Apps for Social Justice: Motivating Computer Science Learning with Design and Real-World Problem Solving

Sarah Van Wart
UC Berkeley School of Information
Berkeley, CA 94720
vanwars@ischool.berkeley.edu

Sepehr Vakil
UC Berkeley School of Education
Berkeley, CA 94720
sepehr@berkeley.edu

Tapan S. Parikh
UC Berkeley School of Information
Berkeley, CA 94720
parikh@ischool.berkeley.edu

ABSTRACT
In this paper, we describe a twelve-week *Apps for Social Justice* course that we taught at an after-school program. Students read social justice literature, identified local community needs, and went through a design process to create fully functional mobile applications to address these needs. Using Nasir and Hand's concept of practice-linked identities [13], we argue that an integrative approach to introducing computer science – where CS principles are used in pursuit of meaningful community goals – provides multiple opportunities for students to participate in software development while connecting these skills and dispositions to their own experiences and to larger social issues. Unlike a concepts-first approach, which introduces computer science ideas using small, often decontextualized examples, a practiced-based approach that builds on student experiences may foster a more motivating and meaningful learning environment.

Categories and Subject Descriptors
K.3.2 [**Computers and Education**]: Computer and Information Science Education–computer science education.

General Terms
Design, Human Factors

Keywords
App development, apprenticeship, social justice, HCI, computer education pipeline

1. INTRODUCTION
Computer Science (CS) careers offer many material and intellectual rewards, and yet women and low-income people of color are grossly under-represented in the field. This disparity can be seen starting in K-12. For instance, in 2011, only 19.2 percent of the Advanced Placement Computer Science (AP CS) test takers were female – "the lowest ratio of female-to-male test-taking rates of any of the offered Advanced Placement tests" [6]. Even more troubling, Jane Margolis reports that in California – home to Silicon Valley and several of the nation's top computer science programs – although "underrepresented students of color make up a combined 49 percent of the high school student population, they account for only 9 percent of the AP computer science test takers (California Department of Education 2005; College Entrance Examination Board 2005)" [9]. Without a concerted effort to make computer science education more equitable, the field will remain stratified across gender and racial dimensions.

Computer Science has many of the same qualities as mathematics: both tend to attribute success in the domain to one's innate intelligence [7]; both serve a gatekeeping role in granting access to high-opportunity careers; both are critical tools in Science, Technology, Engineering, and Math (STEM) professions; both claim to teach abstract thinking and in particular mathematical/computational habits of mind [14][16]; and both have wide achievement gaps among groups [8][9]. Given these parallels, this paper draws from several equity-minded ideas in the mathematics education literature to explore how learning environments may be organized to provide more equitable opportunities for all students.

1.1 Equity in Mathematics Education
Danny Martin, a mathematics education scholar, asserts that "eliminating inequities in access, achievement, and persistence in mathematics is not an issue that can be separated from the larger contexts in which schools exist and in which students live" [10]. Many mathematics education scholars agree with this notion: that mathematics achievement disparities are embedded in larger, structural inequities. However, how these larger contexts might be addressed from within an educational setting is up for debate. Jo Boaler offers the notion of relational equity, which emphasizes the social opportunities provided to students to develop *equitable relations* with one another based on mutual respect, trust, and accountability [2]. She advocates for a classroom structure that focuses on *complex instruction*, a teaching method that aims to disrupt broader racial hierarchies from the inside out through the promotion of equitable classroom relations. A second perspective, articulated by González, Andrade, Civil, & Moll [12], questions the curricular choices that privilege certain types of knowledge over others in the standard mathematics curriculum. These scholars argue that by drawing from students' cultural Funds of Knowledge, a broader range of mathematical competencies can be valued in the classroom, which will in turn make mathematics more relevant and meaningful to minority students. A third perspective taken by Rico Gutstein [5] argues that mathematics should be used as a tool to directly critique the larger social and economic structures. This version of a social justice pedagogy applies statistics to real data sets to shed light on structural inequities in the "real world." Finally, Nasir and Hand [13], like Boaler, look to the ways in which learning environments are

ITICSE '14, June 21–25, 2014, Uppsala, Sweden

ACM 978-1-4503-2833-3/14/06…$15.00
http://dx.doi.org/10.1145/2591708.2591751

structured to promote student engagement. Their study analyzes how opportunities for engagement differ across two settings – a high school basketball team and a mathematics classroom – focusing on three distinct aspects of these settings: 1) access to the domain, 2) opportunities to take on integral roles, and 3) opportunities for self-expression in the practice. Building on previous scholarship examining the importance of engagement for academic achievement, they argue that students develop practice-linked identities – or feelings of connection to an activity – through their experiences participating in the activity, and through the kinds of opportunities for engagement that the activity makes available to students.

Using Nasir and Hand's idea of practice-linked identities and incorporating Gutstein's notion of a social justice pedagogy, this paper examines how students engaged with computer science ideas as they designed and implemented a mobile app with relevance for their community. Using interview data, presentation transcripts, student project artifacts, and field notes from a case study in an after school program, we argue that such an approach can (1) give students access to the domain of computer science, (2) provide multiple and evolving ways in which students are able to demonstrate competency, and (3) allow students' to connect their own experiences and values to larger social issues.

2. BACKGROUND

In the spring of 2013, the first and second author co-taught a twelve-week *Apps for Social Justice* (mobile applications) course, which met once a week in the evening for two hours. The course was sponsored by OSMO – an after school STEM program in the San Francisco Bay Area that connects university mentors with low-income middle and high school students of color. OSMO provides a number of services for local youth and families, including weekly homework help sessions, a parent support group, and a variety of STEM enrichment classes.

Our research team provided instruction, technical support and mentoring to two student teams of middle- and high-school youth. The first student team – the focus of this paper – consisted of two female African-American high school students, Soraya and Mayra,[1] who were continuing work on an app idea that they had had formulated during the previous semester [15]. Soraya and Mayra's application, BAYP (Bay Area Youth Programs), was intended to support local teenagers' academic and extracurricular needs by helping teens to search for positive youth organizations. The second student team consisted of four students who were new to the program and were just beginning to devise their own mobile application idea.

Because Soraya and Mayra's team worked primarily on software implementation[2] (as compared to the brand new team's initial focus on ideation and design), Soraya and Mayra will be the focal point for this paper, so that we may highlight some of the ways in which the *Apps for Social Justice* course introduced the students to various computer science concepts. We highlight Soraya and Mayra's experiences to demonstrate *how* the course supported computer science learning. As we will show, Soraya and Mayra made remarkable progress over the course of the twelve weeks, iterating on their initial design and implementing a functional Android application that could query and update a live database of local after school programs, which the students designed themselves (Figure 2). BAYP's various screens interacted with the database in different ways, showing program listings, detailed information pages, and maps of locations. Soraya and Mayra presented BAYP to family and friends at OSMO's year-end student showcase (Figure 1).

Both Soraya and Mayra were high achieving, college bound students who attended two different high schools and did not know each other before their enrollment in OSMO. Both students' parents were actively involved in their daughters' academics, and they regularly communicated with program mentors about their daughters' academic progress. Neither student had taken a computing course in school, but Soraya had participated in another out-of-school program, 'Technovation,' where she learned about App Inventor [11], the programming environment that was used to develop BAYP.

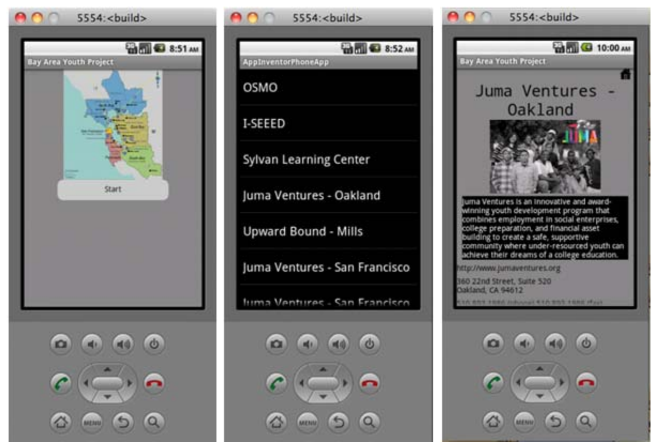

Figure 1: Screenshots of tbe BAYP mobile app, which allows users to browse for local, youth-oriented after school activities.

3. METHODS

During the twelve-week course the research team took field notes, documenting how Soraya and Mayra negotiated their responsibilities and worked together; engaged with social justice ideas as they proposed and refined their designs; used the various design and development tools; and applied various computer science principles to their app's functionality. We also recorded (video and audio) Soraya and Mayra's various practice presentations, as well as their final presentation to parents and community members, in order to capture how their official 'pitches' changed over time with respect to the social justice and computer science ideas they conveyed. In addition, we archived their weekly code iterations, prototypes and mock-ups during various stages of their design process, and their PowerPoint presentations. Finally, we interviewed Soraya and asked her to reflect on her experience in building BAYP, and how it might be relevant to her future academic pursuits. We were unable to interview Mayra, who was away at college.

4. FINDINGS

Because OSMO focuses primarily on teaching STEM, our research team was curious to explore how a more integrative learning environment – which engaged design thinking skills and a social justice orientation – might foster a more contextually

[1] The names of the study participants are all pseudonyms.

[2] In the previous semester, Soraya and Mayra had focused on ideation and design, and had taken several App Inventor tutorials. Thus, they were primed to begin implementing their ideas at the very beginning of the twelve weeks.

grounded approach for learning computer science. Using the BAYP project as an illustrative example, this paper will examine the learning environment in terms of (1) how it gave students access to the domain of computer science, (2) how it provided multiple ways in which students could demonstrate competency within the project, and (3) how it allowed students' to connect their own experiences and values to larger social issues.

4.1 Access to the Domain

Nasir and Hand define access to the domain as "the extent to which participants have the opportunity to learn both about the practice as a whole and about the specific tasks and sub-skills that make up domain knowledge" [13]. In the context of computer science, while the domain is limited by the Turing model of computation, CS has an ever-growing body of algorithms, design patterns, and best practices that can get quite sophisticated, and it therefore takes quite a while to become adept at using the tools of the language in meaningful ways. Thus, it is important to introduce learners to CS in ways that are intuitive and conceptually generative; incorporate diverse interests; and allow for some control over what is learned (and how fast).

4.1.1 Scoping the Domain

In the practice-oriented learning environment of app development, Soraya and Mayra had the ability to scope the computer science domain to only those concepts needed to build their proposed app's functionality. Together, they brainstormed various features of the BAYP app (which involved a Google map, a rating system, automatic notifications and alerts, categorization of activities by type, and the display of pertinent organizational information like price, hours, descriptions, a photo, etc.), and worked with mentors to translate their ideas to specific functional elements within the design. Hence, unlike a traditional introductory course which tends to focus on specific algorithmic concepts (variables, control structures, loops, etc.), Soraya and Mayra's context-driven list of design features drove the corresponding list of computer science topics: databases, HTTP, user interface design, loops, if/else statements, and event handlers. For example, consider the following transcript, which is an excerpt from a Q&A session immediately following Soraya and Mayra's final presentation:

1. S: "And we're not done yet...we're going to add tons more."
2. M: "Our future plan is, like, if you don't have like a...an Android or a smartphone, we're going to create an actual web page to find after school programs to put their kids in, and also we're thinking about (looks at Soraya)...Oh, we're thinking about partnering with schools and different after school programs that we have on this list to actually send notifications and updates to the app. And with the schools, we want to give it to the counselors or principals, so like if they see students who are like off-track, they can pull them in by showing them different after school programs. And we're also going to have a review page, saying if you guys like the app."
3. A1: "Like Yelp?"
4. Soraya and Mayra: "Yeah."
5. A1: "So, can you search...can you search by location?"
6. M: "Mmmmm. Not yet."
7. A1: "we believe you, we believe y'all will figure it out."
8. M: Smiles. "We'll figure it out."
9. S: "We want to do it by location, by, like, price..."

S = Soraya
M = Mayra
A1, A2 = Audience members

From the new set of enhancements that Soraya and Mayra listed above, we can now add HTML/CSS/JavaScript, SMTP (simple mail transfer protocol), SMS (text messaging), spatial querying, and a slightly more sophisticated database schema to their set of relevant CS domain knowledge. Hence, design continues to iteratively expand Soraya and Mayra's CS horizon, and because these concepts are motivated by genuine design needs, when it comes time to learn these skills, Soraya and Mayra will understand why they're important and what function they serve (discussed below). They also understood that this list need not be exhaustively tackled: although the SMS idea came up several times in design discussions before the final presentation, Soraya and Mayra ultimately decided not to pursue that avenue because it simply wasn't one of their top priorities, given all of the other features they wanted to implement. Thus, the creative process also allows students to not only enumerate the relevant computer science topics, but also to select which concepts to learn first. Though it was our role as mentors (instructors) to make suggestions and try to give Soraya and Mayra feedback regarding the technical complexity of their ideas, it was ultimately the students' responsibility to decide which concepts to learn.

4.1.2 A Needs-Driven Approach

Not only can a bottom-up needs assessment scope the domain, but it can also help to make the domain more intuitive. Consider the following excerpt, also taken from Soraya and Mayra's post-presentation Q&A session with the audience:

1. *A2*: "What are the starting options again [referring to the browsable categories of activities that users of the app could select]?"
2. *M:* "Let me show you the Fusion Table [uses the projector to display the Fusion Table onto the screen].... Wait, do you want to see, like, the..."
3. *A1:* "Like 'peer mentoring,' 'tutoring'...that?"
4. *A2:* "Yeah."
5. *S:* "So like, the categories now are 'peer mentoring,' 'tutoring,' 'employment,' 'science,' 'college prep,' 'math,' 'history,' 'social studies,' 'sports,' and 'music.' But we're going to be adding as time goes on to make this more [inaudible]."
6. *A1:* "And they're cross-listed, right? So OSMO's under science, math, and tutoring?"
7. *M:* "Right. Umm hmm. If you look at our Fusion Table [points to the Fusion Table on the screen], you can see that

Figure 2: The BAYP database, created using a Google Fusion Table

Organization Name	URL	Address	Categories	Verified
OSMO	http://www.oaklandscience	1625 Clay St. Suite 600, Oakland, California	Science, Math, Mentoring, Tutoring, College Prep	1
I-SEEED	http://www.facebook.com/I	1625 Clay St. Suite 600, Oakland, California	Social Studies, Academic Achievement, Science, Men...	1

we have a slot called category [referring to the column heading]. And under the categories...[she points to an organization record in the table, showing that each record has a comma-delimited list of categories underneath it (Figure 2)]."

8. *A2:* "Oh, we see, they have multiple..."
9. *A1:* "So when somebody goes and adds an organization..."
10. *M:* "We have to go in and verify it." [Points to the "Verified" column header]
11. *S:* "This is where we...it [the organization] would be up for a while, but we could always go back in and change it."
12. *A:* "That's smart."
13. *M:* "Over here, we have 1, 1, 0 [reading down the column]. That tells you if it's verified or not. And if it's not verified, we can go on here and delete it."

Soraya and Mayra had created their "youth programs" database using Google Fusion Tables [4], a web-based spreadsheet for organizing, storing and retrieving data (Figure 2). During the design process, Soraya and Mayra decided which attributes to collect (hours of operation, location, etc.) and an appropriate set of categories ("peer mentoring," "tutoring," "employment," etc.), and slowly populated their tabular repository of resources. Because Soraya and Mayra had designed this table by envisioning the kinds of questions that their users would ask the BAYP app (e.g.: "Show me all of the dance programs"), we conjecture that the task of learning how to query the database became more accessible. A *data-first* approach to learning – where students build their own databases and formulate questions and ideas about how this information might be used – could provide an alternate way to motivate CS learning and to make it more intuitive and relevant.

4.2 Multiple, Shifting Roles

Another benefit of embedding a CS curriculum within a design pedagogy is that the process requires many competencies: gaining a deep, contextualized understanding of the problem you are trying to solve; identifying your users and their potential needs; understanding constraints; conducting a 'social analysis'; communicating design ideas and advocating for them; and determining the precise features of the application. How will information be stored, updated and maintained? What will the screens look like? Who will have access to the app? How will users find what they need? Each of these needs and questions requires a particular kind of expertise, which in turn provides many important roles to be filled in a design project. This closely parallels Nasir and Hand's notion of *integral roles*, which they define as "the extent to which participants are held accountable for particular tasks in a practice and are expected to become competent and even expert in a subset of activities that are essential to the practice." During the BAYP design and development process, Soraya and Mayra each took on and carried out multiple roles, building upon their existing competencies and gaining exposure to ways in which computer science learning could be applied to their own interests.

4.2.1 From Manager to Designer to Developer

Over the course of the twelve weeks we observed Mayra – who was initially unwilling to engage in some of the computer programming tasks – gradually become more interested in programming. Leading up to the study, she had taken on the role of the "marketing manager" [15], preferring to participate in the non-technical aspects of the project that involved writing, and creating presentations and graphics. However, as she and Soraya began to discuss the specific functionality of the app in detail, Mayra became interested in the more technical aspects of BAYP. For example, she began drafting technical sketches of the app (Figure 3a) to specify the information and options that would be presented to the user, and how each screen would look. From a CS perspective, this process: (1) scoped the many tasks that needed to be completed, serving as a to-do list; and (2) made a blueprint for each screen that precisely specified a discrete unit of work. When the research team introduced Mayra to Balsamiq [1], a drag-and-drop application prototyping tool that allows designers to create Android screens, she immediately began porting her sketches into the software, spending hours formalizing the BAYP design as she learned how to navigate the tool (Figure 3b). Without prompting, she explored how all of the different buttons, fonts, and widgets might represent her ideas, taking great care in her selection of colors and button positioning, and refining her design as she went. As her ideas transitioned from analog to digital, Mayra became interested in making her design "work" (so that clicking a button actually *did* something). When she realized that App Inventor also required skillful screen layouts, she became motivated to explore how button event handlers could be used to hide and show screen elements, and eventually learned to update the BAYP database using an event handler (Figure 3c).

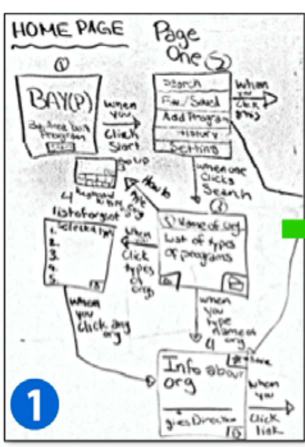

Figure 3a: Mayra's drawings

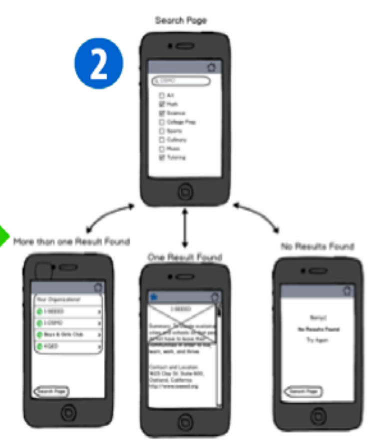

Figure 3b: Mayra's Balsamiq mock-up

Figure 3c: Mayra's App Inventor layout

Hence, the design-based sequencing of activities and tools appears to be a critical element in making computer programming more relevant, familiar and interesting. Translating ideas into fine-tuned, screen-by-screen designs, a focus of Human Computer Interaction (HCI), is one way of motivating the learning of CS concepts. The *Exploring CS* project, developed by researchers from UCLA, adopts a similar HCI-focused approach in their introductory high school curriculum [3].

4.2.2 Broadening Student Visions of Computing

While Soraya seemed comfortable with the idea of using App Inventor from the beginning [15], when we interviewed her around six months after the final presentation, she appeared to have broadened her vision of the kinds of roles and expertise involved in computer science. At the beginning of the twelve weeks, she told us that she liked programming, but that she was more of an "English person" than a "math person." However, after the design process, she had expanded her notion of the kinds of skills and competencies needed to be an app developer:

1. [7:55] First author: "So we were just wondering at this point in time...and obviously it's going to evolve and stuff, where you see computing fitting in, or tech, or design, or business or..."
2. Soraya: "I feel like no matter what I do, some type of digital design or coding or some computing will be involved, just because I find it fun to do it. And like I do like to draw and write and stuff, so like I want to have kinda like that liberal arts feel in my interests, but yeah, I really like coding, and I feel like through technology, you can have your ideas reach more people, so that's kinda like what I want. Yeah. Whether it's like working at a startup or making my own app and trying to be one of those few really rich kids who makes it."
3. First author: "So do you think that when you go to [university name omitted] or wherever you go, you'll take another computer science course?"
4. Soraya: "Definitely. Like when I applied to [university name omitted], they have a school called Society, Technology, and Innovation I think. So that whole school is all about taking your ideas and like using technology to kind of like broaden them and put them out there and enhance them. So you like go into the community...it's kind of like what we're doing now, but like in college. So I'm like, 'that's awesome.'"

Soraya says that she will definitely continue coding – a shift from her earlier stance – and yet she specifically mentions that she wants to incorporate "liberal arts" into her notion of computing (turn 2), where she can draw, write (turn 2) and work with communities (turn 4). Hence, her liberal arts interests no longer appear to be at odds with her notion of computing. Soraya now sees her technology, writing, and community-oriented interests as compatible and mutually reinforcing, and sees technology as a way to achieve a diverse set of goals.[3]

This indicates that design, as it applies to problem solving in the world, might be a way to attract a wider range of students to the computing discipline. Secondly, it suggests that an app-centric as opposed to systems-centric curriculum might be an appealing way to introduce computer science ideas to novices. Finally, it suggests that an overemphasis on math skills as an indicator of potential success in computer science may limit students' appreciation of computer science's relevance and potential.

[3] We have recently learned that Soraya was awarded a four-year scholarship to a top computer science program.

4.3 Connecting to Student Experiences

A final way that a community-oriented design process promotes computer science engagement is by positioning students as local knowledge experts. This idea closely connects with Nasir and Hand's third tenet of an engaging learning environment: *opportunities to make a unique contribution and feel valued*, which they define as the way in which students can incorporate aspects of themselves into the practice. However, the *Apps for Social Justice* course expands on this idea by incorporating Gutstein's notion of a *social justice pedagogy*, which connects structural aspects of students' lives (including social, economic, and environmental factors) into the practice of app-building, thereby allowing students to situate their own experiences within a larger social context. In combining these two elements, Soraya and Mayra were able to (1) draw from their own local knowledge (which improved their design) while also (2) justifying why their app (and by extension their own experiences) mattered in a broader context.

4.3.1 Local Knowledge

Soraya and Mayra drew from their own experiences as participants in after school programs to create well-informed specifications which detailed how other youth would search for interesting programs, what information would be important to display, and why it mattered that students have access to afterschool programs in the first place. Though as technical mentors, we had expertise in software development, it was Soraya and Mayra's creativity, enthusiasm, and knowledge of their communities and peers that made their app relevant and valuable. Because of their deep connection to the design context, they were able to skillfully construct scenarios of young people and their search for direction and activity in Oakland.

4.3.2 The Role of Social Justice

While connecting the app to students' experiences leads to a more informed design, situating students' experiences within a broader social context gives the app legitimacy. In the following transcript, also taken from the final presentation, Soraya and Mayra connect the importance of afterschool programs not only to their own lives, but also to larger structural issues of government divestment in youth:

1. M: "BAYP also provides all different types of after school programs, not just educational ones. So let's say your daughter told you that she wanted to be a flute player. And her school doesn't have a music class anymore because the government's not funding it. You could find an after school program with BAYP that has different music classes. We created BAYP because we saw this problem and we kinda wanted to attack it and in the same way fix our community."
2. S: "We think BAYP will help because it will disrupt social stereotypes in the world and our community. It's like reinvesting back into the futures of our youth in our community. All the potential in Oakland as you can see through this program, Everyone needs to have access to that."
3. M: "It will also even the playing field for all ethnicities and races and genders. So let's say that a girl was told that she couldn't be a basketball player because only boys play basketball. Different after school programs can give her a different mindset saying that she can do this, because everybody can do whatever they put their minds to."

Within this transcript, Soraya and Mayra argue that their app will play a role in addressing a variety of larger structural issues: government divestment in schools, social stereotypes, ‘evening the playing field’, racism, and sexism. However, rather than listing these issues indiscriminately, Soraya and Mayra relate them to their own personal experiences. By speaking to “all the potential in Oakland,” they are able to assert with confidence that after school programs give access to important activities and “mindsets,” and that their app is important.

Gutstein argues that in order to prepare young people to be leaders in the creation of a more equitable and just society, they must “understand, formulate, and address questions and develop analyses of their society” using the tools of mathematics [5]. From our experiences working with Soraya and Mayra, we believe that Computer Science is also a powerful tool with which to explore social issues because (1) the app development process can be easily geared toward addressing a community need, and (2) design provides an actionable way to study social inequities. Design has often been criticized as being overly “solution-oriented,” without a sufficient understanding of the situated context of design problems and their potentially structural causes. Having a social justice and user-centered design orientation can mitigate this tendency by encouraging students to spend time investigating the underlying issues that justify their app, leading to better and more impactful designs.

5. CLOSING REMARKS

In this case study, we have tried to show how a more practice-oriented computer science learning environment (1) gave students access to the domain of computer science, (2) provided multiple and evolving ways in which students are able to demonstrate competency, and (3) allowed students’ to connect their own experiences and values to larger social issues. This approach positioned students as local knowledge experts and gave students a way to apply their creativity and expertise to authentic community issues. It also helped to motivate computer science learning by giving students more agency in the concepts they chose to learn – specifically those that were most directly relevant to their goals.

We do not assume that our experiences working with Soraya and Mayra would be typical of any student team, given that Soraya and Mayra were already very hard working and motivated to begin with. Moreover, this activity took place in an out-of-school context, where students were self-selected, free to pursue their own interests without having to worry about performance or assessment, and could depend on the availability of dedicated, skilled mentors to guide their inquiry. That being said, we believe that a practice-oriented approach to learning computer science can provide students with an engaging and intuitive context for learning, which builds on student strengths. In future work, we hope to explore how such an approach might work in an in-school context, to examine what structures and support are needed to provide students with the skills and guidance they need to apply computer science to problems within their communities.

6. ACKNOWLEDGMENTS

We would like to thank the study participants, I-SEEED, and the Boys & Girls Clubs of Oakland (BGCO). In addition, this material is based in part upon work supported by the National Science Foundation under Grant Number IIS-1319849 as well as the Research in Cognition and Mathematics Education (RCME) pre-doctoral training grant.

7. REFERENCES

[1] Balsamiq: *http://balsamiq.com/*.

[2] Boaler, J. 2008. Promoting “relational equity”and high mathematics achievement through an innovative mixed-ability approach. *British Educational Research Journal*. 34, 2 (2008), 167–194.

[3] Goode, J. and Margolis, J. 2011. Exploring computer science: A case study of school reform. *ACM Transactions on Computing Education (TOCE)*. 11, 2 (2011), 12.

[4] Google Fusion Tables: *https://support.google.com/fusiontables/*.

[5] Gutstein, E. 2003. Teaching and learning mathematics for social justice in an urban, Latino school. *Journal for Research in Mathematics Education*. (2003), 37–73.

[6] Lewis, C. 2012. *Applications of Out-of-Domain Knowledge in Students’ Reasoning about Computer Program State.*

[7] Lewis, C. 2007. Attitudes and beliefs about computer science among students and faculty. *ACM SIGCSE Bulletin*. 39, 2 (2007), 37–41.

[8] Lubienski, S.T. 2008. On“ Gap Gazing” in Mathematics Education: The Need for Gaps Analyses. *Journal for Research in Mathematics Education*. (2008), 350–356.

[9] Margolis, J. 2008. Stuck in the Shallow End: Education, Race, and Computing. (Sep. 2008).

[10] Martin, D.B. 2003. Hidden assumptions and unaddressed questions in mathematics for all rhetoric. *The Mathematics Educator*. 13, 2 (2003), 7–21.

[11] MIT App Inventor: *http://appinventor.mit.edu/explore/*.

[12] Moll, L.C., Amanti, C., Neff, D. and Gonzalez, N. 1992. Funds of knowledge for teaching: Using a qualitative approach to connect homes and classrooms. *Theory into practice*. 31, 2 (1992), 132–141.

[13] Nasir, N.S. and Hand, V. 2008. From the court to the classroom: Opportunities for engagement, learning, and identity in basketball and classroom mathematics. *The Journal of the Learning Sciences*. 17, 2 (2008), 143–179.

[14] Schoenfeld, A.H. and Kilpatrick, J. 2008. Toward a theory of proficiency in teaching mathematics. *The international handbook of mathematics teacher education*. 2, (2008), 321–354.

[15] Vakil, S. 2014. A Critical Pedagogy Approach for Engaging Urban Youth in Mobile App Development in an After-School Program. *Equity & Excellence in Education*. 47, 1 (2014), 31–45.

[16] Wing, J.M. 2006. Computational thinking. *Communications of the ACM*. 49, 3 (2006), 33–35.