EMPOWERING MIDDLE SCHOOL STUDENTS TO CREATE DATA-ENABLED SOCIAL APPS *

Lijun Ni, Farzeen Harunani, Fred Martin
University of Massachusetts Lowell, Lowell, MA 01854
lijun.ni.gt@gmail.com, farzeen@outlook.com, fredm@cs.uml.edu

ABSTRACT

MIT App Inventor has enabled middle school students to learn computing while creating their own apps-including apps that serve community needs. However, few resources exist for building apps that gather and share data. There is a need for new tools and instructional materials for students to build data-enabled, community-focused apps. We developed an extension for App Inventor, called AppVis, which allows app-makers to publish and retrieve data from our existing web-based collaborative data visualization platform. We used AppVis with supporting instructional materials in two one-week summer camps attended by a total of 33 middle school students. Based on student interview data and analysis of their final apps, we found that this approach was broadly accessible to a diverse population of students and motivated them to build apps that could be used by their own communities.

INTRODUCTION

In 2015, researchers from the University of Massachusetts Lowell partnered with two urban school districts to run summer camps using MIT App Inventor. Students built socially relevant apps to help local communities [9]. Over half of the students built games. Out of 38 total apps, only one was a utility app (meant to be used as a tool by others), and only about 60% of the students felt that they had built apps they would like to share with the community. In a second iteration, we introduced new technology and instructions focused on building data-enabled, community-focused apps. Students built more utility apps, and more were proud to share with their communities.

---

* Copyright © 2017 by the Consortium for Computing Sciences in Colleges. Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the CCSC copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Consortium for Computing Sciences in Colleges. To copy otherwise, or to republish, requires a fee and/or specific permission.

BACKGROUND

MIT App Inventor is a web-based platform for developing Android applications [8]. It uses a blocks-based programming language that provides a low barrier to entry, and it has become a widely used platform for introductory computer science. It is also often used in informal education [1, 4]. App Inventor works well in this context, because "'what is being built' is the driving force, not the abstract goal of learning to code... many more [students] get excited about solving real-world problems and improving their lives and those of their friends" [11]. Students enjoy working with hands-on projects that give them an opportunity to be creative within the structured program curriculum [6]. App Inventor lends itself to the "use-modify-create" model for engaging youth in computational thinking [5].

Most App Inventor users "tend not to create apps that are functionally different from completed tutorials" [12]. This lends itself to Blikstein's "Keychain Syndrome" [3]: "Educators should shy away from quick demonstration projects and push students towards more complex endeavors," else the students will continue to build variations of demo projects and not progress forward. This idea, compounded with Barba's advice [2]: "tutorials are best treated as a kind of scaffolding rather than a primary source and, when used, should not be scaffolded any further," provide a solid groundwork for where to start in designing materials for introducing students to App Inventor concepts.

CURRICULUM OUTLINE

In prior work, we ran 5-day App Inventor summer camps (Camp 2015) that we considered to be very successful [9]. When designing the second iteration of the camp (Camp 2016), the research team had two goals in mind: first, to have more students feel that they had built "real" apps that they wanted to share, and second, to minimize the number of students whose final apps were games.

In the prior year, not many students expressed a desire to share their work with the world, leading the research team to believe that the students did not think they had built "real" apps. However, the idea of a "real" app is highly relative to the student in question, and there exists no standard for what makes an app "real" in the student's mind. If we extend "real" apps to mean "real-world" apps, then this becomes an easier problem to solve by providing the students with a client, an incentive, and tools for creating an app that can solve a real-world problem.

In order to provide a client, we brought in community partners on the first day to discuss some issues within the students' cities, which allowed the students to reflect on some real-world issues that might affect their own communities.

To incentivize the students to build apps that they would consider to be "real," we offered to publish any finished students apps on the Google Play Store. Related to publishing, we add a lesson on Creative Commons and copyright.

Finally, to provide a tool that would allow the students to solve a real-world problem (on top of what already exists in App Inventor), we introduced the concept of data science via the iSENSE Project [7]. We used a custom version of App Inventor integrated with iSENSE and allowed students to upload, visualize, and collaborate on data. We refer to the iSENSE-augmented version of App Inventor as AppVis.

Table 1: Camp 2016 Schedule (Tutorials Highlighted)

<table>
<tr><th>Monday</th><th>Tuesday</th><th>Wednesday</th><th>Thursday</th><th>Friday</th></tr>
<tr><td>Introductions</td><td>Pair programming Intro</td><td>SURVEY TUTORIAL</td><td rowspan="6">Project work</td><td>Research surveys</td></tr>
<tr><td>Research surveys</td><td>Final app brainstorming</td><td>Project work</td><td rowspan="2">Finalize apps</td></tr>
<tr><td>How to use AI</td><td rowspan="2">GPS TUTORIAL</td><td rowspan="2">How to publish to Play Store</td></tr>
<tr><td rowspan="2">JUMP COUNT TUTORIAL</td><td rowspan="3">Present apps (“App Fair”)</td></tr>
<tr><td>Creative Commons & copyright</td><td>Project work</td></tr>
<tr><td>Community Partner visit</td><td>“Storyboard” apps</td><td>Class-wide demos</td></tr>
</table>

Our approach also required a different set of learning materials from what has been used in the past. Rather than using the small, standard App Inventor tutorials (about ten were used in last year's camp), we designed three larger tutorials that encompassed multiple concepts, rather than one concept per tutorial. These tutorials had guidelines, but allowed considerable student freedom and flexibility. By using these new tutorials, we highlighted the ideas we wanted the students to explore, which were then ultimately the concepts we expected them to build into their final apps. Table 1 shows a simplified overview of the second camp's schedule. The schedule varied between the two camp sessions, but the overall sequences were the same.

APPROACH: APPVIS

From the App Inventor standpoint, AppVis consists of two new components, the iSENSE Publisher and the iSENSE Viewer.

iSENSE Publisher

The iSENSE Publisher is a non-visible component (it does not show up on the screen of the Android application). It takes a project number and a “Contributor Key” (a project-specific password for data contribution) as parameters. As seen in Figure 1, a user needs to name the dataset, provide the names of the fields, and provide the data in order to upload.

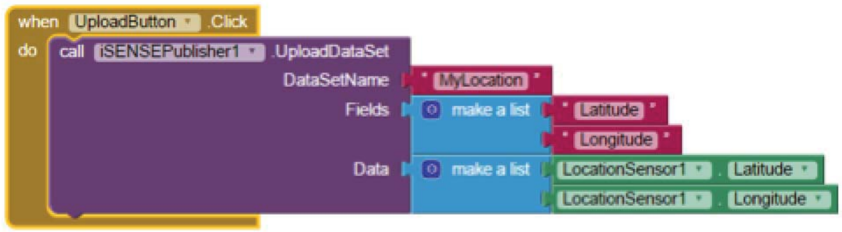

Figure 1: Example Usage of iSENSE Publisher

iSENSE Viewer

Though it is possible to use App Inventor's WebViewer component to view visualizations, having the students type in a URL and specify options is overly complex. So, the iSENSE Viewer component was built, which simply asks for a project ID and displays the corresponding default visualization for the project. As the component is streamlined to be as simple as possible, the only function associated with it is the ability to be refreshed, as seen in Figure 2.

Figure 2: Example Usage of iSENSE Viewer

PARTICIPANT DEMOGRAPHICS

A total of 33 students from two school districts attended the two Camp 2016 sessions. 19 were male (58%) and 14 were female (42%). Many of the students were first-generation; 19% had parents who did not speak English, and 71% had a language other than English (or along with English) at home. Given that student participation was elective, we were satisfied with the gender and ethnic diversity of the camps. Table 2 shows the ethnic composition of camp participants.

Table 2: Ethnic Composition of Districts & Camp 2016 Participants

Ethnics	*District 1*	*District 2*	*Campers*
African American	14.50%	18.00%	12.90%
Asian	8.70%	4.90%	16.13%
White/Caucasian	62.60%	30.80%	48.39%
Hispanic	9.80%	43.90%	16.13%
Native American	0.30%	0.50%	6.45%

Pacific Islander	0.00%	0.10%	0.00%
Other	4.10%	1.90%	0.00%

STUDENT APPS

We analyzed all the apps from Camp 2015 and Camp 2016, and categorized them by themes, intended usage, and complexity.

App Themes

Overall, student apps remained socially-oriented both last year and this year. Out of 38 apps from Camp 2015, three were not community-focused; out of 18 apps from Camp 2016, only one was not community-focused. The two students who created this outlier app made a complex game much outside the scope of introductory App Inventor. These two students had the most prior coding experience. Out of the seventeen community-focused apps from Camp 2016, eight addressed personal health, six addressed local park issues, two addressed local endangered wildlife, and one addressed renewable power.

When the community partners came in on the first day, they discussed some issues that they had that could be solved via technology. One partner mentioned she would go to parks and fill out a form counting the number of people engaging in specific park activities. She had an example sheet that she passed around to the class, and students were amazed by just how much data she collected by hand.

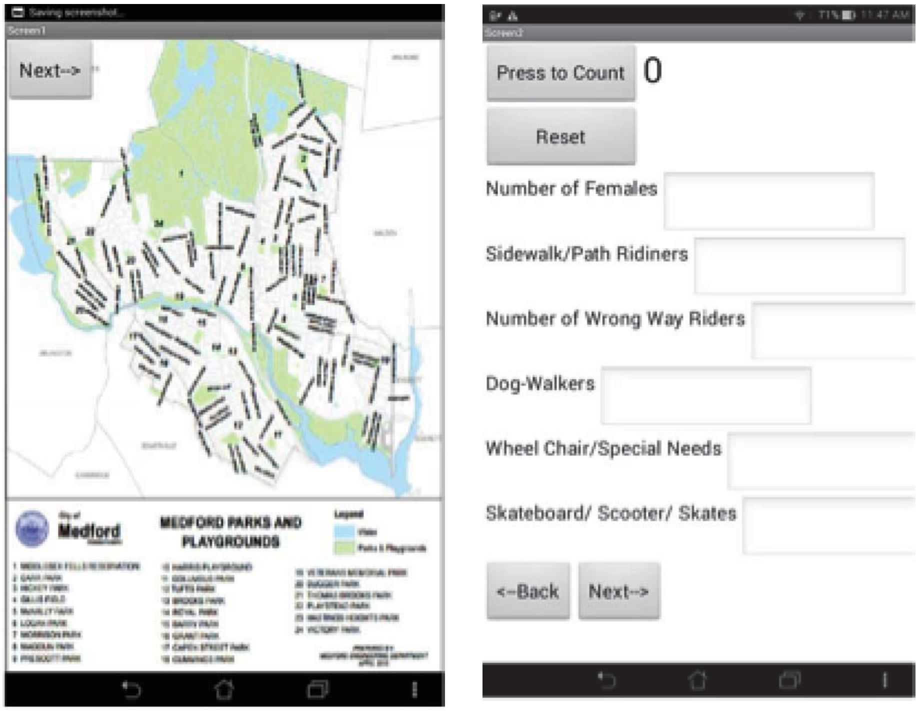

Figure 3: HeadCount App Screenshots

Two students decided to crowdsource and automate her job by making an app that had all the same fields as the sheet the partner passed around. Their app also uploaded all the data to iSENSE, so that anybody at any time could see the trends over time of how people use the local parks. Such a tool would not only alleviate the partner's job, as park attendees would now be able to update her on park usage, but also provide any other interested party with an easy, convenient way to visualize all this data. Figure 3 shows screenshots of this app.

App Classification

We classified student apps as Informational, Utility, or Game[1], where Camp 2015 apps were without AppVis and Camp 2016 apps are with AppVis (Figure 4).

[1]Informational: an app aimed at providing the user with information. Utility: an app designed to be used as a tool by the user. Game: an app aimed at entertaining the user.

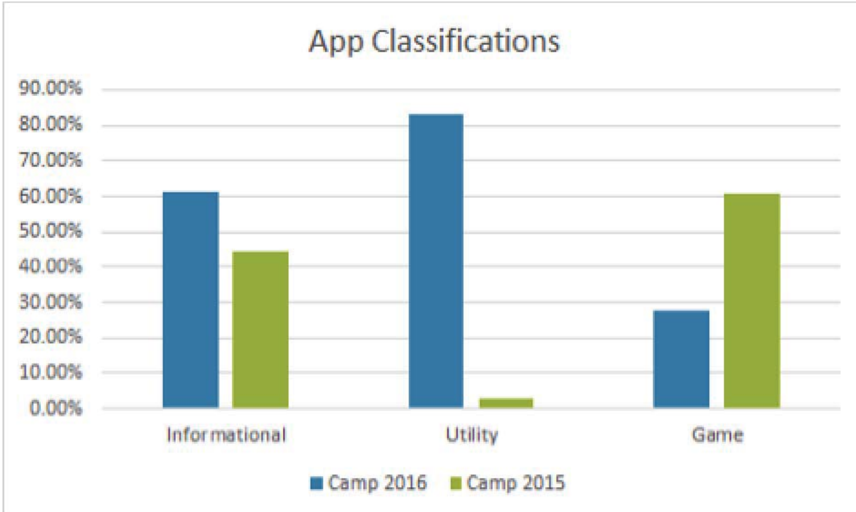

Figure 4: App Classifications: Less Games, More Utility

In the tutorials, we introduced the students to Utility components (the iSENSE components), rather than showing them games as we did last year. While a similar percentage of students chose to make informational apps, we can see that showing students games will lead them to make games, but showing them utility concepts will encourage them to make utility-based apps.

This supports the research discussed above that students will lean towards being interested in using technology to solve problems in their local communities. However, the correct tools need to be provided. Using game-based tutorials is not sufficient to enable the students to think about using App Inventor for making socially relevant utility apps, but maybe for socially themed games. For example, last year, several students made recycling themed games (e.g., tap on the items that are trash), whereas this year, several students made fitness apps that aimed to provide tips for a healthier lifestyle. These apps were informational but also utility-oriented, as several included daily calorie intake calculators based on a user's weight, gender, and height.

Camp 2016 students embraced the idea of multi-modal apps. For example, one app was separated into a kids section and an adults section. The kids section had games, but the adults section had recipes for healthy meals.

The main takeaway is that it very much matters how a tool is introduced to the students. If it is introduced as a tool for building games, then it will remain as such. However, if it is introduced as a tool for helping the community, then the students will treat it as such. With the two students who built a game, they had already had significant prior coding experience, so this was not an introduction for them. They already had an idea of what they wanted to do, and what they thought computer science is about, so our approach was least effective with them.

App Complexity

In order to measure the complexity of the students' final apps to see if the students had made comparable apps with and without AppVis, we turned to Sherman and Martin's mobile computational thinking assessment rubric [10]. We used the rubric to score all of the apps from Camp 2015 (without AppVis) and Camp 2016 (with AppVis). Figure 5 shows the app complexity scores based on the rubric.

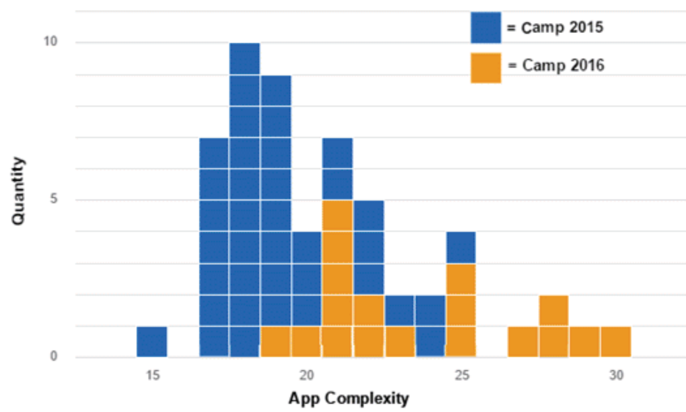

Figure 5: Distribution of App Complexity by Camp Year

Apps from Camp 2015 had an average score of 19.21, and apps from Camp 2016 scored 23.38. This was largely attributable to students' use of the AppVis components. The iSENSE Publisher component yielded four points in the “data sharing” category according to the rubric, and the iSENSE Viewer yielded two points in “public web services.” The average “data sharing” score without AppVis was 1.0, but the average score for the same category is 2.22 with AppVis. Similarly, the average “public web services” score without AppVis was 1.24, but the average score with AppVis is 2.11.

Students in Camp 2016, in general, incorporated data-sharing into their work, while retaining the level of sophistication demonstrated by students of Camp 2015. This result shows that the AppVis material was readily accessible to the campers.

CAMP IMPACT

We conducted semi-structured interviews with 25 students with consent on the end of Day 4 and Day 5, during the stage of finishing final projects. 14 of the 25 students were female. 9 out of the 25 students reported prior experiences with App Inventor including four alumni of our 2015 summer camp. Another five students learned App Inventor through a school-year class. Two students had some programming experience with Scratch.

We used the same interview protocol as last year's to understand participants' perspectives, experiences and programming practices through engaging them in conversations about their final projects and the design processes [9]. Two researchers coded the interviews using thematic analysis. Starting with a code list created from prior analysis of interviews with Camp 2015 participants [9], the researchers identified common ideas clustered into major themes on students' attitudes and experiences including addressing community needs, positive App Inventor experience, and interest in learning more about computer science. Meanwhile, new codes were created and new themes were identified in the analysis.

Findings

Similar to last year's findings, students were able to connect the app ideas with the community partners' visits; they enjoyed the camp and would like to learn more about creating apps and about computer science. Comparing with last year's results, we found that students in camp 2016 were more excited about sharing and publishing their apps. They saw App Inventor as a vehicle to help community and people. They enjoyed working with friends, partners, and teachers, and reported this aspect as one of the most fun part of the camp. We also found that students in camp 2016 reported more programming-specific challenges they encountered in creating apps while students in 2015 camp reported challenges more focused on technical issues and screens design. These findings were consistent across gender.

Creating Apps for Social-Good

Per this project's focus, students were guided to create their final apps addressing a specific community need. When asked where their app ideas came from, most students (except 3 out of 25) explicitly connected their ideas with the community partners' visit or identified the needs from their communities. The other three students linked their apps with their personal interests. Below is an example of one student explaining how her app was linked with community partner's visit.

> [MP6]: "[The community partner] came in and said it was her job to make parks better. She goes to parks and records how many people she sees biking, walking, etc., to see if it's getting good use. We made an app to make that easier."

App Inventor Experiences

Students reported favorable experiences with the summer camp. They saw App Inventor was "fun and cool" and felt proud of the products they created.

Figure 6: Most Fun Part of the Camp

Students enjoyed interacting with students and teachers, making their own apps, learning programming, problem-solving, experimenting with blocks, as well as helping the community. In addition to seeing making their own apps as one of the most fun part of the camp as last year's students reported, students in this year's camp enjoyed interacting with friends, partners and teachers in the camp. Seven students explicitly reported this point. For example, one student mentioned about enjoying teamwork with friends and teachers.

> [MP7]: "[The most fun part is] working as a group, trying to solve the bug of the app and the teamwork of my friends and the teachers. There were things I didn't understand and they helped me a lot. I know a lot more now about app inventor and apps."

Students also enjoyed creating real apps that could help the community. For example, one student perceived this as a most fun part:

> [MP2]: "The most fun part was making your own app, choosing something you want to do, helping the community by doing something they want to do but making things easier for them."

Sharing and Publishing Apps

When asked about their plans for sharing final apps, students were excited about publishing their apps to Google Play (an app store). This was a new activity introduced in this year's camp.

Figure 7: Plan to Share Final Apps

18 of the 25 students explicitly mentioned they were interested in sharing the app with the community or publishing to Google Play store to benefit more people. For example, one student expressed her willingness of showing the app to a community partner and other communities:

> [MP6]: "I want to show [the community partner name]. I want to show her because it's for her basically, but also for other parks and recreation things in other communities too."

The two students who were nervous about sharing the app were at stage of still working on completing the app at the time of interview.

Connecting App Inventor with Community

When asked how they would introduce App Inventor to their friends, all the students provided favorable comments about App Inventor, seeing it as a simple form of creating their own apps, being powerful and fun to use, and being useful in helping people. In particular, we saw students explicitly connect the idea of serving the community with App Inventor this year. Six students mentioned that App Inventor could help people or community when describing "what is App Inventor".

> [EP1]: “They should do App Inventor because it's good for you to learn about… if you want to make an app, you can make it and put it on the app store. This is to help the community be better.”

Interest in Learning More

Most students expressed their interest in learning more about computer science including creating apps except that one student said she would not take any further course without having a friend accompanied.

> [MP11]: “I don't want to be the only one there without a friend and I'll probably get confused a lot and not know what to do.”

The rest of the students expressed their interest in learning more. Most of them talked about future plans in terms of learning more about computers, creating apps, debugging, other programming languages, and taking computer science courses. Four students said “Yes”, but did not provide specific information.

Challenges and Difficulties

Students reported varied challenges they encountered in creating apps. Most of the difficulties were programming-specific, such as publishing data through iSENSE, creating mini database, or programming game score (with variable).

> [MP2]: “[The most difficult part is] figuring out what to put in the programming part to send everything to iSENSE. If you don't have the correct labels on iSENSE or this, the whole thing doesn't work.”

There were only four students mentioned non-programming related challenges (organizing pictures/screen layout, finding information, getting non-copyright pictures). In contrast, last year's participants reported the simpler technical challenges (screen design and WiFi connectivity) [9].

Girls vs. Boys

When we disaggregated the interview data by gender (14 females and 11 male students interviewed), no significant difference was identified related to the findings presented above. Overall, we saw similar points distributed across the two groups showing how they enjoyed the camp, felt proud of their work, and interested in learning more. We saw a few slightly different points. First, two boys reported they enjoyed most being able to help the community, while two girls reported “experiment with codes” as the most fun part. In terms of interest in CS, one girl expressed her need of friends' company in taking more CS courses. Although the same number of boys and girls expressed their interest in publishing the apps, four girls and one boy wanted to share their products with their friends and family.

REFLECTION & DISCUSSION

Per the Keychain Syndrome [3], where Blikstein's students were engaged and excited by fabricating keychains, our Camp 2015 students were engaged and excited by building simple games. These games were enough to impress the students' friends and families, so students never thought to use App Inventor for building apps that were not games.

We consider Camp 2016 to have been successful, not just in a pedagogical sense, but also in a research sense. This year, during our App Fair, all the students presented their apps, not only to their family and friends, but also to the community partners who had come on the first day to talk about community issues. One community partner this year said, “I wish we could merge all these apps together. That would be the perfect app for our project.” Another community partner asked two students if they minded putting the community partner's logo on their app so that it could be identified as being the official app of the partner's initiative. We had no comparable responses last year, before introducing AppVis.

Overall, Camp 2016 had more advanced content available to the students who were interested in going above and beyond, as we wanted to challenge alums returning from last year's camp. We also had a higher instructor-student ratio to better assist all students and alleviate logistical issues. The students had an overwhelmingly positive reaction to the camp-when asked what could be improved, many could not think of any suggestions. The community partners (the end “customers” of the apps) had only positive reactions to the students' work.

CONCLUSIONS

This paper has introduced a new tool, AppVis, for engaging students in computer science learning through a data-rich App Inventor environment, with the specific goal of encouraging them to build apps for social good. After using it in a summer camp with middle schoolers and seeing the positive results in both their work as well as their excitement, we could see that it successfully engaged students across minority and majority groups. The students felt as if they had done something constructive that benefited their own communities, and the community partners agreed. We feel confident that AppVis is a tool that can be used more widely to engage students in building apps that gather and share data.

ACKNOWLEDGMENTS

The authors would like to thank our project team members (Akira Kamiya, Molly Laden, Diane Schilder, and Mark Sherman) and summer camp instructors and TAs (Lori Blank, Mark Cheffro, Damian DeMarco, Ashley Hale, Jessica Hamerly, and Amy Lieberman). This material is based upon work supported by the National Science Foundation under Grant Number Grant No. 1433592.

REFERENCES

[1] A-Ghamdi, S., Al-Rajhi, N., Al-Onaizy, N., & Al-Khalifa, H., Using App Inventor 2 in a summer programming workshop: Improvements over previous years. IEEE Global Engineering Education Conference, 383-388, 2016.

[2] Barba, E., & Chancellor, S., Tangible media approaches to introductory computer science. ITICSE '15, 207-212, 2015.

[3] Blikstein, P., Digital fabrication and 'making' in education: The democratization of invention, FabLabs: Of Machines, Makers and Inventors, 1-21, 2013.

[4] Grover, S., & Pea, R., Using a discourse-intensive pedagogy and Android's App Inventor for introducing computational concepts to middle schools, SIGCSE '13, 723-728, 2013.

[5] Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J., Werner, L., Computational thinking for youth in practice, ACM Inroads, 2(1), 32-37, 2011.

[6] Martin, F., Scribner-MacLean, M., Christy, S., Rudnicki, I., Londhe, R., Manning, C., & Goodman, I., Reflections on iCODE: Using web technology and hands-on projects to engage urban youth in computer science and engineering, Auton Robot, 265-280, 2011.

[7] Michalka, S., Dalphond, J., & Martin, F., Inquiry learning with data and visualization in the STEM classroom, Society for Information Technology & Teacher Education International Conference, 2204-2211, 2016.

[8] MIT App Inventor | Explore MIT App Inventor, www.appinventor.mit.edu/explore, retrieved August 23, 2016.

[9] Ni, L., Schilder, D., Sherman, M., Martin, F., Computing with a community focus: Outcomes from an App Inventor summer camp for middle school students. Journal of Computing Sciences in Colleges, 31(6), 82-89, 2016.

[10] Sherman, M., & Martin, F., The assessment of mobile computational thinking, Consortium for Computing Sciences in Colleges, 30(6), 53-59, 2015.

[11] Wolber, D., Abelson, H., & Friedman, M., Democratizing computing with App Inventor, GetMobile, 53-58, 2014.

[12] Xie, B., Shabir, I., & Abelson, H., Measuring the usability and capability of App Inventor to create mobile applications, PROMOTO 2015, 1-8, 2015.