IoT4Fun Rapid Prototyping Toolkit for Smart Toys

A. P. de Albuquerque CIn-UFPE, Brazil <u>apa@cin.ufpe.br</u> Judith Kelner CIn-UFPE, Brazil jk@cin.ufpe.br Thiago D. Nogueira CIn-UFPE <u>thiago.nogueira@gprt.ufpe.br</u> R. S. Rocha Junior CIn-UFPE railton.rocha@gprt.ufpe.br

Abstract

Rapid prototyping tools turn the design of smart toys faster and easier for creative teams. Appropriate tools for smart toys should meet a list of requirements, which include distributed data collection and adaptability for assorted toy shapes and size. The IoT4Fun toolkit innovates by mixing the embedded, modular, and plugand-play approaches. It supports motion tracking data, wireless communication, and contactless identification. IoT4Fun demonstrates its effectiveness to design a variety of smart toy solutions by fitting into a hula-hoop toy until spherical, cubic, and wearable shapes. Solutions connect with either mobile applications or other toys and play rules range from open-ended to closed behaviors. End-users exhaustively tested developed solutions, and technical assessment evaluates their integrity after playtesting sessions. Results show comparative data on battery consumption and vulnerabilities threats for data security and privacy of each design. Future versions of IoT4Fun can benefit from miniaturization, robustness, and reliability improvements.

1. Introduction

A generic Rapid Prototyping Tool (RPT) can make high-fidelity prototyping of smart toys faster and easier for creators than using custom hardware solutions for each design. The rapid approach offers to creators more freedom on the editing and testing of features during development cycles [1]. However, smart toys may appear in various shapes and size, such as a plush toy, a doll, a ball, a companion robot, or a wearable gadget. Also. solutions explore different computing which include everything technologies, since Augmented Reality (AR) applications to advances in robotics, wireless connectivity, Artificial Intelligence location-based recognition, and (AI), speech applications. In that way, a research challenge resides on making a generic RPT that can support such a variety of smart toy solutions. In this article, we propose IoT4Fun Toolkit as a generic RPT for smart toys development. In Section 2, we elicit five requirements based on literature and industry mappings to build it [2]. First, a generic RPT for smart toys must promote adaptivity for different interface setup. Second, it must support distributed data collection through connectivity with objects and devices. Then, it must offer multimodal feedback to the users and allow different social and embodied interplays. Finally, it should focus on mitigating potential privacy breaches by limiting *Personal Data* (PD) collection [3].

As a means to demonstrate adequacy with elicited requirements, in Section 3, we first compare IoT4Fun features with other RPTs from literature and industry [1, 4, 5, 6, 7, 8, 9, 10, 11, 12]. In Section 4, we introduce IoT4Fun, a toolkit that combines embedded, modular, and plug-and-play approaches. The first approach aims to embed the same RPT into different physical toys. Modularity permits better distribution of the hardware components and allows creators to decide which modules are essential to their solutions. The plug-andplay approach offers a rapid and easy-to-use experience for the creators to manage the modules. IoT4Fun uses Printed-Circuit Board (PCB) manufacture to favor miniaturization and robustness. It collects real-time motion tracking information, supports wireless communication with devices, and contactless identification of objects or users. Besides, it offers visual, auditory, and tactile feedback and permits programming all play behaviors using Arduino IDE.

In Section 5, we detail how a group of 27 graduate students embedded IoT4Fun into five smart toy prototypes. Students selected RPT modules that were suitable for their designs. Prototypes present a variety of shapes and sizes (e.g., a hula-hoop toy, a plush toy, a hand-sized cube, a large box, and a glove). Solutions either connect with mobile applications or with tagged objects, and programmed behaviors range from openended play to closed-rules. A total of 40 end-users (23 males/17 females) exhaustively tested the prototypes in playtesting sessions. In Section 6, we check for the physical integrity of the RPT modules after playtesting, and battery consumption of each design. We also carry a vulnerability analysis for data security and privacy threats [13], which include simulating attacks to access and configure the communication modules and mobile applications (when applicable). Then, we select security strategies to solve the identified vulnerabilities [14]. Finally, in Section 7, we summarize needed improvements that cover topics on miniaturization, robustness, and reliability.

2. RPT Requirements

Requirements to build a generic RPT for smart toys development must account for a variety of interface and play features that smart toys can have [2]. Interface features for smart toys distinguish types of toy components, connected devices, and peripherals, including their size, symbolic representation, and both connectivity and interactivity aspects. Play features of smart toys vary since general and play purposes to play rules, dynamics, thematic, target audience, among other physical, social, and environmental aspects. In general words, interface features allow a smart toy to acquire, to transfer, and processing real-world data. Data collection serves then to regulate the play features to support human-toy interaction. A smart toy can manage data collection independently, or it can share those capabilities with a connected local device, and sometimes via Cloud, by accessing online services.

Data collection management is what distinguish smart toys from general-purpose toys [13]. The types of data that a smart toy can collect will depend on the interface features that it has (e.g., its physical affordances, embedded sensors, processing capacity, and communication channels). Smart toys may collect a variety of PD from their users to enable playing time. PD include voice recordings for speech recognitionenabled play, facial pictures for emotion detection, and geolocation coordinates to allow pervasive play. It may, however, open opportunities to several privacy breaches for the collection and storage of undesired data. A Non-Personal Data (non-PD) collection approach for smart toys focus on limiting the smart toy to collect types of data that can minimize harm to user's privacy, even in the face of data disclosure [3].

According to the non-PD approach, it is possible to collect the following types of data from both smart toys and users. First, *Non-Personal Identification* (non-PID) consists of identifying either a real or a virtual entity (object or user) without recognizing the entity as an identifiable individual. It can be achieved in the following ways: single, multiple, collective, and state non-PID. Second, *Non-Personal Positioning System*

(non-PPS) is a mechanism that determines the location of a real or virtual entity in a designated space without collecting PD. It varies according to its accuracy and the *Degrees Of Freedom* (DOF). The types of non-PPS that allow both object and user tracking are 2D positioning coordinates, 3D positioning coordinates, angular positioning, relative positioning, *Indoor Positioning System* (IPS), and *Local Positioning System* (LPS) [3].

Finally, once established the object and user tracking strategies used to collect both non-PID and non-PPS, it is possible to estimate motion-tracking data. Moving physical bodies generate a variety of data, including speed and acceleration — external forces such as momentum, gravity, and atmospheric pressure influence these bodies. The current motion-tracking data that can be obtained for smart toy design are relative motion, circular motion, oscillation, momentum, and 3D Kinematics. Hence, a generic RPT for smart toys can benefit from allowing motion tracking data collection to offer a broader range of design opportunities to creators.

Furthermore, how data collection will be managed by a smart toy will depend on the nature of the play features they promote. Play rules are what regulate physical and social dynamics to determine a play state machine [2]. There are two general types of play rules: open-ended rules and closed rules. The open-ended rules introduce open or negotiable rules, which are determined by the users and by the toy's inputs and outputs. In contrast, the closed rules establish predefined rules, which enable the designers to create levels and degrees of challenges that are similar to those used by digital games. Thus, a generic RPT for smart toys must support the implementing of these different play features. Following, we elicit five requirements to build a generic RPT for smart toys.

- *R1*. RPT should support adaptability by design.
- *R2*. RPT should allow distributed data collection.
- *R3*. RPT should offer multimodal user feedback.
- *R4*. RPT should support different play features.
- *R5*. RPT should limit personal data collection.

3. Related Works

In related literature, we classify existing RPTs for smart toys into smart devices [1, 4, 5], AR-based platforms [6, 7, 8], mobile-based platforms [8, 9, 10, 11], and modular toolkits [12]. In Table 1, we detail RPTs from the literature, including related products and licenses, and compare them with the IoT4Fun Toolkit. Each RPT approach has its advantages and disadvantages. Smart devices can be considered smart toys themselves. They are ready to use and play, and usually promotes inter-device connection and embodied interplays. However, they present fixed interface features, which limit creators to only editing the play features. AR-based platforms use cameras to detect objects (e.g., tokens, cards, and toys) by using either marker-based and markerless recognition techniques (i.e., recognition of shape, color, lighting, saturation, texture, and other image descriptors). In addition to cameras, AR-based approach often requires complex setup to support detection and displaying virtual contents, such as mobile devices, *Head-Mounted Displays* (HMD), and *Infrared* (IR) tabletops. Note that AR-based platforms may expose the user's privacy due to the collection of PDs such as facial pictures or videos of the players manipulating the toys. Mobile-based platforms explore multitouch, conductive materials, or contactless technology to detect objects using smartphones or tablets. This approach reduces setup complexity and privacy issues when compared with the

RPT	Approach	Technology	
RaPIDO	Smart	Arduino Mega CU, accelerometer, 4 RGB LEDs, sound chip+speaker (SD	R2, R3,
[1]	Device	card), vibration motor, and battery. Interconnection via RF and RFID.	
Body Bug/	Smart	ARM7 CPU, 512kB Flash memory, 3-axis gyroscope/accelerometer, 6	
Oriboo [4]	Device	RGB LEDs, speaker, vibration motor, steeper motor, touchscreen, and	
		battery. It may connect via RF and is configurable using the USB port.	
Sifteo	Smart	ARM CPU, 8MB Flash memory, 3-axis gyroscope/accelerometer,	R2, R3,
Cubes [5]	Device	touchscreen, and battery. Offers and RF (2.4 GHz) and NFC connectivity.	R4, R5
Sketching	AR-based	Standard PC, high-definition camera and artificial illumination to enable	R1, R4
LEGO [6]	platform	color detection, a translucent LEGO plate, and LEGO bricks' widgets.	
NIK Vision	AR-based	Standard PC, IR camera, IR diffuse illumination, microphone, speakers,	R1, R4
[7]	platform	monitor/TV, video projector and mirror to display contents in a translucent	
		table, and wooden tokens attached to printed fiducial markers.	
Touch	AR/Mobile-	3-coordinates multitouch patterns recognition, and Android application	R1, R4,
Token [8]	based plat.	(touchscreen) or TUIO app (IR tabletop).	R5
TriPOD [9]	Mobile-	3-coordinates recognition template to identify manipulatives in the	R1, R4,
	based plat.	touchscreen using conductive materials (capacitive pins and cooper).	R5
Widgets [10]	Mobile-	3-coordinates recognition template to identify manipulatives in the	R1, R4,
	based plat.	touchscreen using conductive materials (tinfoil and conductive tape).	R5
Flexibles	Mobile-	3D-printed objects made of conductive polymer layers (cPLA) and	R1, R4,
[11]	based	deformable dielectric elastomer layers (NinjaFlex TPU), and mobile	R5
	platform	application.	D / D
MakeWear	Modular	Wearable kit made of 32 modules among sensors, receivers, actuators,	R1, R3,
[12]	I OOIKIT	and modifiers to code behaviors. It does not offer connectivity.	R4, R5
Osmo's	AR/MODILE-	Osmo mirror attached to the tablet's front-camera that allow object	R1, R4
	Dased plat.	detection, <i>Osmo</i> base, manipulatives, and mobile application.	
Volumique s	NODIIe-	3-coordinates recognition patent to build sell-capacitive manipulatives for	R1, R4,
	Mobilo	Contactions identification (NEC/PEID) board with a grid antenna to	
license	hased plat	identify the relative position of tagged objects	R_1, R_2, R_3
Nintendo's	Smart	lawConcontroller (motion tracking denth sensor vibration) cardboard	R1 R2
Labo	Device	kits and Switch game console. It offers NEC and Bluetooth connection	B4 B5
Sphero	Smart	Spheric robot embedded with several motors, and motion tracking	R2 R4
SPRK+	Device	proximity and lighting sensors. Connects with the application via	B5
	Device	Bluetooth	110
Ozobot EVO	Smart	Semi-spheric robot embedded with motors, and proximity and optical	B1, B2,
	Device	sensors. Offers NFC and Bluetooth connectivity.	R3. R4
SAM labs	Modular	Modular cubic components (sensors, actuators, motors, connectors,	B1, B2,
	Toolkit	jumpers, protoboards) to assemble with maker kits, and programmable by	R3, R4,
		application. Connects with application via Bluetooth to support coding.	R5
LittleBits	Modular	Modular magnetic components (sensors, actuators, motors, connectors,	R1, R2,
	Toolkit	jumpers, protoboards) to assemble with maker kits, and programmable by	R3, R4,
		application. The kit has RF or Bluetooth modules to support connectivity.	R5
loT4Fun	Modular	8 PCB modules are attachable with flex ribbon cables (Arduino Mini-Pro,	R1, R2,
	Toolkit	10 DOF IMU sensor, 3 RGB LEDs, speaker, vibration motor, battery, and	R3, R4,
		USB recording module). Connection modules offer short-range	R5
		(NFC/RFID), and long-range protocols (Wi-Fi or Bluetooth/BLE).	

AR-based one. However, these platforms are also limited to promote token-tabletop interaction (e.g., placing tokens on the touchscreen). Finally, modular toolkits consist of a collection of sensors, actuators, communicators, and other electronic circuits that are attachable and programmable. They offer more freedom to the editing of both play and interface features since they permit creators to select components that best fit their projects. Still, the level of programmability, size, and distribution of modular components can limit its adaptability features. Thus, an adequate hardware specification is essential when implementing a modular toolkit.

In the smart toy industry, we find technologies that are comparable to the RPTs literature. First, both mobile and AR-based patents and licenses have demonstrated their effectiveness to adapt to different solutions, but they are too focused on the manipulation of tokens. Educational programming toys, usually smart devices or modular kits offer more freedom to edit features. However, their programmable functions are for children, so they are either limited or not open source. For example, SAM Labs, Sphero, and Ozobot use mobile applications for coding. LittleBits had an Arduino kit, which was discontinued to promote a new proprietary coding application that uses JavaScript blocks. LitteBits is a play product that focusses on supporting children in their "maker" projects and not on embedding high-end smart toy solutions. Kits values range from USD 39.95 to 299.95, including more expensive kits for schools. In August 2019, Sphero has acquired LittleBits, and together they released kits compatible with the spheric robots.

4. IoT4Fun Toolkit

Our goal is to propose a new generic RPT for smart toys that complies with the five elicited requirements. In this section, we describe how the IoT4Fun toolkit meets them all. First and foremost, R1 concerns that to achieve adaptability, a generic RPT must fit into different designs without compromising its usage and essential functionalities. Thus, it is essential to favor the miniaturization of hardware components. PCB manufacture supports the development of custom circuit solutions that can minimize the use of wires, resistors, capacitors, and inductors. In that sense, the implementing of the RPT can benefit from exploring PCB manufacture. Then, to achieve better adaptability, the RPT can incorporate modularity. It can support a better distribution of the hardware components into the physical restricts of each smart toy. Moreover, modularity allows the creators to select only the hardware components that they need for each design.

Also, by manufacturing the modules "plug-and-play" can help to deliver an easy to use end-user development RPT for the creators.

Second, R2 accounts that adequate RPT should support distributed data collection. Smart toys usually embed hardware components like sensors, actuators, microprocessors that introduce limited and computational capabilities, which can compromise full on-board processing. Thus, smart toys often connect more powerful computing devices like with smartphones, tablets, game consoles, or companion robot components to share those capabilities [13]. Communication channels are what support transferring data between the smart toy and other interface components. A generic RPT can benefit interoperability by supporting both short-range and long-range communication protocols. Alternatives for short-range protocols are Near-Field Communication (NFC) and Radio Frequency Identification (RFID). As for longrange communication, options include Wi-Fi, Bluetooth, and other Radio Frequency (RF) protocols.

Third, smart toys need to provide continuous feedback to the player's actions [1]. Therefore, R3 defines that a generic RPT must support the design of multimodal user feedback. Modalities may include visual, auditory, and haptic feedback. Feedback must also respect the distributed aspects by appearing in both smart toy and connected interface components. Visual feedback in the toy component can use low-resolution displays such as LED panels. Auditory feedback can use essential solutions like buzzers or small-sized speakers enabled to reproduce since 8-bit to MP3 audio files. Then, haptic feedback may include the use of small vibration motors. Feedbacks that are more sophisticated can use connected devices, such as by using their highresolution displays and speakers.

Fourth, the R4 advocates that RPT should support the implementing of different play features [2]. Smart toys can mix play features from traditional toys and games, and such a combination may result in multiple social and physical play modalities. Play modalities include social competition, collaboration, parallel play, physical manipulation, and full-body interaction. The play rules regulate all these modalities (i.e., closed or open-ended). Thus, the RPT must support means to creators fully implement the rules and behaviors that are essential to their designs. The Arduino IDE offers a cross-platform application with free-software licenses, which is compatible with Arduino boards and several third-party boards. It supports programming languages C and C++, many libraries, and has an extensive development community. For example, some libraries support integrating the hardware functionalities to interact with 3D or 2D environments and applications.

Finally, microcontroller's technology is still facing challenges to ensure sufficient security against data disclosure due to its limited processing capabilities. For the sake of security, the smart toy component should only collect non-PD [3]. Then, any PD must be gathered by connectivity with secondary components, such as since they may offer an adequate infrastructure for data security. As determined by the non-PD approach, motion-tracking data can supply creators with a range of design possibilities. Sensors like accelerometers, gyroscopes, magnetometers, and barometers can collect the required motion tracking data. Besides, the RPT must support LPS and non-PID collection by combining both long and short-range wireless communication protocols (e.g., NFC/RFID, Bluetooth). Note that by limiting PD collection does not minimize the need for adequate privacy policies and data security approaches [13, 14]. In that way, all communication between the interface components must comply with data security models and universal standards.

In Figure 1, we introduce the IoT4Fun Toolkit as a generic RPT for smart toys that meets all five elicited requirements. It consists of eight individual and attachable PCB modules. These are one hub module attached to a motion-tracking sensor: three output modules for haptic, visual, and auditory feedback; two connectivity modules for short and long-range communication; a battery source module: and a recording module to ease configuration. All modules are attachable to a central hub module using plug-and-play 6-pin flat flex ribbon cables.



Figure 1. IoT4Fun Toolkit for smart toys.

5. Smart Toy Prototypes

The IoT4Fun toolkit was experienced during coursework of the graduate program in Computer Science in the Brazilian *Federal University of Pernambuco* (UFPE), between August to December 2018 [15]. The coursework lasted for 16 weeks, and students started working with the toolkits in the 8th week. A total of 27 creators participated in this assessment, among 15 M.Sc. students and 12 Ph.D. students with multidisciplinary backgrounds on

Computer Science, Engineering, Design, and related areas. The class results permitted to embed our RPT into five smart toy solutions, as we show in Figure 2. The five solutions incorporate different play rules. Two explore closed rules and single-play by connecting the smart toys with mobile applications and the other three offer open-ended rules for pervasive multi-play. The teams selected the modules according to the set of play rules and planned behaviors of their projects — Table 2 shows how each smart toy solution uses the modules of the IoT4Fun Toolkit.

Cube Music in Figure 2 is a smart toy inspired by the toys SIMON and Rubik's cube. It introduces closedrules for single-play interaction. The smart toy is used to play with a music application, in which the player has to memorize the musical sequence, and then, replicate it by flipping the cube. At that moment, the Android application regulates the play rules by recording each inserted note so that the player can complete the entire music. Students used black Ethylene Vinyl Acetate (EVA) to build the body and attached a colorful geometric shape in each cube's face to represent parts of a multiple non-PID (e.g., a red star, a blue circle, a green square). Cube Music uses the following IoT4Fun modules. First, it connects with the application using the BLE module. It uses the motion-tracking sensor attached to the hub module to collect 3D positioning information to estimate the location of the cube's upper face. Then, it sends the collected non-PPS data to the connected application to regulate the playing turns. The application augments the visual and auditory feedback assigned to each face. Thus, the cube only provides the haptic feedback to the user using the vibrating motor module. Notably, students made all feedback accessible to visually impaired users. During playtesting sessions, the team provided a headphone to the players to better hear the sounds. Cube Music uses the battery module as the only power source.

Cobi in Figure 2 is a smart toy that resembles the Sesame Street's character Cookie Monster. The Frisbee toy and the open-ended rules inspire the cookie's disks by the Japanese game Kan Jam. The gameplay offers open-ended rules for parallel multi-play competition, in which each player attempts to launch the cookie disks by aiming it at the target. The smart toy can distinguish when the cookie is inside its mouth and when it hit its external body. Students used green EVA to build the body, and milky white acrylic to amplify the visual feedback in the eyes. Inside the toy, a cardboard ramp assists the disks to slide down through its mouth. Cobi uses the NFC module to detect the cookie's disks, which are attached to NFC tags. They located the NFC module in the bottom of Cobi's internal ramp. Moreover, Cobi uses the motion-tracking sensor to estimate when the disk hits the toy in the outside area. In that way, the

smart toy augments appropriate visual and auditory feedback using the modules to the two possible outcomes: the NFC module detects the cookie's disk, or the motion sensor collects momentum data when hit by the disks on its outside. As a result, the visual feedback module transforms the blue RGB LEDs into green when the cookie is inside or to red when it hits the body. Concomitantly, Cobi emits auditory feedback using the 8-bit speaker module to each state; these are the waiting, complaining, and eating sounds. Note that in Table 2, students decided to use an external power source to keep Cobi active for more time during playtesting sessions. Still, the smart toy can work connected to the battery module independently.

Open-ended ball games like Hot Potato influenced Magic Potato toy in Figure 2. The smart toy is composed of a plastic ball that embeds the toolkit modules, and coming out of the ball, a larger flat ribbon cable, attached to the visual feedback module, passes it through a cloth leash that mimics a bomb's wick. A plush-toy shield shaped as a potato-like character attached to the "bomb" then covers the whole body. The plush-toy was handmade and filled with acrylic stuffing and cloth. The only regulated rule consists that players must keep the toy in constant movement so that it will not "explode" in the user's hand. Therefore, players may tease each other by holding the toy before passing it away. Magic Potato uses the motion-tracking module to measure the 3D kinematics movements that it uses then to regulate the play rules. The state non-PID updates from safe to the bomb according to the collected movements, which are then, augmented by visual, haptic, and auditory feedback modules. The auditory module plays an 8-bit song similar to a lullaby; then, the song intensifies speed and volume when its state change. If the bomb explodes, the song is resumed, and the smart toy resets automatically for the next turn. Similar behavior occurs to the RGB LEDs, which transforms from green to red, and the vibrator motor intensifies its oscillation rate. Table 2 shows that Magic Potato uses extra batteries to work correctly. Once it is the solution that uses all output modules, it requires higher battery consumption. Hence, the team adapted the battery module to fit six Li-Ion batteries working in parallel.

Moreover, the Magic Potato is the only solution that does not offer connectivity features. However, the solution suits to multi-play activities better by not limiting the size of the group.

Hula-hoop Hero in Figure 2 resembles Nintendo's Guitar Hero and Just Dance closed-rules by exploring full-body interaction. The user plays with the smart toy according to the instructions provided by a connected application. The user can move the toy in different parts of his/her body (e.g., right and left arms, waist, and neck). Then, it can distinguish vertical from horizontal positioning to validate the movements during play. Hula-hoop Hero adapts a traditional plastic hula-hoop, which is filled with foam to protect the toolkit modules. Plus, they covered it with metallic adhesive paper. Students assembled selected module altogether. Furthermore, the body was filled with extra foam in some parts to achieve counterbalance. The hula-hoop toy uses the motion-tracking sensor to collect angular frequency data in the yaw, pitch, and roll axis to measure the cyclic movements. Then, it uses the BLE module to send this information to the connected application. The Android application augments all visual and auditory feedback by displaying animations and playing songs. The playable character can transform its virtual hula-hoop prefab into three states, namely right arm, left arm, and waist. The application, then verifies if the smart toy is replicating the required movements to determine the player's performance. Note that Hula-hoop Hero requires challenging physical skills to use it. In that sense, the designers adapted the closed rules to replace the waist movement by using the smart toy in the user's neck if needed.

Zombie-tag, in Figure 2, is a smart toy inspired by the open-ended game *tag*. It offers multi-play experiences and parallel competition to the players. One player wears the smart toy, the zombie-glove, and then he/she uses it to tag the other four players wearing bracelets attached to NFC tags. As the primary open-ended rule, the zombie glove uses the NFC module to collect each bracelet's single non-PID in order to tag the players. There are three sets of open-ended rules that can be regulated by the smart toy: the survivor, 60 seconds, and secret modes. In survivor mode, the zombie player



Figure 2. Smart toy solutions embedding the IoT4Fun modules and user testing them.

can tag the others freely; in the 60 seconds, he/she has to capture at least one player until the time is up; and the

secret mode randomly selects the order of the players that must be tagged. Players can select the play modes by reading NFC playing cards with the smart toy; it will automatically select the desired play mode. The Zombie-tag glove adapted a *Halloween* costume. It has two parts made of rigid plastic; then, the two parts were hot glued on synthetic leather, which is filled with foam and uses velcro to assemble them. The glove embeds the hub, visual, and battery modules in the upper part of the glove and positions the NFC module in the bottom part. Besides, in the upper part, a triangular box made of

milky white acrylic amplifies the RGB LEDs light to provide better visual feedback. The glove uses the visual feedback according to each play mode; it can either guide the order that the player should tag the other players or can validate when each player is captured. It also uses auditory feedback to augment when a player is tagged. However, due to the size of the 8-bit speaker, the toolkit was adapted to use a buzzer instead. Moreover, note that Zombie-tag is the only smart toy that does not collect motion tracking.

6. Technical Assessment

The IoT4Fun Toolkit allows the design of different smart toy solutions by permitting the creators to select which modules are suitable for the play rules of each project. In Figure 2, a group of 40 end-users (23 males and 17 females) tested the five smart toy solutions during a playtesting event as part of the coursework [15]. Participants were students of Physical Education and Sports Department, which was likely fit for the intended target audience. They age from 18 to 23 years old, plus one participant age 50, and one was a wheelchair user. Between 8 to 15 users tested each prototype, and 26 volunteers played with more than one. Teams presented to volunteers the overall functions of each prototype and recorded their play experiences for further analysis (e.g., videos, pictures, transcripts, and questionnaires responses). All embedded RPTs worked adequately and presented as robust enough to allow 90-120 minutes of playtesting sessions. However, there

were reports from the teams about minor defects and overall battery consumption. In that sense, this technical assessment consists of three types of testing: (a) functional integrity, (b) battery consumption, and (c) data security and privacy vulnerabilities. First, (a) functional integrity tests used the Arduino IDE to check core functionalities of each module, and when necessary, a multimeter checked for punctual defects of the PCBs. In overall, all third-party components are fully working after the playtesting sessions. However, some parts of the manufactured PCBs were damaged after testing (e.g., some circuit rails and connectors). It happened due to collisions during playtime, and by the way, that teams handled the modules during development. Damages in the PCBs appear in two visual modules, one auditory module, and one BLE module. It may suggest that the PCB manufacture should be better performed to conquer adequate robustness.

Second, the (b) battery consumption tests look to estimate the battery autonomy of each project to support further improvements in our RPT. The instantaneous current analysis was performed using the Current Shunt method [16]. It consists of measuring the current of consumption, second by second, and according to the active time of each solution. The sensor ACS 712 5A was attached in series to the hub module to measure the current circuit of connected modules. In Figure 3, we compare the results of the current consumption for all projects by calculating the average battery consumption of each one. Once knowing the average consumption, it is possible to estimate the battery autonomy of each project in Table 3. Battery autonomy calculation consists of the relation between 80% of the total battery capacity and the average battery consumption. The 80% rate simulates the behavior of a lithium polymer battery since generally in this type of battery, the circuit stops running before the voltage is entirely over. The total capacity of the battery module is 350mAh; it uses the battery LP702035 3.7 V. Recall that the Magic Potato uses an adapted battery module with six batteries working in parallel; thus, we estimate its total capacity to 2100mAh. Moreover, in Table 3, we show a separate current consumption analysis of the three Zombie-tag playing modes, namely, (A) survivor, (B) 60 seconds, and (C) secret mode. The Zombie-tag has a single script,

Module	Cube Music	Cobi	Magic Potato	Hula-hoop Hero	Zombie-tag
Hub	\checkmark	\checkmark	\checkmark	\checkmark	Doesn't use motion
NFC	×	\checkmark	×	×	\checkmark
BLE	\checkmark	×	×	\checkmark	×
Visual	×	\checkmark	\checkmark	×	\checkmark
Auditory	×	\checkmark	\checkmark	×	Uses a buzzer
Haptic	\checkmark	\checkmark	\checkmark	×	×
Battery	\checkmark	External power	Extra batteries	\checkmark	\checkmark
Recording	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark

Table 2. IoT4Fun Toolkit usage according to each design.

and it uses the NFC playing cards to define the starting line of the script, which determines the selected playing mode. The goal was to compare the battery autonomy of the same modules using only different parts of the programming script. All play modes make use of the NFC and visual modules. The average consumption of each play mode demonstrates that (B) 60 seconds mode consume more power than the other two, therefore, resulting in battery autonomy loss. That happens due to the RGB LEDs are always turned on in the 60 seconds mode, while in the other modes, they serve as feedback for trigged play actions (e.g., it blinks when catches one player or to show the order to catch the next player).



Figure 3. Comparative Battery Consumption.

Table 3. Battery Autonomy Results.					
Smart Toy Solution	Average Battery Consumption	Battery Autonomy			
Cube Music	55mA	305' 27"			
Cobi	80mA	210'			
Magic Potato	168mA	600'			
Hula-hoop Hero	33.5mA	501'			
Zombie-tag A	79mA	216' 39"			
Zombie-tag B	98mA	171' 25"			
Zombie-tag C	76mA	221'1"			

Results suggest that battery consumption is mostly related to implementation decisions. Although the output modules do consume more power than the BLE module, for instance, both Cobi and Magic Potato embedded visual and auditory modules, but the second

solution demanded two times of battery consumption. Magic Potato needed more consumption (168mA) than Cobi (80mA) since it requests continuous sound feedback when active, while Cobi requests by demand (e.g., when it read a disk or creates motion tracking). Finally, vulnerability analysis points out the main risks for data security and privacy of each project. Shasha et al. [13] define a vulnerability taxonomy for smart toys that have connected mobile applications. They classify those threats into physical and remote access types. In Table 4, we list the types of threats that apply to our RPT and for each smart toy solution. The IoT4Fun Toolkit is always sensitive to Unauthorized-config-physical threat since it offers a USB recorder to update the hub module. It is intended to make the programming and updating of contents faster and easier, but it can be used for malicious configuration since it does not require any authentication.

The other types of threats are dependent on RPT implementation. For example, IoT4Fun is sensitive to the Unauthorized-config-nearby threat, but both implemented mobile applications do not support configuring their smart toys through them (i.e., Cube Music and Hula-hoop Hero). However, none of the solutions employed security standards to support local data protection, and the two connected applications permitted both *tampering* of information and *denial of* service threats. Note that the vulnerabilities items do not cover scenarios that use NFC communication, in which occurs the exchange of data between two toy components (e.g., Cobi and the cookie's disks). Thus, the present article adds a new item named Insecure-NFC-practice, which is similar to the Insecure-Bluetooth-practice, and the Unencrypted-commchannels item was adapted to cover NFC communication. Hence, to make the vulnerability analysis comprehensive, it includes testing the security of both BLE and NFC modules.

First, the Android app, named BLE Console, was used to examine the security of the BLE connection. As the parameters for a secure BLE connection, we considered if it requires user authentication and if the MAC address dynamically changes. As a result, both

-	-				-	-
Vulnerabilities	Cube Music	Cobi	Magic Potato	Hula-hoop Hero	Zombie- tag	loT4Fun Toolkit
Unauthorized-config-physical	\checkmark	\checkmark	\checkmark	\checkmark	√	\checkmark
No-local-data-protection	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	*
Unauthorized-config-nearby	×	×	×	×	×	*
Insecure-Bluetooth-practice.	\checkmark	×	×	\checkmark	×	*
Unencrypted-comm-channels	\checkmark	\checkmark	×	\checkmark	\checkmark	*
Denial of Service	\checkmark	×	×	\checkmark	×	*
Tampering	\checkmark	×	×	\checkmark	×	*
Insecure-NFC-practice	×	\checkmark	×	×	\checkmark	*
		*Some vulne	erabilities may	/ appear depend	ling on the im	plementation.

Table 4. Data Security and Privacy Vulnerabilities of the IoT4Fun Toolkit and by Project.

Cube Music and Hula-hoop Hero pairs without authentication and present fixed MAC addresses. The BLE Console app allows accessing the smart toy information that includes all non-PD sent by the serial port (i.e., motion-tracking information), among manufacturer's information such as model, serial number, and firmware revision. Similarly, the NFC Tools app supports to examine the security of the NFC connection. Although NFC is a safer technology than other protocols for authentication, it still opens breaches for data disclosure. The NFC communications in both Cobi and Zombie-tag were not encrypted. It gives to the attacker the opportunity to get information from the tags (cookie's disks and bracelets) or the reader (Cobi and the zombie-glove). Using the NFC Tools app, it is possible to access recorded non-PID on the NFC tags, and there is an option to limit the tag information for further readings since it can rewrite the tag information.

In summary, the modularity approach of our RPT was efficient to achieve adaptability. Still, some projects demanded custom changes. For instance, the Zombietag glove replaced the speaker for a buzzer due to its size. Magic Potato needed more batteries to work correctly, but it was account on their implementation. Among the critical challenges, the Hula-hoop Hero struggled to fit the components inside such a thin body. They had to assemble the sensors altogether, which required them to stuff foam inside the toy to conquer counterbalance. The flex cables are efficient solutions for the attachment of the module. Some projects required larger cables to distribute the modules better, which was an easy solution to take. Alternatives in the future versions may include stack the modules or try magnetic attachment.

Moreover. results suggest the for need improvements in miniaturization, robustness, and reliability. First, to improve miniaturization, the next version of our RPT must incorporate all sensors and components in the PCB design. By incorporating the shields, it will turn possible to reduce the size of all modules. For example, the NFC module has 40 x 43mm, while its shield dimensions are 6 x 6mm. Also, future versions can replace some components; in particular, the 8-bit speaker is more prominent than desired and produce poor-quality feedback. Alternatives for the auditory module include using an MP3 module attached to a small-size speaker, which has internal memory to store the audio files. Also, the BLE module can be exchanged by as ESP32 shield (it supports alternating between Wi-Fi and Bluetooth).

Second, concerning robustness, our RPT was manufactured by PCB design, which supports more freedom to distribute the components, and minimizes the need for wires, resistors, and capacitors. However, the quality of our PCB design was not sufficient to secure the functionalities and physical integrity of the modules after extensive usage. Some modules suffered minor damages during playtesting sessions. In that sense, to improve its robustness, a third party will professionally manufacture future IoT4Fun modules. Moreover, protective cases may help to secure the hub module and other components; they can use hard plastic, acrylic, or flexible materials to reduce external impact during collisions.

Finally, to improve reliability and circumvent the identified vulnerabilities, we select the following data *Security Requirement* (SR) items for smart toys, which were proposed by de Carvalho and Eler [14].

- *SR5*. Communication between physical toy and mobile device must use a protocol that allows authentication and authorization mechanisms.
- *SR7*. Configuration file integrity must be maintained and verified in every mobile app play session.
- *SR8*. Every communication in toy computing environment must use cryptographic mechanisms.
- *SR15*. The mobile app must monitor and limit database growth.
- *SR10.* The *Database Management Systems* (DNS) must provide security mechanisms against external modification of stored data.
- *SR21*. The smart toy should avoid exposing unnecessary information once implemented.

Once again, the SR items fulfill smart toys that have connected mobile applications. Thus, we adapted the SR5 to cover NFC communication. It may suggest that taxonomy of vulnerabilities [13] and SR items [14] must be expanded to cover a broader range of smart toys solutions [2]. In Table 5, we relate the SR items with potential threats. In future assessments, it is essential to ensure that creators consider those SR items since the design planning. They must assure security standards for data encryption and build access control including parental mechanisms, control and management of privacy policies. A development framework can favor secure hardware and software integration. It may assist creators in both implementing the SR items and programming behaviors that are the most cost-efficient for battery consumption.

Table 5. SR for the Potential Vulnerabilities.				
Vulnerabilities	SR items			
Unauthorized-config-physical	SR5			
No-local-data-protection	SR21			
Unauthorized-config-nearby	SR7			
Insecure-Bluetooth-practice.	SR5			
Unencrypted-comm-channels	SR8			
Denial of Service	SR15			
Tampering	SR10			
Insecure-NFC-practice	SR5			

7. Conclusion and Future Work

RPT approaches for smart toys include smart devices, modular kits, AR-based, and mobile-based solutions. In the present article, we offered, implemented, and tested the IoT4Fun Toolkit, a generic and innovative RPT for smart toys of different shapes and size. We obtained success with adaptability by combining the modular approach with embedded and "plug-and-play" features that permitted creators to select only modules that were essential to their designs. Multidisciplinary teams experienced our RPT and successfully created five high-fidelity prototypes of smart toy solutions. Solutions varied in shape, size, play rules, and interaction modalities. Prototypes were robust enough to allow exhaustive playtesting with a group of 40 end-users.

Technical assessment checked for the integrity of modules, and future versions require more robustness to protect the components from resisting physical collisions. Moreover, enhancements in miniaturization and exchange of hardware components can benefit its adaptability and increase the quality of multimodal feedback. Battery consumption and security vulnerabilities were both dependent on technical implementation. Teams implemented their solutions using the Arduino IDE, and the same modules resulted in more or less battery autonomy. Teams chose not to encrypt communication channels, and they were not careful with authentication. Therefore, it is essential to build a reliable development framework to assist the creators in delivering the IoT4Fun Toolkit best potential. Ultimately, the current version of the RPT has achieved its research goals, and future versions can become commercially available to support students and professionals in the smart toy industry.

8. References

[1] Soute, I., T. Vacaretu, J.D. Wit, P. Markopoulos, Design and evaluation of rapido, a platform for rapid prototyping of interactive outdoor games. ACM Transactions on Computer-Human Interaction, 24, 28, 2017.

[2] de Albuquerque, A. P., and J. Kelner, Toy user interfaces: systematic and industrial mapping, Journal of Systems Architecture, 97, pp. 77-106. 2019.

[3] de Albuquerque, A. P., and J. Kelner, Non-personal Data Collection for Toy User Interfaces, Proceedings of the 52nd Hawaii International Conference on System Sciences, 2019.

[4] Merrill, D., E. Sun, J. Kalanithi, Sifteo cubes, CHI'12 Extended Abstracts on Human Factors in Computing Systems, ACM, pp. 1015–1018, 2012 [5] Máquez Segura, E., A. Waern, J. Moen, J., C. Johansson, The design space of body games: Technological, physical, and social design. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, pp. 3365–3374, 2013.

[6] Gohlke, K., M. Hlatky, M., B. de Jong, B, Physical construction toys for rapid sketching of tangible user interfaces. Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction ACM, pp. 643–648, 2015.

[7] Marco, J., E. Cerezo, S. Baldassarri, Tangible interaction and tabletops: new horizons for children's games. International Journal of Arts and Technology, 5, 151–176, 2012.

[8] Appert, C., E. Pietriga, E. Bartenlian, R. M. Gonz'alez, Custom-made tangible interfaces with touchtokens, Proceedings of the 2018 International Conference on Advanced Visual Interfaces, ACM, p. 15, 2018.

[9] Di Fuccio, R., G. Siano, A. De Marco, Tripod: A prototypal system for the recognition of capacitive widget on touchscreen addressed for montessori-like educational applications. Recent Advances in Information Systems and Technologies, Springer International Publishing, pp. 664–676, 2017.

[10] Bech, C., A. H. Bork, J.B. Memborg, L. S. Rosenlund, M. Kraus, The effect of interacting with two devices when creating the illusion of internal state in passive tangible widgets. Interactivity, Game Creation, Design, Learning, and Innovation, Springer, pp. 197–204, 2016.

[11] Schmitz, M., J. Steimle, J. Huber, J., N. Dezfuli, M. Muhlhauser, Flexibles: Deformation-aware 3d-printed tangibles for capacitive touchscreens. Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, ACM, pp. 1001–1014, 2017.

[12] Kazemitabaar, M., McPeak, J., Jiao, A., He, L., Outing, T., Froehlich, J. E. Makerwear: A tangible approach to interactive wearable creation for children. In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, ACM, pp. 133-145, 2017.

[13] Shasha, S., M. Mahmoud, M. Mannan, and A. Youssef, Playing with Danger: A Taxonomy and Evaluation of Threats to Smart Toys, Internet of Things Journal, IEEE, 2018.

[14] de Carvalho, L. G., & M. M. Eler, Security Requirements for Smart Toys. ICEIS, pp. 144-154, 2017

[15] de Albuquerque, A. P., Kelner, J., Hung, P. C. K. Humancentered Design Tools for Smart Toys. In Proceedings of 9th IEEE International Symposium on Cloud and Service Computing, (to appear), 2019.

[16] Nakutis, Z. Embedded systems power consumption measurement methods overview. MATAVIMAI, 2 (44), pp. 29-35, 2009.