Learn-o-Bot: Educational Application Using Augmented Reality and Sphero SPRK+

Andreea Camelia Tripon
Computer Science Department,
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
camelia.tripon1@gmail.com

Teodor Stefanut
Computer Science Department,
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
teodor.stefanut@cs.utcluj.ro

ABSTRACT

This paper presents a mobile application with educational purposes designed for kindergarten children which is called Learn-o-Bot. The application uses augmented reality and the Sphero SPRK+ robot to create an appropriate environment for learning.

The purpose of Learn-o-Bot is to provide preschoolers with means to acquire basic knowledge through a fun game which is easy to understand and use. The game consists of five levels with gradually increasing difficulty. Each level's scene is augmented with appropriate virtual objects and the child should control the robot to select the right objects and reach level's goal.

Finally, this application opens a new perspective to different learning methods, using robots, mobile devices and virtual learning materials.

Author Keywords

Augmented Reality; Sphero; Education; Mobile applications.

ACM Classification Keywords

H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

General Terms

Human Factors; Design.

INTRODUCTION

Kindergarten children have a broad capacity to accumulate knowledge, but they also lose their focus very quickly. Therefore, they need an attractive learning environment, which is able to keep them interested and to provide them with the proper knowledge for their age. We propose this augmented reality game in which the Sphero SPRK+ robot is a bridge between the real world and a virtual one, designed towards learning purposes.

Traditional Educational Methods

The classical educational methods are still very popular in kindergartens and consist in involving the child in activities like drawing, singing or solving simple tasks. The main disadvantage is that children get distracted before finishing all the tasks and that these methods require an adult, an instructor to assign activities and supervise children's progress.

Technology in Education

Using technology in kindergartens can bring many benefits to the learning process, making it both fun and interesting. For example, children can listen to an alphabet song, watch movies or cartoons, use tablets or smartphones to play educational games.

Children also love toys, and with the increased development of technology, they prefer toys like tele-guided cars or robots. Kids these days already know how to use a smartphone [1], a laptop and how to control a tele-guided car, therefore a ball controlled by the phone is a good choice to draw their attention

Augmented Reality and Education

Augmented reality (AR) allows users to view virtual objects on top of the physical world observed through the mobile device's camera.

In education, AR would bring numerous advantages and there are infinite ways of using it. It can make every classical learning activity way more interesting. For example, instead of hearing stories and viewing pictures of wild animals, the child can see them in 3D with proper sounds and animations. Math can become funnier if, for example, a card with a number is augmented with the same number of animals as in the game Math Alive [2].

In the following sections, we will present the Sphero SPRK+ robot, the project components, algorithms and the game from a user perspective (each level of the game). The paper will end with the conclusions and future development.

SPHERO SPRK+

Sphero SPRK+ [3] is a ball-shaped robot designed for educational purposes which can be controlled by a mobile device or computer. The robot can move in any direction with the selected speed and has 2 RGB LEDs, feature which make this toy even more attractive for kids. One important design feature is that Sphero's shell is transparent and kids can actually see its electronics (see Figure 1).

Hardware specifications

The robot is equipped with accelerometer and gyroscope sensors, which are useful to determine collisions, measure velocity or determine robot's orientation.

Figure 1. Sphero SPRK+ components [4]

The communication between Sphero and the controlling device is made through Bluetooth. Sphero SPRK+ uses Bluetooth 4.0, known as Bluetooth Smart or BLE. This feature is the main upgrade from the previous version, Sphero SPRK, which required additional steps in order to establish a connection. The robot can be controlled from a distance up to 30 m.

Software specifications

Sphero's creators offer support for different platforms as Android, iOS, Windows, Unity, through a Software Development Kit (SDK). These libraries are used to send proper commands to the robot. Sphero understands commands such as: driving in a direction with a selected speed, changing the color of the LEDs or changing the orientation. The SDK also handles connection lifecycle and can receive asynchronous information from the sensors.

RELATED WORK

The main features of the current application imply using Sphero and AR for educational purposes. Since Sphero itself was designed for learning, there is a large variety of related applications which are used in this scope.

The most significant is the Lightning Lab [5] application which was designed by the same company as the robot. The application contains a full set of instructions to control the robot and read its state. It also has a feature that allows users to develop small programs that can be executed by the robot. These programs can help users understand basic programming principles as conditional blocks or loops.

Another example where Sphero is used as an educational mean is presented in article [6]. Since Sphero offers free SDKs to allow developers design their own programs, the authors are using this particularity to teach students basic Object Orientated Programming (OOP) principles. The paper presents a study where university students, with no background in programming, can learn basic C# and programming principles by designing a program which controls the Sphero robot. The students claim that this learning method is more interesting and effective because they can see the practical use of programming by controlling Sphero's moves.

Sphero and AR meet in another application made by the creators of Sphero: The Rolling Dead [7]. The gameplay consists in using Sphero as the main character in order to defeat the zombies that are spawning in an augmented environment.

SMAUG is a collaborative AR game that also uses Sphero as the main character. The game is presented in article [8], where the authors show that Sphero's control can be shared to complete more complex tasks. The environment is augmented with objects like diamonds, dragons or treasures and Sphero has to collect or avoid these virtual objects to complete the level's goal.

Quiver [9] is a game which uses augmented reality to create living drawings. It is designed for children to explore their creative aptitudes. The game consists in printing and coloring the drawings and then using the app to see the drawing coming to life. The interesting part is that the colors used in the real drawing are also used for the augmentation, giving the artist a special sense of ownership. Also the characters are animated and the user can interact with them by tapping on the screen and selecting different actions. The purpose of this game is to make coloring more fun, interesting and to allow the child to interact with the drawings.

Zoo-AR [10] is another educational augmented reality game where children can use custom pictures with the animals to see them in 3D. Although it does not use Sphero, the application is a good educational app example because it is a fun way to learn about animals and users can interact with them or take screenshots. It is useful to see how the animals would behave in real life, without actually being in their natural habitat.

PROJECT STRUCTURE AND DEPENDENCIES

The application is made using Unity development environment. Unity permits to create cross-platform games, but this application is currently available only for Android devices.

Software Dependecies

Figure 2 shows a diagram of all the libraries and dependencies used to create this application. These libraries are presented as follows:

1. Sphero Android SDK [11] – a library which is used to send commands to the robot. Since Bluetooth 4.0 is a newer version, the open source SDK is only in beta version and is not available for Unity. To overcome this obstacle, we created an Android plugin for Unity which acts as a bridge between Unity procedures and Java methods provided by the

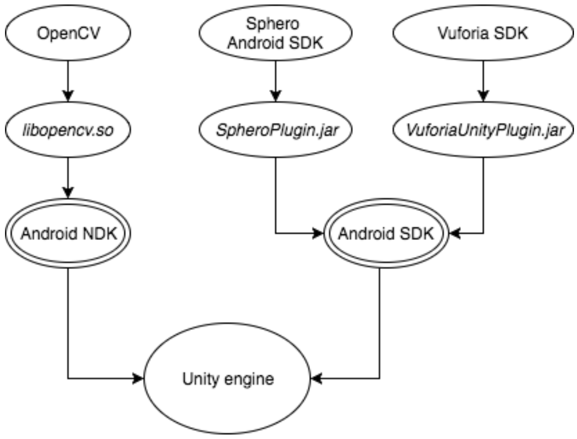


Figure 2. Dependencies diagram

Sphero SDK. The plugin supports simple instructions like connecting to the robot, sending commands to move or to change colors and calibrating.

2. Vuforia [12] – a library used for AR support for Android, iOS or Unity. It is used directly as a plugin for Unity and provides scripts and prefabs (objects already built in the library that include the necessary properties and scripts) to recognize markers and show augmented objects on top of the target area. The markers supported by Vuforia are: image, frame, object (cylinder or custom object that is scanned and loaded into the app). For this application, we used a simple image target evaluated with 5 stars on Vuforia website. All the scenes that use augmented reality use a prefab called ARCamera and an ImageTarget.

3. OpenCV [13] – is a library for image processing. It is used to detect the Sphero object in the image captured by the camera. Because it is not available for Unity, we wrote a small program in C++ which is then compiled in a shared object *(.so)* and called using NDK directives. The shared object library has dependencies to the open source OpenCV library and contains the processing method necessary to detect the robot.

Application scenes

Unity uses scenes to distinct the "pages" of the game. In the current application, we have 5 scenes, one for each level, and four others which provide a complete user experience in the game:

- *MainMenu* – this scene contains only UI objects: three buttons that redirects the user to the desired action: *Play, Drive sphero, Help*
- *DriveSphero* – this scene is dedicated entirely to controlling Sphero. It does not contain any AR layer because it acts as a training room to get used to the commands for directions, speed or calibration process.
- *HelpScene* – shows all the information about the game, connection to Sphero or how to use the app.
- *LevelsMap* – this scene contains UI elements that allows the user to select one of the five levels avaialable. The screen also displays the rank (1, 2 or 3 stars) and level's availability (locked/unlocked).

DEVELOPMENT SOLUTIONS

Sphero Tracking Algorithm

Sphero is the main character of the game and we must always know its position in the 3D scene. To do that we tried two different approaches: using Vuforia library and using a custom algorithm to detect light sources.

Tracking Sphero with Vuforia

Vuforia offers support to track 3D objects by scanning all the faces of the object and uploading the scan into the developer portal. The scanning process finds feature points on the object which will be later recognized and tracked from the video feed.

The problem is that Vuforia can track objects that have many edges and contrasting areas, which is exactly the opposite of Sphero's shape and colors: no edges and only shiny surfaces. To overcome at least one of these obstacles, we decided to wrap the ball with a material that has a lot of features. A picture of the used fabric was first uploaded to Vuforia developer portal and it was evaluated with 5 stars rating. We can see the wrapped Sphero and the coverage from scanning in Figure 3. As it is observed in the picture, Vuforia found 1148 points of interest that will be used to track the ball.

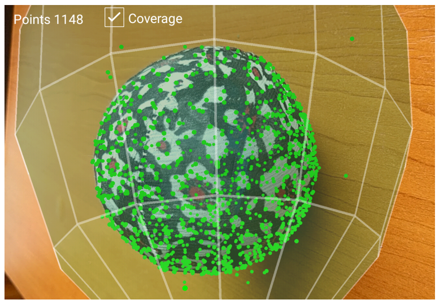


Figure 3. Wrapped Sphero and interest points

Although the scanning algorithm found many points of interest, the tracking results were not very satisfactory. The ball was detected only from small distances and if the room was well-lit. The tracking was quite unreliable if the ball was moving or the distance was more than 50 cm. Because the results from this approach were not good enough, we decided to use another method for tracking.

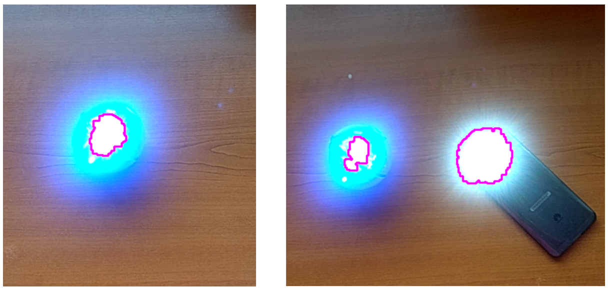
Tracking Sphero as a Light Source

Since Sphero's characteristics were not easily detectable with the previous method, we decided to use the light source and the spherical form for tracking Sphero. Using the image captured from the video feed, in each frame, the following algorithm is applied:

- transform image to grayscale
- binarize image using a threshold of 245

- noise reduction by applying the morphological operation "opening" with a 7x7 kernel
- find the contours from the remaining data in the image. The center of the contour is sent to Unity, where an algorithm is applied to transform the 2D point in a 3D object.

These operations are written in C++ and use the OpenCV library to process the image. The result of this algorithm is shown in Figure 4. The disadvantage of this method comes when more than one light sources can be identified in the image (see Figure 4b.)



a. Processed image with single light source **b. Processed image with multiple light sources**

Figure 4. Processed image with OpenCV

Transform 2D point to 3D object

Because we want to have Sphero as a virtual object in the scene, we need to translate the light source center point which is in screen coordinates (2D) into scene coordinates. The virtual scene is positioned based on a custom image marker detected by Vuforia. This marker and the robot should be on the same plane, meaning that we can determine Sphero's 3D position by intersecting the ray starting from the screen point with the plane determined by the marker as observed in Figure 5.

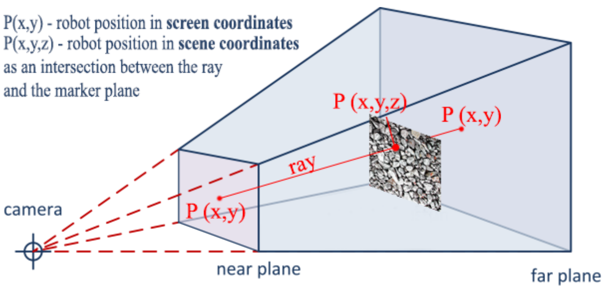


Figure 5. Screen position to 3D position

After knowing the 3D position of Sphero's center, all we have to do is create a 3D object with Sphero's dimensions and move it to the computed point in scene. The algorithm performs well on medium performance devices and the position is updated in each frame following the light source.

Sphero Auto-Calibration Algorithm

Being a spherical object, we do not know which way Sphero considers to be the "forward" direction. To be sure that Sphero is moving according to our joystick input, before using it, we should calibrate the robot. In other words, we should teach the robot which way is forward for us. This step consists in aligning the robot backlight in front of the user, by rotating the robot.

Although calibration is not difficult, children would prefer to just dive into the game, without any additional steps for the setup. Also, they can do mistakes in calibration which will result in sending wrong directions to Sphero. For these reasons, we decided to develop an auto-calibration feature. This consists in automatically aligning the robot to user forward direction, when the game is started.

The algorithm is represented in Figure 6 and has the following steps:

- choose a position S_{target} in front of Sphero considering camera position. The distance between Sphero and position S_{target} is d
- move Sphero with the direction "forward" (0 degrees) until d is the distance between Sphero's initial position S_{init} and Sphero's current position S_{curr}
- compute the angle between the vectors determined by $S_{init}S_{target}$ and $S_{init}S_{curr}$ and adjust Sphero's rotation with this angle.

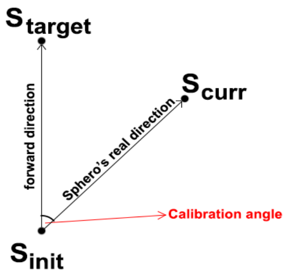


Figure 6. Calibration algorithm representation

GAME STORIES

As it we previously mentioned, the game is divided in five levels with progressive difficulty. Each level will teach the child one different concept appropriate to his age (numbers, geometrical figures, animals, etc). The user can play a level, only if he/she completed all the previous levels.

Common Elements in Levels

A level is represented as a Unity scene. Sphero is represented in all scenes by a translucent sphere and its position is updated according to the light source center. The collision detection is performed in every frame and is based on the virtual representation of Sphero and the virtual objects placed on the scene. A collision between virtual Sphero and another object in scene is followed by visual and audio indicators: Sphero will have green lights when the object is correct and red lights when a wrong object is selected and a proper sound will be played in both cases.

Star rating

To evaluate a level, we use the number of mistakes and the time spent to achieve the level's goal. Each level has different thresholds for these parameters, depending on the number of elements and the complexity of the scene. The performance on each level will be ranked with a rating from 1 to 3 stars. Three stars rating can be achieved only if the user did not make any mistakes like selecting a wrong object and if he/she completed the level in a reasonable time.

Level UI

Every level has one UI layer with the following elements:

- *joystick* – it controls the robot and accepts moves in any direction from 0 to 360 degrees
- *calibration button* – initiates the calibration of the robot
- *elapsed time* – shows the time spent in the current level
- *mistakes count* – shows the number of wrong selected objects
- *reset button* – resets level's progress, elapsed time and number of mistakes
- *help button* – replays instructions for the level
- *back button* – goes back to level selection scene
- *start level dialog* – shows a short description of the level and also plays an audio with the instructions
- *end level dialog* – shows the score of the level (1, 2 or 3 stars), the time and the number of mistakes.

Levels Description

Level 1 – Colors and 3D shapes

This level will help children to accumulate basic knowledge about colors and shapes. Although this kind of exercise is commonly used in kindergartens, this method is funnier and more interesting. The scene consists of 7 geometrical figures like cube, sphere, cylinder which are colored in 4 base colors (see Figure 7). The gameplay consists in selecting the correct objects as requested in a message at level start and after each correctly selected object. There are no restrictions regarding the path to the correct object, but Sphero should not collide with a different object because it will be considered a mistake.

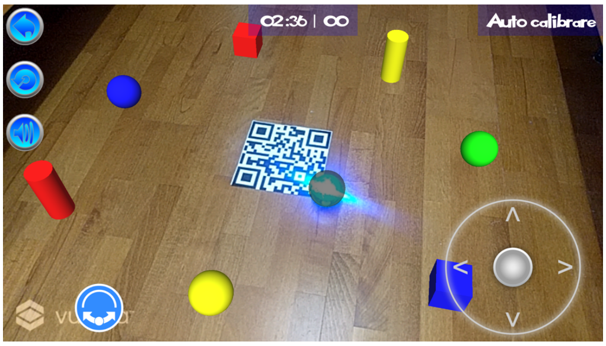


Figure 7. Level 1 - Colors and 3D shapes

Level 2 – Learning numbers

Since numbers are one of the most important concepts that should be learned by children, the second level proposes a fun way to learn counting. The scene shows the numbers from 1 to 9 in a random order (see Figure 8) and the child must select all numbers in ascending order. If mistakes are made, these will affect the final score, but the user can continue selecting another number without restarting the level.

Figure 8. Level 2 - Learning numbers

Level 3 – Wild Animals

One important part in child's development at a preschool age is to learn the animals and their habitat. Because some of these animals could be dangerous, this application proposes another way than seeing them in real life. 3D projections of animals will appear in the scene (see Figure 9) and the child can interact with them by tapping the screen. For now, the interaction is basic: play the sound of the animal and execute an animation, but more complex interactions will be added.

To pass this level, user has to select each animal in the correct order. The order will be displayed step by step: first message is shown at level start and the following are displayed after each correct selection.

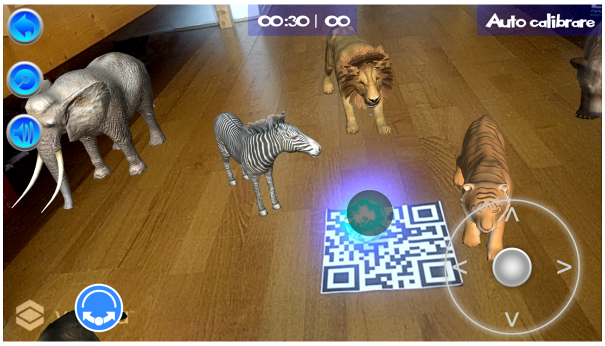


Figure 9. Level 3 - Wild animals

Level 4 – Drawing dot-by-dot

Drawing is another common activity in kindergarten. For this application, we included the dot-by-dot drawing technique which is both fun and educational. Dot-by-dot drawing implies showing a pattern and the control points of that drawing. The child should connect the points in the correct order to obtain the final figure. For this application, we included 2 simple patterns for now: house and star. The control points are shown as letters (see Figure 10). Therefore, while playing this level, the child will learn the first letters of the alphabet and to draw simple figures.

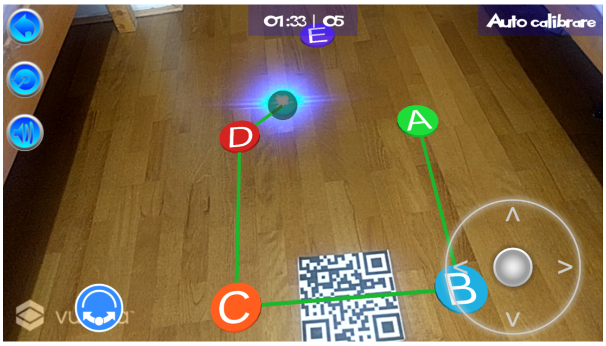


Figure 10. Level 4 - Drawing dot-by-dot

Level 5 – Collect Chests

This level is a little different than the others. It does not teach the child any new concepts but rather it tests his abilities to control and to think ahead his moves.

The scene is composed of 3 moving obstacles and 3 chests (see Figure 11). To complete the level, the child must collect all the chests but without touching the obstacles. The first two obstacles are boxes that are moving with different speed on Ox axis. To collect the first two chests, user should observe the course and the speed of the boxes and carefully control Sphero to avoid those obstacles. The third obstacle is a moving rock that is rotating and protecting the final chest. The child must collect the final chest when the rock passed, but he should be careful not to get hit when it rotates back.

When Sphero is hit by one of the obstacles, the level is reset and Sphero is sent back to the starting point. No mistakes are permitted to reach level's goal.

Figure 11. Level 5 - Collect chests

CONCLUSIONS

As a first conclusion, the proposed application can be considered an innovation for supporting preschoolers teaching process. Learn-o-Bot can be used both at home or in kindergartens and it does not require a supervisor to explain the task to the child. The messages are clear and always appear in two forms: written and audio, making the communication process easier.

The actual learning process happens when the kid controls the robot into achieving level's goal. If the child completes a level, it is considered that he accumulated the necessary information from that level and can pass to the next one. The star rating given at level end acts like a reward and a grade for finishing the level. To not disappoint kids, at least one star is given just for finishing the level. This rating will stimulate him/her to try again and receive the maximum number of stars. This means that a small rating will most likely make the child replay the level and get better rating, while a three-star rating means that the concepts are learned and understood correctly, without mistakes and in a good response time.

In current state, the application offers support only for a small area of knowledge that is necessary for children. To become a better educational application, the first improvement will be to add more levels in order to cover as much information as it is possible. Also, for now, the speed of the robot is fixed and cannot be changed from within the app. As an improvement idea, a good feature would be to allow the users to control the speed (for example in games with car driving, a slide from bottom to top means accelerating). And of course, an important step would be to improve Sphero's tracking algorithm.

REFERENCES

1. Rosie Flewitt, David Messer, Natalia Kucircova, *"New directions for early literacy: The iPad",* Journal of Early Childhood Literacy
2. Sphero SPRK+: *http://www.sphero.com/sprk-plus*
3. Math alive website: *http://alivestudiosco.com/math-alive-kit/*
4. Sphero's components image source: https://www.maximisetechnology.com.au/product/sphero-sprk-plus/, available on 25 May 2017
5. Lighting Lab, *http://sprk.sphero.com/*
6. J. M. Rodriguez Corral, A Morgado-Estevez, D. Molina Cabrena, F. Perez-Pena, C. A. Amaya Rodriguez and A. Civit Balcells, *Application of Robot Programming to the Teaching of Object-Oriented Computer Languages,* International Journal of Engineering Education, 2016
7. The Rolling Dead: *http://blog.sphero.com/blog/spheros-the-rolling-dead-now-available-on-ios-and-android/*
8. M. Panzariu and A. Iftene, *Sphero - Multiplayer Augmented Game (SMAUG),* RoCHI 2016
9. Quiver website: *http://www.quivervision.com/*
10. Zoo-AR website: *http://zoo-ar.com*
11. Sphero Android SDK source: *https://github.com/orbotix/Sphero-Android-SDK*
12. Vuforia, *http:// vuforia.com/*
13. OpenCV documentation, *http://docs.opencv.org/*