CEUR-WS.org/Vol-2193/paper12.pdf

Reflections on Teaching App Inventor: Challenges and opportunities

Paolo Musmarra

Department of Computer Science. University of Salerno
pmusmarra@gmail.com

Abstract. App Inventor is a visual programming language developed by Google in 2010 and currently hosted and maintained by the MIT Center for Mobile Learning. It has been successfully used to teach introductory computer because is an intuitive, visual programming environment that allows everyone – even children – to build fully functional apps for smartphones and tablets science concepts (CS0) and introduction to programming (CS1) skills for students in CS and Non-CS majors. And what's more, App Inventor is blocks-based tool facilitates the creation of complex, high-impact apps in significantly less time than traditional programming environments. The rapidly advancing mobile computing technologies along with abundant mobile software applications ("mobile apps" hereafter) make ubiquitous mobile learning possible. This study helped reveal the educational value of mobile app design activities and the challenges and opportunity of teaching/learning mobile app design. The findings can also encourage teachers to explore and experiment on the potential of incorporating these design learning activities in their respective settings, and to build mobile apps for their diverse needs in teaching. This paper presents some Challenges and Opportunities observed while teaching courses on Mobile Application Development with App Inventor. In particular this paper shows the following topics that instructors should take into consideration when designing their courses with App Inventor: designing User Interfaces, discussing Data communications, Planning and Designing Apps, Visual vs. Textual Programming to name a few.

Keywords: App Inventor, mobile learning, mobile apps

1. INTRODUCTION

App Inventor is a visual programming language developed by Google in 2010 and currently maintained by the MIT Center for Mobile Learning. App Inventor has been successfully used to teach programming concepts to beginners in both secondary and higher education courses[1][2][3]. MIT App Inventor is an intuitive, visual programming environment that allows everyone to build fully functional apps for smartphones and tablets. Furthermore, our blocks-based tool facilitates the creation of complex, high-impact apps in significantly less time than traditional programming environments. App Inventor can be utilized to teach computing and other computing concepts for student that already have some programming skills. In fact, Gestwicki and Ahmad[4] suggest that Studio-Based Learning approach and App Inventor can be used not only to "introduce non CS-majors to concept of Computer Science-not just programming, but also ideas tend not to be covered in conventional CS1 courses such as human-computer interaction, incremental and iterative design processes, collaboration, evaluation and quality assurance" (p. 55). Challenges and opportunities that teachers should be aware of when preparing a course of mobile applications development together an introductory programming course as prerequisite are presented by Soares [5]. Moreover, Pocket Code was presented as a useful tools to build app on your own smartphones or tablets [12]. Pocket Code is a Scratch-like programming environment for Mobile Devices that allows you to build your own apps, especially own games, animations, interactive music videos, and many kind of other apps, directly on your phone or tablet. Blocks-based programming has become the lingua franca for introductory coding and programming. Studies have found that experience with blocks-based programming can help beginners learn more traditional text-based languages. Bau et al.,[6] have been explored how blocks environments improve learnability for beginners by 1) favoring recognition over recall, 2) reducing cognitive load, and 3) preventing errors. We have observed that a greater participation in computer science has led to an escalation of interest in block-based programming. A lot of students receive their first approach to programming in courses and activities like *Code.org's Hour of Code*. Simple blocks allow beginners to compose programs without struggling with the frustrations of syntax.

2. APP INVENTOR DEMOCRATIZES THE CREATION OF MOBILE APPS

A project/app made with App Inventor consists of a set of graphic components and a set of program blocks that provide functionality to these components. Components include items visible on the phone screen (e.g. buttons, layout, images, label, text boxes) as well as nonvisible components (e.g. TinyDB, Notifie r, Sensors). App Inventor has reached a broad international audience for use both in and out of classrooms and formal learning environments. As of May 2016, 4.7 million people from 195 countries have created over 14.9 million apps [7]. App Inventor is taught to a broad audience, ranging from grade school to college students. Industry professionals also use App Inventor, often as end-user developers who write programs to support their primary work or hobbies [8]. Another aspect that we highlighted is that App Inventor is often used outside of formal or academics learning environments. Decreases in usage during school holidays also support the fact that a significant portion of people use App Inventor outside of formal learning environments. During the course the primary learning resources are step by step tutorials. Each tutorial typically presents on either introducing a new component, visible or not visible (such as a Tiny DB or List or Canvas) or additional features for a previously introduced component. The App Inventor resources page on the App Inventor website contains 26 tutorials ranging from beginner level to advanced difficulty [7]. For these reasons many of App Inventor's users can be categorized as *end-user programmers*. An important definitions of end-user programming is "programming to achieve the result of a program primarily for personal, rather than public use" (Ko 2011,[8]).

2.1 THE COURSE

The course, Mobile Development for Android, was offered during spring semester in 2016 and 2017, and had an introductory programming course as a prerequisite. A total of 124 students enrolled in the course completed the survey, 72 in spring 2016 and 52 in spring 2017. Students were asked which specific courses should be prerequisites for the Mobile Development Android course. A database course was also declared by about 48 percent of students as a recommended prerequisite.

2.2 MOBILE APP DESIGN ACTIVITY

In the first part of course, each students developed some simple apps following following the tutorials in the textbook authored by Wolber et al.,[9] and incorporated treatment of two main components of App Inventor called *Designer* to project User Interfac*e* and *Blocks* to design logical behavior of your own app. The students then developed their own app design proposals individually or in groups, using the templates provided by the teacher. The proposal included three main sections:

- Component map, where the students described the Designer components and how users would interact with their apps;
- A planning table that involved user analysis and future upgrade with this app in a time scale;
- App behavior flowchart, where the students presented the logic of app behavior in terms of blocks and logical programming

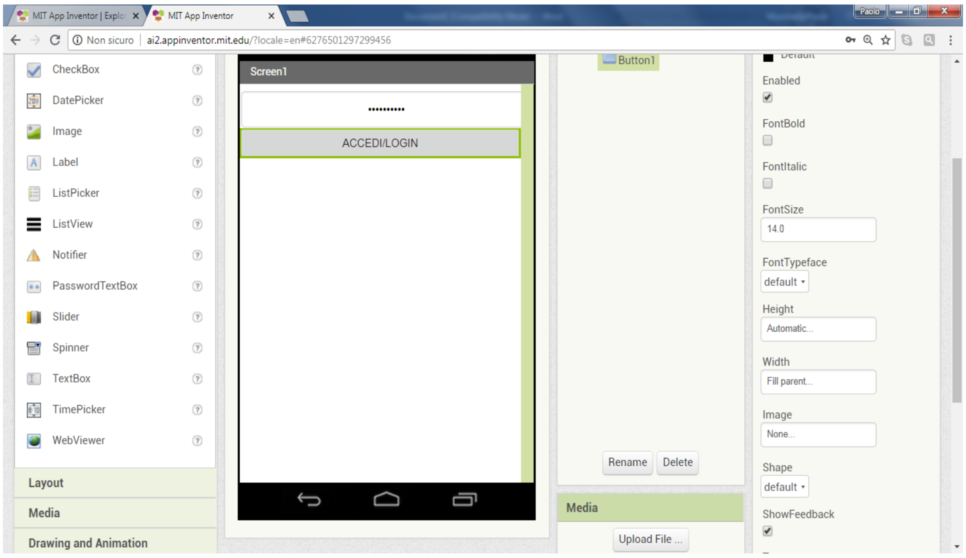
3. DESIGNING THE COMPONENTS

App Inventor has been successfully used to teach introductory programming concepts to beginners in both secondary and higher education courses [10]. We summarized the subject and our proposal to the students use the principles of security in critical and aware way using the method of learning by doing; the steps of the project and the features of this app will be described.

We show the main steps to build an app with a subject about “security” proposed in the second part of the course. We create a first screen; it has a password TextBox in which we write a password and a button to verify password, as shown in Figure 1.

Now, the main steps to create own mobile app with App Inventor are described:

- First you choose what you want your app to do, chose the needed components , and set the properties for each component using Component Designer.
- Second, try to work with the Blocks Editor, using blocks of codes as a visual code and connected like puzzle pieces, to make the components work just drag-and-drop it into the Blocks Editor.
- Third, test the app with smartphone or or an emulator. We prefer to use the cell phones, because it’s more fun and it’s easier to test phone-specific functionality.



Enabled
☑

FontBold
□

FontItalic
□

FontSize
14.0

FontTypeface
default

Height
Automatic...

Width
Fill parent...

Image
None...

Shape
default

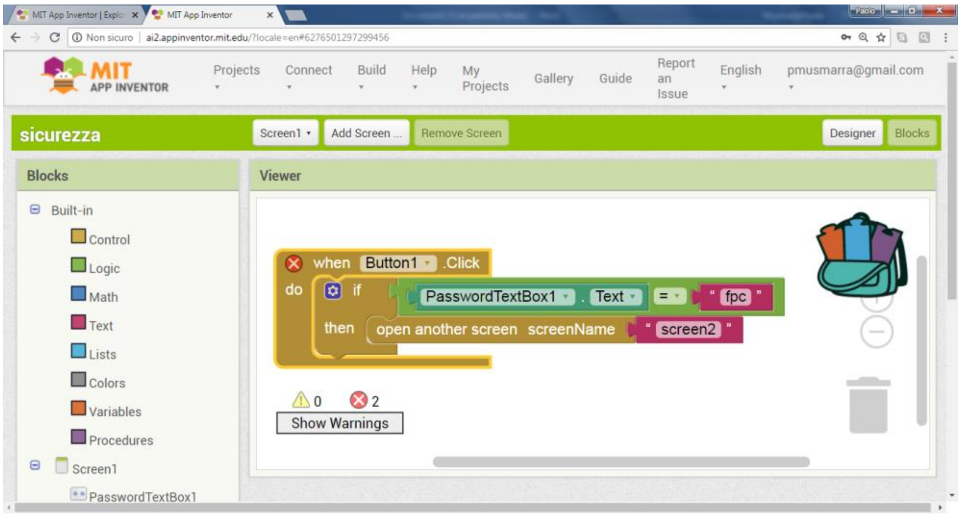
ShowFeedback
☑

Fig.1 Screen 1

Adding Behaviors to the Components. Screen 1

We will start the programming the behavior when the button LOGIN is clicked and Password TextBox is triggered. Only if the password is equal to the value saved before, open another screen. Finally, it is possible add and connect blocks as shown in Figure 2

.



```
when Button1 .Click
do  if PasswordTextBox1 . Text = " fpc "
    then open another screen screenName " screen2 "
```

Fig. 2. Check password

3.1 VARIATIONS

Even if the students close the App, turn off the cell phone, they can retrieve the stored data on the cell phone. For this reason, this app is more secure than the apps we can find in the play store because the data are on the smartphone or tablet. We can modify the login screen with two or more passwords to enter in the App; we can create many security systems because we know them. There is not more secure system than that we can create.

4. CONCLUSIONS

In this paper we have presented a summary of security and the actual issues in High School's context. This paper discussed challenges and opportunities that teachers should take into consideration when designing their courses with App Inventor: In particular we introduce Pre-Requisite for the course and the need of planning and designing app.

App Inventor has the potential to be included in the curriculum of other courses where students could take a basic course on App Mobile Development early in their curriculum. For instance, in a Software Engineering course, students could use the tool to help with requirements and utilize Designer to realize a mockup.

A high level framework for the teaching aspects of security has been proposed, together with elements of mobile programming with App Inventor, a visual programming tools that enables students to see and test your own app. In our experience, we want build an app to make secure and protect personal information. We will continue to discuss with teachers and obtain opportunities to integrate important security topics into other non-security courses. We hope our experiences can be helpful for other teachers to improve the security integration approach in other courses.

5. REFERENCES

1. Abelson, H. (2012). *From computational thinking to computational values*. Paper presented at the Proceedings of the 43rd ACM technical symposium on Computer Science Education, Raleigh, North Carolina, USA..
2. Haungs, M., Clark, C., Clements, J., & Janzen, D. (2012). *Improving first-year success and retention through interest-based CS0 courses*. Paper presented at the Proceedings of the 43rd ACM technical symposium on Computer Science Education, Raleigh, North Carolina, USA.
3. Robertson, J. (2014). Rethinking how to teach programming to newcomers. *Communication of the ACM, 57*(5), 18-19. doi: 10.1145/2591203
4. Gestwicki, P., & Ahmad, K. (2011). App inventor for Android with studio-based learning. *Journal of Computer Science in Colleges, 27*(1), 55-63:.
5. Soares, A. (2014). Reflections on Teaching App Inventor for Non-Beginner Programmers: Issues, Challenges and Opportunities. *Information Systems Education Journal, 12*(4), 56-65. (A preliminary version appears in The Proceedings of ISECON 2013).
6. Bau, D., Gray J., Kelleher C. Sheldon J., Turbak F. Learnable programming: Blocks and Beyond. (2017)
7. MIT App Inventor "http://appinventor.mit.edu/explore/". Accessed June 18, 2017.
8. Andrew J. Ko, Robin Abraham, Laura Beckwith, Alan Blackwell, Margaret Burnett, Martin Erwig, Chris Scaffidi, Joseph Lawrance, Henry Lieberman, Brad Myers, Mary Beth Rosson, Gregg Rothermel, Mary Shaw, and Susan Wiedenbeck. The state of the art in end-user software engineering. ACM Comput. Surv., 2011
9. Wolber, D., Abelson, H., Spurtus, E., & Looney, L. (2011). *App Inventor: Create your own Android apps.* Sabastopol, CA: O'Reilly Media.
10. Andrey Soares and Nancy L Martin. 2014. Teaching Non-Beginner Programmers with App Inventor: Survey Results and Implications. In Proceedings of the Information Systems Educators Conference ISSN, Vol. 2167. Citeseer, 1435
11. Musmarra, P. The safety on your own app with App Inventor, CISIS 2017, Torino
12. W. Slany Proceedings of the companion publications of the 2014…–2014 dl.acm.org