come from the IMU on the Uno32 until the end of the allows communication using the Bluetooth device and is based of the serial module. The Timers, board, serial and I2C modules are authored by Max Dunne, and sourced from the classes we have taken, they allow interaction with various features of the UNO32.

7 LEDs

At the beginning of our initial LED design, we hoped to utilize six RGB LEDs, one for each face of the cube. However, this design implementation required us to use eighteen different pins on our UNO32 in order to work. This design caused an issue because these LEDs would take up the majority of the pins on the UNO32 and the buttons still had to be implemented. In order to work around this we quickly came across a simpler solution by using NeoPixel LEDs instead, these are individual color LEDs that daisy chain together and use a single-wire control protocol. This design allowed us to use only three pins for all the LEDs, with a pin used each for power, ground and data. The data wire was connected to pin eleven of the UNO32 which is used for the SPI communication protocol. The Neopixels expected a specific waveform which they interpret as binary one's and zero's, the SPI protocol was used to emulate this, with one byte of SPI data being translate by the NeoPixels as a single bit. The first NeoPixel would receive the entire data stream for all the NeoPixels, use the first twenty-four bits of data and then pass the rest of the data stream onto the next NeoPixel, with each successive NeoPixel using and removing the first twenty-four bits and passing on the rest. In order to control the brightness and color of the pixel, eight bits were assigned to each color in the Green, Red, Blue scale meaning a total of twenty-four bits were sent to each pixel. The SPI bytes used to represent each NeoPixel bit were 0xF0 and 0xC0 which represented one and zero respectively. The NeoPixels have very strict timings tolerances since they interpret two differently timed high and low periods as ones or zeroes. Once the specific timing was achieved we were able to display the desired colors on the NeoPixels.

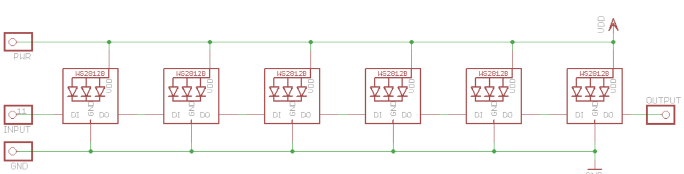

Fig. 9. Schematic of six Neopixels

8 Buttons

As implementation of the six button game design began we discovered that the BUTTONS library would need to be modified and a method to read in external buttons to the UNO32 would need to found. In addition, the buttons would need to be debounced so that each press could be registered as a single event only. In an attempt to address both of these requirements at the same time we attempted using the change notify interrupt method which would be triggered each time there was a change in the port of the UNO32. This method was effective in that it did a great job of noticing when a button was pressed. However, because there was only a select amount of pins with the change notify capability we found that there was interference within the port when playing the game and this resulted in incorrect game results. In an attempt to fix this issue we then decided to try using a timer interrupt. This gave us a larger selection of pins we could use within the UNO32 since any general I/O pin could be used. We were also able to take care of the debouncing issue by modifying how often the timer interrupt would be triggered. This is the method we stuck with as it does a great job of solving both of our requirements.

9 Inertial Measurement Unit (IMU)

Diagnosing a childs physiological behavior would require an analysis of the childs interaction with the smart-toy. Specifically, the toy must be able to record the childs gross and fine motor skills by measuring how fast and with how much control the child orients the cube. An IMU is sufficient for generating this data.

The MPU-9250 was considered as several team members already had experience using it, but the gyroscope's readings had a variable drift that would need to be constantly monitored and accounted for and, more importantly, they also reported difficulties calibrating the magnetometer, which resulted in very unstable absolute attitude estimation. After some investigation another IMU was found that had built in calibration features, with only a slight cost increase. The BNO055 IMU sensor expects, but does not require, a simple calibration procedure on start up, which, when performed, produces very good raw sensor readings and can also report Euler angles. Unfortunately the roll and pitch Euler angles generated by the BNO055 are limited to $\pm 90°$ and $\pm 180°$, respectively. A suit of procedures previously developed for the MPU-9250 which used accelerometer, magnetometer, and gyroscope readings to derive a directional cosine matrix (DCM) and extract the Euler angles of orientation were tested using data from the BNO055. They produced excellent results, and the Euler angles for pitch, roll and yaw were seen to be reported within $\pm 360°$.

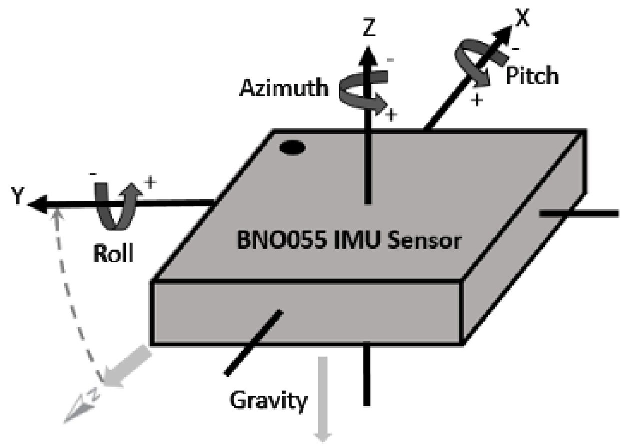

Fig. 10. The BNO055 measures translation, rotation and the local magnetic field.

The fusion of calibrated sensors have many uses in general and some specific uses for the cube. While none have been implemented yet, the code is available to make future integration as simple as possible. Currently only a relative method of rotational position is used for diagnosis and when the IMU is calibrated this produces more than adequate results.

10 Bluetooth

During the initial stage the team deliberated on which data transfer module to use: Bluetooth or Zigbee. We decided to use Bluetooth primarily due to its wide spread use, allowing the toy to potentially connect with any device which has Bluetooth built in. We first tried using the nRF USB dongle but a lot of time was spent making slow progress, with the total time to achieve what we needed remaining unclear. Around that time, we learned that another student team, the sailboat team, had made considerable progress with the HC-05 Bluetooth module. Their team member which had worked on their Bluetooth requirements was able to assist and instruct us on using the HC-05 and we were quickly able to have it performing the tasks we needed.

The HC-05 Bluetooth module is used to provide wireless data transfer from the smart toy to a computer. The UNO32 uses one of it's UARTs to communicate with the module, which in turn passes the data to a computer over Bluetooth.

11 Data Processing

In order to visualize and analyze the data we first had to process the raw data which had been collected from the IMU, transmit it over Bluetooth, and then write it into a comma separated values (csv) file. The IMU data that was collected and written to the csv file were the raw values for each axis of the gyroscope and the accelerometer. The csv file also included the time, score, prompt and response for each question. In order to process the raw data we chose the python language because it has powerful standard libraries which offers a fast, reliable and cross-platform environment for data visualization and data analysis.

The python libraries that were necessary for visualizing and analyzing the data were matplotlib, numpy, pandas and scipy. The library called reportlab was used to generate the diagnostic report. The matplotlib library was useful to create 2D bar graphs and plots of the gyroscope, accelerometer and game data. The numpy library was useful because it was used as an efficient multi-dimensional container of the data. This library also allowed us to plot the Fourier transform of the integrated gyroscope data (relative angular position) using a function called fft and fftfreq. The pandas library was useful to parse through the csv file easily allowing the use of data based on column headers in the csv file. This library was also useful for data manipulation and data analysis. The scipy library was useful because it had an integrate function which was used to cumulatively integrate y(x) using the composite trapezoidal rule. This allowed us to plot the integrated gyroscope data over time. Finally, the reportlab library was necessary to generate a diagnostic report with all the plots, graphs and tables.

12 Cube Integration

For the actual toy that would encompass our game, we opted to go with a cube box since many toys intended for young children are in the form of cubes. We started off with a regular wooden box which we painted white to provide a good background for the LED lights and to contrast with the black shapes. We drilled holes into all the sides for the buttons and LEDs and soldered wires on to each of the components which were then connected via ribbon cables to the UNO32. A perf board was designed and built to take care of all the power and ground supplies as well as to route the output signals from the buttons.

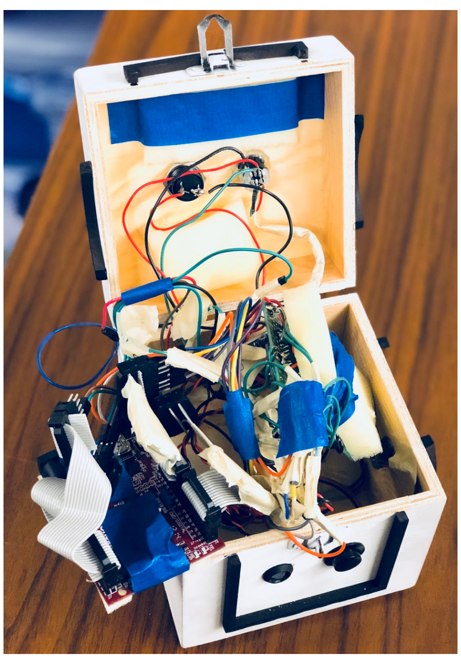

Fig. 11. The inside of the cube.

We initially designed shelving inserts to slide into the cube and create compartments for our components but they ended up taking too much space so ended up attaching some foam to the components to limit any kind of negative impact when the cube would be turned.

In order to power our cube, we implemented two 9V batteries in parallel as a supply to the UNO32. With the built in regulator on the microcontroller board, we were able to supply 3.3V for the buttons, LEDs and IMU as well as 5V to the bluetooth module.

13 Step-by-Step Smart Toy Process

In order to conduct an actual test using the smart toy a series of simple steps would need to be followed. First the cube would be turned on and calibrated by setting it down on each of its face for 5-10 seconds and then by moving it around in a figure 8 motion. Then the users

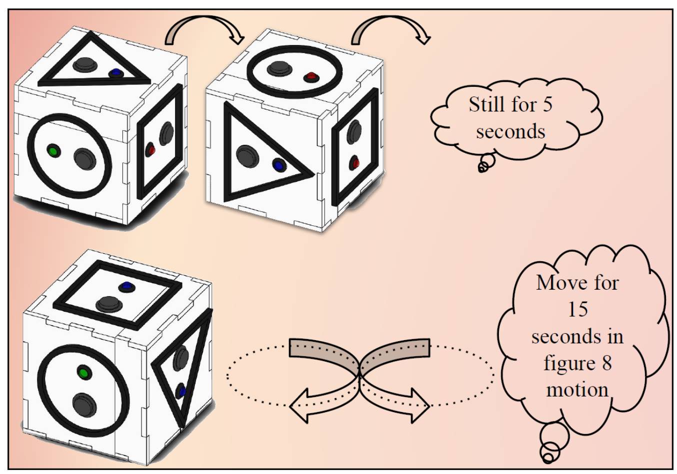

Fig. 12. Calibration steps.

personal information such as name, date of birth, gender, etc. would be taken and entered into a text file. The cube would be introduced to the user and the script handling the Bluetooth connection on the base station would be run. The user would be asked to start the game by pressing the start button and at this point the prompts would start getting displayed on the computer. They would be read to the user and as they user would respond to them the game data as well as the IMU would be streaming in a window. At the end of the game the script handling report generation would be enabled so that a report of the users performance could be generated that would take into account the game and IMU data and also the text file containing the user's information in it.

14 RESULTS

Once the game is complete, we have a python script that produces the patient's diagnostic report. The first page of the report includes the patient's information such as name, age, gender and date of birth. It also includes a description about the smart toy and a description about whats being monitored. The report also includes various plots, graphs and tables that visualizes the results of the data.

14.1 Physical Development

The first plot displays the integrated x, y, and z angular velocities collected from the IMU's gyroscopes. Integrating angular velocities produce angular positions, but without referencing other factors (such as gravity and magnetic fields) the starting reference position for these angles is random. This isn't a problem, though, as we are only considering the way the cube is being handled, not it's orientation relative to the Earth's center and magnetic north pole. The black dots on this plot mark

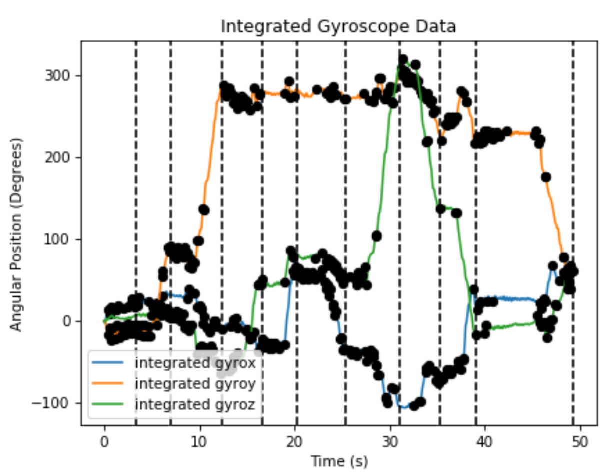

Fig. 13. This plot displays the integrated x, y and z angular velocities collected from a gyroscope.

the inflection points. The black dashed vertical lines mark the time in which each button was pressed. The least amount of dots between the peaks and valleys would suggest a stable rotation. The report also includes a list of these inflection point times as well as the quantity between each button press.

Another plot displays the Fourier Transform of the integrated gyroscope data. The examiner can determine the steadiness of the cube by analyzing the frequency from the Fourier Transform plot. For additional analysis, a table is included that displays the sums, means, medians, standard deviations, minimums and maximums for each axis data.

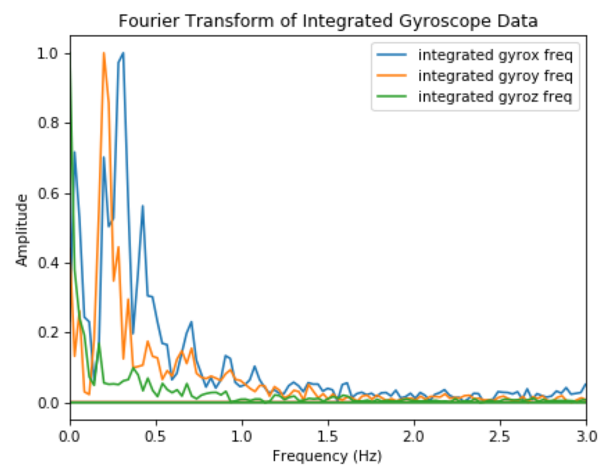

Fig. 14. This plot displays the Fourier Transform of the integrated gyroscope data.

14.2 Mental Development

The last set of tables and graphs summarize the results of the test. The first bar graph keeps track of incorrect answers which are categorized by shape and color. Another bar graph shows the total time, average time, maximum time, minimum time and time taken to answer each question. The final bar graph shows the average time taken to recognize each color and shape. There is also a table that displays the score, response time, prompt and response for each test question.

14.3 Rating

The final data table gives an arbitrary rating from 0 to 10 based on the score, total time, and average number of inflection points.

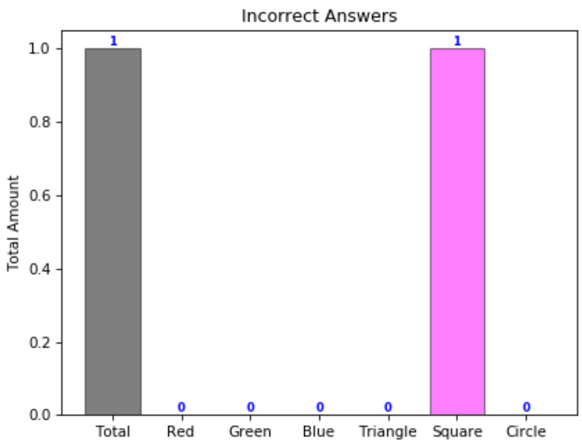

Fig. 15. This bar graph keeps track of incorrect answers which are categorized by shape and color.

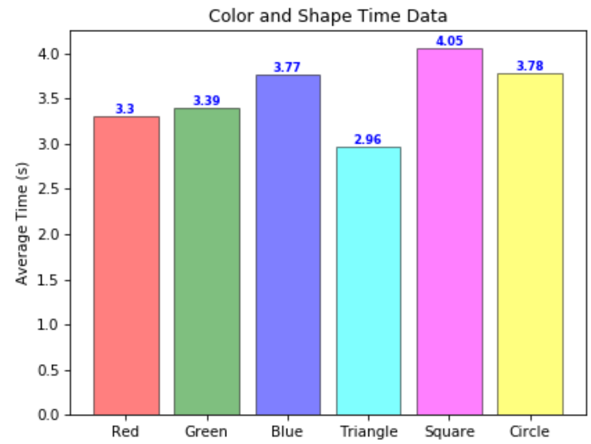

Fig. 17. This bar graph shows the average time taken to recognize each color and shape.

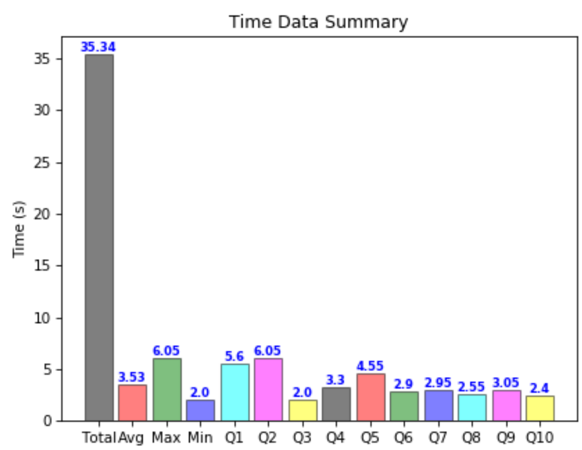

Fig. 16. This bar graph shows the total time, average time, maximum time, minimum time and time taken to answer each question.

Rating	Total Score	Total Time (s)	Avg Inflection Points
0	0	>55	>70
1	1	50 - 55	63 - 70
2	2	45 - 50	56 - 63
3	3	40 - 45	49 - 56
4	4	35 - 40	42 - 49
5	5	30 - 35	35 - 42
6	6	25 - 30	28 - 35
7	7	20 - 25	21 - 28
8	8	15 - 20	14 - 21
9	9	10 - 15	7 - 14
10	10	0 - 10	0 - 7

Fig. 18. This figure shows the different ratings that can be given based on the score, total time, and average number of inflection points.

15 Conclusion

In conclusion, through our work the past three quarters our group has been able to design and produce a proof-of-concept smart toy that can aid in the early diagnosis of developmental delays within young children. With our integrated sensor suite consisting of buttons, LEDS, an IMU module and Bluetooth module, a young child is able to play an interactive game while analytical data regarding their performance is tracked and stored. Following the completion of the game, we are able to take the data from each game, send it to a laptop via Bluetooth and generate a diagnostic report which summarizes the findings from the game. This diagnostic report has multiple plots and graphs that visualizes the data collected from the IMU module and the game. This type of report will enable guardians to have an at-home, user friendly idea of what is going on with their child without having to go into the doctors office. Sending this report to a health care professional is the ideal path of action our team would suggest as it would allow for further advise and additional insight into the child's development.

16 Future Plans

Moving forward, this device can be built upon to increase its use and testing capabilities. Some of those ideas include: adding a mobile component to the test for gross motor skills, building multiple cubes that could interact with one another, adding sounds and adding numbers. In addition to possible expansions that can be made to the physical device, the data generated by the sensors can be used in a variety of ways to help child development experts and doctors analyze as well as test for different conditions.

Fig. 19. Completed Smart Toy.

If our product was going to be brought to market it would need further development. Our current design uses two 9 volt alkaline batteries which would need to be replaced with a permanently installed rechargeable battery, or batteries, which could match their performance while also being suitably safe to be installed within a children's toy. The current 9 volt batteries provide the approximate 110mA drawn while operating the toy, while also maintaining function when buttons are pressed and held down, causing the current to jump to 180mA. The 9 volt batteries have also been found to function for three hours of active use, a period which would hopefully be exceeded with the new batteries in the next iteration of the smart toy. Related to changing the batteries, an external on/off switch, charging port, and possibly a USB port would need to be added, probably in a recess and protected by a cover. Custom PCBs would also need to be designed to save space within the toy. Two PCBs should be able to hold all of the required components. The current microcontroller has many features which are unused, and a simpler, smaller one could be integrated onto a PCB, with other required components being the Bluetooth, IMU, power supply and management and connections for the buttons and LEDs. Saving space within the toy would be critical as it has been found that the current version of the smart toy is slightly larger than would be ideal for young children to handle. We would also have to make the new version of the toy from a suitable material for children to interact with.

References

[1] D. Rivera, A. Garcia, B. Alarcos, J. Velasco, J. Ortega and I. Martinez-Yelmo, "Smart Toys Designed for Detecting Developmental Delays", Sensors, vol. 16, no. 11, p. 1953, 2016.

[2] YouTube. (2018). Part 1 Bayley Infant Toddler III Screen. [online] Available at: Bayley Test. [Accessed 14 Dec. 2018].

[3] Healthofchildren.com. (2018). Bayley Scales of Infant Development - Definition, Purpose, Description. [online] Available at: Bayley Scale [Accessed 14 Dec. 2018].

[4] A. Industries, "NeoPixel Diffused 8mm Through-Hole LED - 5 Pack", Adafruit.com, 2019. [Online]. Available: Adafruit Neopixel [Accessed: 11- Mar- 2019].

[5] "Using WS2812 RGB LEDs — Dev Center", Developer.electricimp.com, 2019. [Online]. Available: Neopixel Datasheet. [Accessed: 11- Mar- 2019].

[6] 4cdn.sparkfun.com. [Online]. Sparkfun. [Accessed: 14-Mar-2019].

Fig. 20. The Team (left to right): Jacques Hendrich Fajilago Villaflor, Ameer Khan, Jordan Cox, Naina Sharma, Tolu Familoni.