UDOO App Inventor: Introducing Novices to the Internet of Things

Antonio Rizzo, University of Siena, Siena, Italy
Francesco Montefoschi, University of Siena, Siena, Italy
Sara Ermini, University of Siena, Siena, Italy
Giovanni Burresi, University of Florence, Florence, Italy

ABSTRACT

This paper describes the opportunities provided by the new graphical tool UDOO App Inventor (UAPPI) for enhancing the programming learning experience. With this tool, coding and programming are no longer limited to screen pixels but will incorporate real objects in the physical world. The aim of the authors’ research is to develop new tools for coding alphabetization, by focusing on live programming, event programming, physical computing and overcoming syntax obstacles by using blocks programming. They describe two simple Research through Design case studies carried out with different categories of attendees in order to illustrate the potential of the UAPPI platform. In the first case, the authors render a door interactive, while in the second, they build a simple rover.

Keywords: App Inventor, Designer, Event Programming, Internet of Things, K-12, Live Programming, Makers, Physical Computing

INTRODUCTION

The most common visual programming environments for youths and newcomers to the computing world (for example, Scratch, Alice, Greenfoot, AgentSheets) provide solid platforms for creative activities such as designing games, animating interactive stories and running simulations. However, much of the activity is confined to a computer screen and limited to the programming of pixels behavior. A new programming environment, App Inventor, provides an important step toward programming real-world tools.

App Inventor was designed to make the development of mobile applications easy and rewarding (Wolber et al, 2015). Today’s mobile devices are better equipped to interact with the physical world than a computer thanks to, for example, GPS, magnetometers, gyroscopes, accelerometers,

DOI: 10.4018/IJPOP.2015010103
near-field communications and so forth. App Inventor offers a specialized environment for programming real-world tools, which in turn makes programming more attractive to novices.

In what follows, we present an extension of the App Inventor 2 (AI2) Integrated Development Environment (IDE): the UDOO App Inventor (UAPPI for short; see http://appinventor.udoo.org for a beta version hosted by University of Siena with temporary access). This extension transforms App Inventor into an environment for programming not only smartphones and tablets but also physical objects, by integrating microcontroller-managed sensors and actuators onto a UDOO board running Android (http://www.udoo.org; Palazzetti, 2015). UDOO App Inventor aims to gently introduce beginners to the intriguing world of the Internet of Things, a fast-growing frontier of computing that is transforming everyday objects into human-centered interactive systems (Kortuem et al, 2010).

Below, we briefly introduce App Inventor and two of its key features, live programming and event-driven programming. We then explain how UDOO for App Inventor works and the range of incentives that it offers novices of any age, acquainting them with coding by addressing real world issues with innovative solutions right from the start.

**APP INVENTOR**

App Inventor for Android is a visual programming platform for creating mobile applications (apps) for Android-based smartphones and tablets. It was developed at Google Labs by a team led by Hal Abelson on sabbatical from MIT (Abelson, 2009). Today, App Inventor runs as a web application administered by staff at MIT’s Center for Mobile Learning - a collaboration of MIT’s Computer Science and Artificial Intelligence Laboratory and the MIT Media Lab. The App Inventor source code is available under an open-source license, allowing anyone to deploy App Inventor servers. In 2014, MIT hosted the web IDE, servicing 87,000 users per week, with a total of over 2.2 million registered users who developed 5.5 million apps (Schiller et al, 2014).

App Inventor is similar to Scratch (http://scratch.mit.edu) and Alice (http://www.alice.org). Developing apps in App Inventor or these other languages does not require writing textual code. The look and behavior of the app is developed visually, using a series of building blocks for each intended component. Like its competitors, App Inventor aims to make programming enjoyable and accessible to novices.

Of all these programming platforms, however, only App Inventor allows for the creation of apps for smartphones and tablets. Given the increasing popularity and ubiquity of these and other mobile devices, App Inventor has enormous potential for attracting newcomers of any age to computing, coding and computational thinking. In a very short time (for example, a few days), beginners can build apps that are not only fun, but have real-world utility. “App Inventor allows creative people to transform their ideas into working, interactive apps that can be taken up by large companies, used by non-profit organizations and turned into startups” (Wolber, 2011).

App Inventor lets users create apps with real life impact, yet its learning curve is gentle. The visual nature of its language reduces the syntax problems common among programming beginners first starting to design an app. Even more important are two key features of the programming environment, namely live programming and event-driven programming. These features help beginners address the formidable challenges of developing a robust programming logic and specifying interactive behavior with a static, graphic language. They are instrumental to the UDOO App Inventor extension.
Related Content

From E-Learning Tools to Assistants by Learner Modelling and Adaptive Behavior
[www.igi-global.com/chapter/learning-tools-assistants-learner-modelling/24482?camid=4v1a](www.igi-global.com/chapter/learning-tools-assistants-learner-modelling/24482?camid=4v1a)

Diffusing Management Information for Legal Compliance: The Role of the IS Organization Within the Sarbanes-Oxley Act
[www.igi-global.com/article/diffusing-management-information-legal-compliance/3839?camid=4v1a](www.igi-global.com/article/diffusing-management-information-legal-compliance/3839?camid=4v1a)
Understanding the Impact of Household End Users' Privacy and Risk Perceptions on Online Behavior
[www.igi-global.com/chapter/understanding-impact-household-end-users/18150?camid=4v1a](www.igi-global.com/chapter/understanding-impact-household-end-users/18150?camid=4v1a)

The Changing Environment of Software Copyright: The Case of Apple Computer v. Microsoft Corp.
[www.igi-global.com/article/changing-environment-software-copyright/55652?camid=4v1a](www.igi-global.com/article/changing-environment-software-copyright/55652?camid=4v1a)