Trading Zones and Cultural Differences in Computer Science Education

Michael Lachney

Abstract
The purpose of this paper is to explore the role of "trading zones" in designing and implementing visual programming software that accounts for cultural differences in U.S. PK-12 computer science education. The paper builds on previous theory-work in computer science education that employs the concept of trading zones to build interdisciplinary standards for research. While standard making may create a creole discipline where trading partners have a new and shared vocabulary, it may also act as a form of assimilation, smoothing over and diluting differences between culturally distinct domains of practice. I introduce the field of ethnocomputing to show the strengths of pidgin trading zones—where exchange is possible but differences are maintained between trading partners—in creating disciplinary inclusivity. Instead of assuming computer science education can be neutral for all students, ethnocomputing confronts the diversity problem by framing cultural differences as assets to build upon as opposed to barriers to be overcome.

Keywords: trading zones; cultural differences; ethnocomputing; diversity; PK-12 education

Word Count: 6,889

1. Introduction

Computer science (CS) education research has developed as an interdisciplinary field (Fincher and Petre 2004). From its outset in the late 1960s with Seymour Papert and the Epistemology and Learning Group at MIT, CS education research has consisted of mathematicians, artificial intelligence researchers, social scientists, educationalists, and others (Papert 1980; Solomon 1986; Harel 1991; Papert 1993; Kafai 1994; Resnick 1997). While much of this work provides a strong foundation for the development of academic societies and journals, the interdisciplinary construction of the field has hindered the development of a shared paradigm and common research practices. This has inspired calls by CS education researchers to be more conscious about how they borrow from other disciplines in an effort to stabilize the field (Fincher and Petre 2004).

Fincher and Petre (2004) propose one pathway toward stabilization by borrowing the concept of *trading zones* from the history of science and science studies. Using the work of anthropologists to describe trading between different cultural groups, Galison (1997) developed the concept as a way to understand how disunified traditions between experimentalists, theorists, and instrumentalists are overcome in the physics laboratory. In particular he stresses that,

> cultures in interaction frequently establish contact languages, systems of discourse that can vary from the most function-specific jargons, through semispecific pidgins, to full-fledged creoles rich enough to support activities as complex as poetry and metalinguistic reflection. (Galison 1997, p. 783)

Fincher and Petre (2004) argue that framing CS education research in terms of trading zones helps reveal that it is currently equivalent to a pidgin: sufficient only for limited epistemological trade between its different disciplines. In an effort to create "distinct disciplinary norms and practices" for CS education research, they propose an active effort to move from a pidgin to a complete and unified trading zone, otherwise known as a creole. While there has been important work to create creole trading zones between CS education research and gender studies in an introductory CS course using the theme of "gender neutrality" (Svedin and Bälter 2016), this paper problematizes the applicability of identity neutrality when creating a creole trading zone to confront the problem of racial and ethnic underrepresentation in U.S. PK-12 CS education. As I will illustrate with the visual programming language Scratch, a neutral position runs the risk of assimilating cultural differences into the white middle class status quo, reproducing color-blind racism in CS education.

Ethnocomputing provides an alternative approach to the concept of trading zones. It assumes the value of trading zones lies not only in the creation of a creole, but also in the maintenance and care of cultural differences through building on the strengths of CS education research as a pidgin. While the creation of creole trading zones for CS educational research should continue to be explored, this paper makes the case that part of the value of trading zones includes the work pidgins can do to highlight cultural differences and power relationships between trading partners.

2. Trading Zones

In his cultural-historical study of physics, Galison (1997) shows three autonomous subcultures within the field: experimentalists, theorists, and instrument makers. While Galison (2010) affirms Kuhn's (2012) historical observation of "diachronic incommensurability" between paradigms—for example the paradigm shift from Newtonian Laws to Einsteinian special relativity—he also observes "synchronic incommensurability" between these subcultures at a given period in time (p. 27). In other words, the autonomy of theorists—who have their own journals and societies—may result in beliefs about black holes, for instance, that are different from those of their experimental colleagues. Still, there is the reality of local coordination and action between subcultures in the face of synchronic incommensurability. How do subcultures talk to each other and collaborate at a local level? Galison turns to work in linguistic anthropology on language exchange across cultures. Dutton (1983) provides a foundation to show how languages can hybridize at different scales. Adapting this work to scientific knowledge production, Galison develops the concept of *trading zones* to describe how subcultures communicate while either maintaining disunity or forming into a new unified discipline.

The key to understanding exchange in trading zones is that there is no presupposed universal meaning of the object being traded. Each party can still engage in trade without agreement on the significance of what is being traded. Pidgin trading zones limit meaning enough between two groups for the occurrence of trade without smoothing over cultural differences. Indeed, linguistic or schematic structures are simplified and reduced to necessitate trade. Galison (1997) uses trading practices between physicists and engineers in developing radar systems as an example of pidgin trading zones in science.

Both sides maintain separate identities while also collaborating. But sometimes, just as with ordinary language in trading zones, Galison's epistemological pidgins will change into creoles. Creoles in science are the development of new unified fields. For example, ongoing communication between mechanical engineers and biologists resulted in biomechanical engineering, which has its own journals, societies, and can be taught as its own discipline to new generations of students. It is important to note that just as linguistic anthropologists do not treat pidgins and creoles as inferior to "natural languages," pidgin and creole trading zones in science are not inferior to other locations of knowledge production since each has historically produced knowledge deemed legitimate by mainstream scientific communities. This symmetry of value between different levels of coordinated action provides insight into how disciplinary boundaries are created and the mechanisms behind interdisciplinary collaboration.

Both pidgins and creoles play roles in the success of interdisciplinary intersections, like large-scale physics projects. Pidgins allow functional efficacy without forcing linguistic and epistemological differences to vanish; as long as the engineer knows what to build and the physicist knows how the gadget corresponds to her concepts, each can keep living more or less in their own worlds. Creoles allow for fuller communicative experiences, but at the expense of creating a separate "neutral" space of shared meaning. Physics includes both, at least from Galison's view, and thus might be described as a "polyglot" trading zone where pidgins, creoles, and perhaps other linguistic hybrids co-exist. While the term has been used in CS to generally describe multiple programming languages and paradigms at use in a single application (Wampler and Clark 2010), the polyglot concept has not caught on as a way to build the discipline of CS education, where there is more emphasis on the neutrality of creoles.

2.1. CS Education

Computer science education research employs the concept of trading zones for both discipline building and interdisciplinary coordination. Fincher and Petre (2004) make a stark observation that CS education research is "theory-scarce" and that it should stretch beyond its roots in mathematics and computation to borrow from trading partners in the social and learning sciences. They argue that to "build up a legitimate, well-founded thoughtful use of interdisciplinarity " requires the establishment of methodological and theoretical standards (Fincher and Petre 2004, 33). This is necessary so that theories from outside the reference discipline—in this case, CS education—can be interrogated by insiders who know why methods are valid and how theories can be used in knowledge production. Thus, the emphasis on a neutral creole: a healthy skepticism of trading partners means a criteria of rigor for creating shared practices that will move CS education research away from relying on pidgin trading zones.

Building on the goal to create a creole at the intersection of CS education and gender studies, Svedin and Bälter (2016) attempt to establish a trading zone where concepts concerning gender are recognized and standardized between partners. They are explicitly concerned with broadening gender participation in CS by studying retention in a gender-neutral, self-paced, and introductory university-programming course in Sweden. To confront stereotype threats that have increasingly discouraged women from pursuing CS, as well as the masculine cultures of individualism that subsume the CS field (see

Margolis and Fisher 2003 and Abbate 2012), Svedin and Bälter call for researchers to focus on both the numbers of women retained and the social conditions that produced exclusion in the first place.

Their development of a gender studies and CS education trading zone emerges from an awareness of two dangers: 1) CS education researchers may assume research results are due to innate gender differences and 2) feminist researchers may essentialize gender as a singular research interest. To overcome these dangers from each trading partner, they develop a research design to keep track of retention rates within the course along gender lines and develop course content that does not reproduce gender stereotypes. As one guiding design principal for standardization across the two disciplines, they developed "gender-neutral and non-biased messaging" in course content (Svedin and Bälter 2016, p. 198). The standard of neutrality is well suited for a creole trading zone, as it aims to be as inclusive as possible by eliminating references that speak to gender differences. Instead, they rely on the language of "students" in hopes of increasing the completion rate of women and not marginalizing any gender identity. They succeeded in increasing retention in the course: 5% increase for women and 7% increase for men (Svedin and Bälter 2016, 202).

2.2. Underrepresentation and CS Education

While U.S. women increased representation in CS from the late 1960s to the early 1980s, since then there has been a steady decline in their participation (Abbate 2012, 3). The gender-neutral trading zone between CS education research and gender studies that Svedin and Bälter (2016) created holds promise for broadening participation in light of this fact. However, there has also been research to suggest that, while gender-neutral standards appear to make changes at a surface level, they do little to challenge patriarchal structures and may even function to make organizational inequity invisible.

Kelan (2009) finds that policies of gender-neutrality in technology workplaces create a conflict of perspectives: "People realized that gender is a factor around which discrimination can happen, but at the same time, their workplaces were constructed as gender neutral, therefore making gender discrimination a concept no longer needed" (p. 206). Abbate's (2012) historical work on women and computing treats gender-neutrality not as a goal, but a political position in and of itself that can be overcome by recognizing "that technical practices have historically embodied unequal power relations" (p. 7). Given that history cannot be so easily erased through changes in course content, the ideological function of neutrality in CS education may actually normalize discrimination by suggesting post-gender curricula and schooling. Applying the same logics of neutrality to other forms of underrepresentation highlights how problematic neutrality frameworks can be for broadening participation.

Unlike women, African Americans have remained largely absent from CS since it stabilized as a discipline. To contextualize this absence in the 21st century, consider that in 2015 African Americans made up 13.2% of the population but were only 1.5% of CS PhD recipients (Dillon et al. 2015). Margolis et al. (2008) report that the racial gap in CS is partially due to disparities in high school course offerings and mentorship between schools with predominantly white students and those with predominately students of color. Margolis and colleagues found that students of color are presented with shallow CS

curricula that limit their career and educational opportunities. Given these circumstances, racially neutral language might be appropriate for goals to broaden participation.

Eglash (2007) interviewed black engineers about the "master-slave" metaphor used to describe relationships of control between two devices. While some engineers preferred it to be eliminated, identifying it as a barrier during their own schooling, others thought it revealed a history of power relationships that should be remembered through keeping the metaphor. The same debate might be created around the CS term "primitive." A primitive is the smallest unit of processing that a coder can execute in a programming language, but the term has also been used to denigrate the intelligence of American Indians and African Americans. This double meaning has resulted in the term being negatively associated with colonialism and white supremacy in local communities while also taking on assumed neutrality in CS communities. In some contexts, racially neutral language may appear more inviting for African American students, but at the same time, like gender-neutrality, it may cover up unequal power relationships already being reproduced in education.

As an early childhood educator, Delpit (2012) makes the case that the "basic skills" children learn in elementary school are only basic because of the assumption that white middle class conventions are a standard for all children to possess and build upon. Here, seemingly neutral terms, such as "basic literacy," are actually loaded with cultural values and norms. Delpit's point is not that we should abandon basic skills in education, but that what is considered "basic" may be different for students from different cultural backgrounds. Indeed, Delpit found that African American students came to school with basic skills that were considered advanced for white students. Superimposing the label of neutrality on CS may work to smooth over, instead of build upon, cultural differences between students, running the risk of reproducing "color-blind racism."

Color-blind racism disregards a person's race when considering issues of inclusivity and exclusivity by assuming everyone is inherently equal and success is only gained through individual feats. Bonilla-Silva (2006) makes the case that color-blindness allows whites to "enunciate positions that safeguard their racial interests without sounding 'racist'" (p. 4). Such politics are easily seen in U.S. debates surrounding affirmative action programs in universities. Opponents to affirmative action assume that the problems of entry into higher education for people of color are personal and have nothing to do with the structures of white privilege. In actuality, color-blindness ignores the facts that schools are more segregated today than they were forty years ago (Rothstein 2013), students of color are more likely to attend high-poverty schools than their white counterparts (National Equity Atlas 2016), and the prison-to-school pipeline—that has contributed to making the U.S. the global leader in mass incarceration—disproportionately affects students of color (Heitzeg 2016).

The reproduction of color-blind racism via racial-neutrality may appear to level the playing field, but it simultaneously has the potential to create the conditions where students of color must meet the standards of the hegemonic culture—the white middle class—to succeed. Instead of building on cultural differences as assets to inspire and engage student learning, schools and extra-curricular programs often ask students of color to assimilate. This is already part of popular science, technology, engineering, and mathematics (STEM) programs. The FIRST LEGO League, for example, does not build programming from the bottom-up, but instead takes a standard model and assumes it can

work equally across all educational settings; asking students to meet the same challenges regardless of their context.

Thus, in matters involving both race and gender, and in contexts ranging from early childhood to professional adults, the idea of neutrality as the exclusive solution to diversity in technical education is flawed. While the neutrality of creoles provides some advantages, they also have the potential to cover up power relationships that are still at work. Arguing against the use of neutrality in CS education research and proposing a model based on cultural differences proves problematic for standardizing the discipline as a creole trading zone. However, the risk of assimilating cultural differences into the existing CS culture needs to be taken seriously if trading zones are going to be useful for addressing diversity problems.

In the next section I will explore how creole trading zones in the U.S. music industry and CS education have resulted in assimilationist practices. I will follow this up with an alternative approach called "ethnocomputing" that aims to create not only creoles but also pidgin trading zones to address issues of underrepresentation in CS education. These types of culturally responsive computing programs, such as the Digital Youth Network (Barron et al. 2014) and CompuGirls (Scott et al. 2016), are designed to connect learning to students' situated contexts. My contention is that polyglot trading zones will need both pidgins and creole alternatives, and perhaps other forms of hybridity, to fully serve the needs of diverse populations.

3. Cultural Assimilation

While the history of African Americans in U.S. education has been marked by the violence of segregation (Spring 2016), the 21st century industry demand for STEM expertise has made the inclusion of minorities in these disciplines a national priority. Diversity initiatives based on equity, fairness, and justice in STEM education are important in and of themselves, but often the logics of diversification are the same as those of national economic competitiveness (Lachney and Nieusma 2015). These conflated logics of diversification and economic competition certainly motivate broadening participation, but they often fail to consider the role of STEM in producing underrepresentation in the first place (Eglash et al. 2017). For example, Downey and Lucena (1997) retell of hearing engineering recruiters explain to African American students that they need "to keep their resumes from looking 'too black,' or employers might become suspicious that a student was putting racial identity before engineering identity" (p. 134-135). It is hard to imagine recruiters telling white students to keep their resumes from appearing "too white," since the engineering disciplines have been historically designed as white spaces (Slaton 2010).

Multicultural mathematics has been one way to address the problem of diversity in STEM disciplines. These initiatives may take standard word problems but change the name of characters; instead of Jane counting marbles it is Esteban counting coconuts (Jennings 1996). The idea behind multicultural mathematics is to help minority students identify with the lesson content, seeing themselves or their cultural heritage represented. There is nothing inherently wrong with creating these trading zones, but they tend to use Western standards as foundational norms. So instead of looking at the embedded mathematics of logarithmic curves in Ghanaian Adinkra stamping traditions (Lachney et

al. 2016a), more likely lessons are on the numeric systems that map well onto Western mathematics, such as translating between the Egyptian and European base-ten counting systems (Bazin et al. 2002).

Certainly, multicultural mathematics are welcomed responses to Eurocentrism in curricula (Anderson 1997), but the danger is diluting cultural differences so much that they are no longer recognizable as different from the hegemonic culture. Indeed, Ladson-Billings and Tate (2017) argue that, "Instead of creating radically new paradigms that ensure justice, multicultural reforms routinely 'suck back into the system'" (p. 25). Following Alba and Nee (2017), I define the dilution in multicultural reforms using the concept of *assimilation*: a process where

> a cultural trait gradually loses its association with an ethnic group. In part, this happens because non-group members take it on, so that the empirical correlation between the trait and group membership is weakened. In part, it occurs as the trait is no longer labeled in an ethnic way. (p. 51)

The unification and standardization of communication that accompanies creole trading zones in multicultural education is precisely the danger of assimilation. Eglash et al. (2006) explain how multicultural mathematics "runs the danger of acting as a sort of safety valve, satisfying diversity requirements without challenging the most deleterious misconceptions" (p. 348).

Before showing a clear case of assimilation in the CS educational software Scratch, I will briefly describe the assimilation of hip-hop culture into the "white" music industry. In both cases, African American culture is diluted by cultural and economic forces that try to make it indistinguishable from white hegemony. I find that the case of musical appropriation is more intuitive for many readers: music, after all, is supposed to be culturally specific, and musical innovations are supposed to be credited to innovators. By deepening our understanding of how assimilation can be paired with exploitation in music, we can better spot its more subtle forms in CS education.

3.1. Assimilation in the U.S. Music Industry

The assimilation of African American music into white hegemony is nothing new. It has long been recognized that blues musicians provided an economic and artistic foundation for the successful careers of rock and roll artists and white music executives, despite the fact that they received little to no name or financial recognition (George 2003). Adelt (2010) shows the 1960s assimilation of black music into both the hippy counterculture movement and folk revival likely did not defy or fight against black oppression, but helped it persist. The historical legacy of diluting black culture enough that it appears white is not lost on the hip-hop generation. Musician Mos Def (1999) makes a point to reclaim these black cultural roots in his song "Rock N Roll" by stating, "Elvis Presley ain't got no soul, Chuck Berry is rock n roll. You may dig on the Rolling Stones but they ain't come up with that style on they own," before ending on hardcore punk stylistics with the call—"Who am I?!"—and response—"Rock N Roll!"—lyrics. This move by Mos Def makes clear that seemingly white genres such as punk rock own much of their development to the assimilation of black music (Parker 2017).

While most often associated with rapping MCs, hip-hop culture broadly includes break dancing, graffiti art, and DJ-ing. Since its beginnings in the Bronx during the mid-1970s, hip-hop has become a multibillion-dollar global industry. This necessitated the communities' continual challenge to the threat of assimilation (McLeod 1999). While early hip-hop artists resisted incorporation into mainstream recording practices, by the 1980s it had been identified as a profitable market. At one time, this gave attention to African American artists, who were able to commercialize their work and connect with other industry professionals for self and community promotion. At the same time, this also opened up many avenues for assimilation that stripped hip-hop of its cultural roots. Industry creations like Vanilla Ice and Rappin' Rodney Dangerfield highlight attempts to create a trading zone to treat hip-hop as racially neutral for white audiences by decontextualizing it from its socio-cultural origins. McLeod (1999) has studied how discourses of "authenticity" that surround hip-hop were used to resist these forms of assimilation and identify those who were part of the white hegemony. This struggle persists into the 21st century not only in the music industry but also in CS education to a smaller degree.

3.2. Assimilation in Computer Science Education

Scratch is an introductory visual programming language developed by a team at the MIT Media Lab, led by Mitchel Resnick. It is made up of blocks that users drag, drop, and snap together in a scripting panel to construct code for animations, games, graphic designs, and more. Instead of requiring the syntax of typed out code, Scratch uses these blocks to ease students' frustrations as they create scripts and debug the code to meet their desired outcomes. Since Scratch 2.0 was released, there has been a blurring between coding activities and a community site where youth upload their projects for others to view, play, and remix.

This process of remixing is central to the goals of Scratch as educational software; so much so that Scratch is named after DJs in the hip-hop culture who appropriate and remix existing music (Kafai and Bruke 2014). The term *scratch* "refers to a vinyl recording whose playback has been altered by hand in order to produce rhythm" (Goldberg 2004). While this connection appears to be an exciting opportunity to inspire African American students' interest in CS, references back to hip-hop in Scratch are highly diluted. The most striking example to consider is Scratch's signature icon, "Scratchy" the orange cat (See Figure 1). While easily removed and replaced by users in the actual software, Scratchy appears each time a new project is started, in addition to appearing as a guide in tutorials and user resources. Even though it is common to refer back to hip-hop DJs when explaining where the software acquired its name, this reference has been toned down to the point that the origin story can be easily missed or forgotten. Scratchy the cat does not help since "scratch" can be read in reference to the character's claws. Here, African American culture is diluted by Scratchy—as an icon for CS—to the point that it does not necessitate a reference back to scratching records in hip-hop culture.

Figure 1. Scratchy the orange cat made out of LEGOs at the MIT Media Lab.

The history of African American music and the recording industry includes many attempts to bring together white audiences with blues and hip-hop in ways that can weaken their associations with African American communities. It is, in other words, a history of erasing the origins of a source of cultural capital in order to encourage its sale and acceptability to white audiences. I have identified this as assimilation. Similarly, Scratch has assimilated the cultural practice of scratching records together with programming such that the activity no longer appears as a trait of the hip-hop community. I suspect it is for the same reason: a desire to reap the benefits of its cultural capital, without risking any alienation from white audiences.

This is not to say that creoles should never be created. Eventually, through record companies like Def Jam and Bad Boy, the African American roots of hip-hop became part of the mainstream, and artists like Vanilla Ice no longer made the same kind of sense as part of industry strategies. This new creole trading zone between hip-hop and the music industry was not imposed from outside pressures but drew on the strong connection between local communities and the hip-hop culture. CS education does not have the benefit of these cultural roots. While Eglash (1999) makes a convincing case that the binary code that is essential to modern computing has origins in divination systems of West Africa, this history is rarely known or discussed in CS. Given this lack of a grounded community, how might CS education research move forward to create trading zones that bring programming and culture together without smoothing over and diluting cultural differences through assimilation? In the next section I will propose using ethnocomputing as a strategy for creating pidgin trading zones that maintain cultural differences between African American communities and CS. The goal is not to discourage the creation of a creole but highlight the importance for creating a polyglot of trading zones for CS education.

4. Ethnocomputing

During the 2013-2014 school year, I helped to run a STEM after school program in an urban Upstate New York public middle school. As part of this program, a teacher, software developer, and I used a variety of ethnocomputing applications called "Culturally Situated Design Tools" (CSDTs). Building on ethnocomputing research that reveals the computational thinking embedded in culture and the cultural value embedded

in computing, CSDTs help students explore math-culture and computing-culture connections through vernacular and indigenous designs (Eglash et al. 2006; Bennett 2016). One of these CSDTs is called Cornrow Curves, an applet to teach transformational geometry and computer programming through cornrow hair braiding.

One week during the after school program, I began a Cornrow Curves lesson by having students explore the background section that details the historical, socio-political, and cultural significance of cornrow braiding. Once students had completed their research and shared their findings with the group, I introduced them to the Cornrow Curves software. Standing at the front of a 7th grade science room, I, a white male, projected an early Java-based version of Cornrow Curves on the Smart Board for a tutorial lesson. Students, mostly African American, had small laptops sitting in front of them to follow along as I showed them how to navigate the Cartesian coordinate system, input values to use the different transformations—dilation, iteration, translation, and rotation—and add virtual braids. I tried to be sensitive to the fact that many African American students are probably well aware of cornrow braiding techniques, so I prompted them to participate in the explanation of the software. Some did, but the most dramatic contribution happened after I had finished the tutorial.

Later that week as I walked around the room, looking over students' shoulders while they played with the software, one girl, sitting in the back of the room, raised her hand. While this student was regularly quiet and soft spoken during after school, she often attended the voluntary program and participated in the majority of activities, including a youth-led research project on student health and a documentary video on the mathematics of knitting. I walked over to her, anticipating a question about the software. While she was struggling with the software, she did not have a question. Instead she had a comment: "this is not how cornrows work" (Field Notes 11/26/14). She explained her background with braiding, making clear she had the expertise to critique our software; expertise I could learn from to improve my own understanding of cornrow braiding.

Some software developers and educators may view this as a failure to account for a user-base. But this is exactly the type of interaction that ethnocomputing design and implementation affords by creating pidgin trading zones that maintain and highlight differences between community-based disciplines (e.g. African American hairstyling) and CS, while also affirming their relationships via computational thinking. The student was able to recognize these differences and build on them to: 1) reverse traditional classroom authority where adults give knowledge and students receive knowledge and 2) transgress hierarchies that create demarcations between technical and cultural knowledges. She did this using her local techno-cultural expertise in braiding to gain authority over computing and mathematics. Ethnocomputational design made this possible by limiting the meaning enough on both sides to facilitate trade, but not enough for each side to lose their unique identities in the process. The student exemplified the strength of a pidgin in CS education when she was able to maintain her identity as a braiding expert while learning how to use and critique Cornrow Curves.

4.1. Trading in Design and Implementation

Ethnocomputing design begins with fieldwork and collaboration between researchers, software developers, teachers, and community members; in the case of Cornrow Curves

the community members are hairstylists who have braiding expertise, the teachers have expertise in technology or math, and the software developers have expertise in coding. It is the job of the researcher to coordinate action between these different actors. In the development of Cornrow Curves this meant talking across the social worlds of hairstyling, software development, and education without necessarily having the expertise to contribute to them individually. The researcher acts as a "broker"—having feet in multiple social worlds to facilitate negotiation—to create points of contact that do not dilute cultural differences but instead limit meaning enough for all parties to come together (Lachney 2016). Each side—hairstylists, teachers, and developers—makes significant contributions and compromises.

Consider one compromise between the cultural fidelity of braiding and the human-computer interaction fidelity of the graphical user interface (GUI). Working with braiding experts, it is clear that their patterns start with small plaits—the series of y-shaped twists that make up a braid—that gradually grow in size as hair is added and the braid continues (i.e. it embeds the concept of iterative dilation). However, the Cornrow Curves default braid—like Scratchy, the image that appears when you first launch the software—reverses the order of the construction as it occurs in the traditional practice by starting with a large plait and proceeding to place successively smaller copies. The rationale for stating with a larger plait is two-fold: it is easier for users to concretely visualize the location in which their braid begins on the Cartesian plane, as well as its starting angle. Second, shrinking the braid uses a familiar scaling percentage such as 90%, whereas growing the braid requires a percentage larger than 100 (e.g. 105%). This is a clear example of the cultural practice giving up an element of authenticity to foster trade with CS. On the other hand, concrete visualization is not at all traditional to CS; it is accurate to say it is also giving up an element of authenticity to foster trading with braiding.

Some professionals might reject at the claim that visualization is breaking with the traditional culture of CS. After all, CS is used specifically for data visualization. But that ignores what actually constitutes CS as an identity and "field of cultural production" (Bourdieu 1983). Ask a lay user how they interact with their computer and they will describe concrete metaphors: the old office analogy of files and folders on a desktop. Ask a CS professional and they will describe text inputs on a command line. A web authoring tool versus raw HTML; GitHub GUI verses GitHub command line; etc.

Abstraction—the processes of allowing contextual details to be dealt with at a lower level of the technology, while the human deals with a simplified upper level—has long been a hallmark of CS problem solving. The first digital computers used binary numbers or "machine code." Assembly languages replaced machine code with more human readable symbols. For example, the machine code 00000101, which causes the central processing unit to subtract ("decrement") a number from the storage spot "registrar B" would be represented in assembly language as "DEC B." By the time a user has told the computer that the number to be subtracted is 1 and where to put the difference, three lines of code have been written. Further abstraction in a programming language allows a single line of code: 5-1=4. The user does not have to deal with where it stored the original value, puts the new value, what to do as storage space is used, and so on—all that is abstracted away. Today's programming languages often call on code libraries in which thousands of lines of code are represented by a simple call to a function

or file. For example, in Windows the dynamically linked library (DLL) file GDI32.DLL contains functions for drawing graphics, displaying text, and managing fonts.

These examples describe instances where abstraction is helpful in terms of computing power and generalizability. However, I contend that there is a need to make a distinction between the functional abstraction in the examples above, and what might be called its cultural aesthetic in CS education. Figure 2 is a typical example of a nested loop found in the programming language Processing. The outer cycle shown here moves along each position change in height; the inner loop renders the image of squares arranged in a row at the specified height. In the *functional* sense, there is nothing more "abstract" about showing a Cartesian grid of squares than showing a Cartesian grid of circular Native American beads. But in a cultural aesthetic sense, many CS educators would assume that the grid of squares is more abstract, and somehow better matches what children need to become CS professionals.

Why are we prone to conflating functional abstraction and cultural aesthetic abstraction? Consider two social contexts: in one, a white middle class student asks about applications for figure 2, and I say that the squares are like a floor plan when you have someone remodel the tiles in your kitchen. In the other, a native student asks about applications for a grid of circles, and I say "think of the thousands of different patterns familiar to any native beadworker." The squares seem more abstract and generalizable to most of us because most people spend little time making bead patterns.

Lachney et al. (2016b) identify the claim that the grid is pedagogically more appropriate for CS as "content agnostic" software design: "designers must, throughout the software development process, remain 'agnostic' as to the content that learners will create" (para. 1). Two features are at work here. First, the social context of floor tiles provides an unspoken assumption about a shared cultural background. Second, this conflates the cultural aesthetic of abstraction with functional abstraction. Both errors diminish the diversity of available pathways for CS engagement.

In contrast to the nested loops in figure 2, figure 3 shows an equally "abstract" (in the functional sense) nested loop in the GUI of Cornrow Curves. The output of four braids with twenty-five plaits each is concretely connected to local knowledge in African American communities. The design of Cornrow Curves is "content aware": "educational software is designed to provide students with a kind of 'value added' orientation or balance" (Lachney et al. 2016b, para. 29). I refer to this as "interests-driven" design. While all design includes social interests, the term "interests-driven" indicates an effort to make these more visible, deliberate, and responsive to the needs and values of the community of users. The values added in this case are the interests of cosmetology and natural hair braiding, which have entrepreneurial (Wingfield 2009) and political (Gill 2010) significance for African American women and their communities.

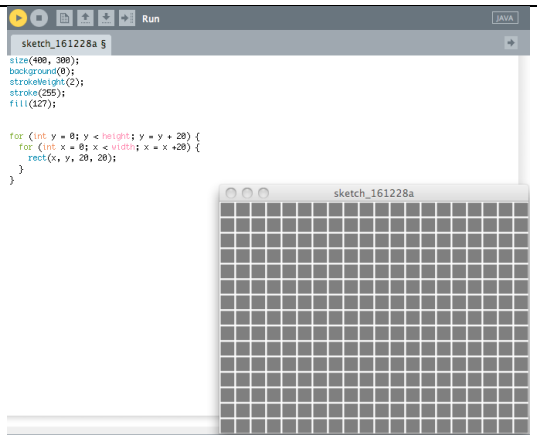

Figure 2. Processing example of a nested loop

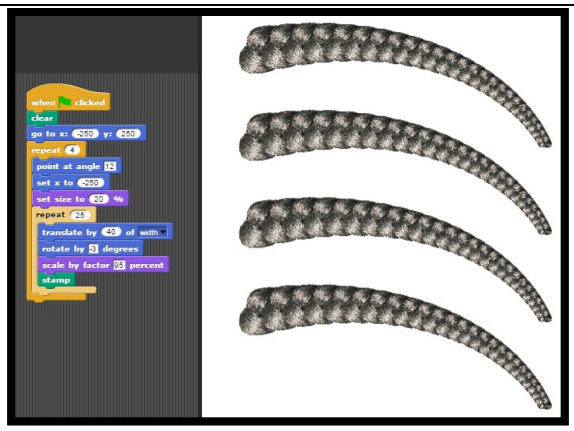

Figure 3. Cornrow Curves example of a nested loop

While some educators worry that aesthetically contextualizing CS will limit the possibility for students to transfer that knowledge to other domains—an empirical question that has yet to be tested—giving up some aspects of abstraction creates the possibility to both diversify who enters the field and also diversify the output of what is considered relevant to CS (Eglash et al. 2017). This output proves most valuable for creating what we might call "physical pidgins" – in this case the materialization of trading zones between classrooms and salons, by which students are able to trade the techno-cultural knowledges of CS and braiding. For example, during the design and development of the JavaScript version of Cornrow Curves the CSDT team began working with Angela, an African American entrepreneur and hairstylist in Upstate New York. Twice during 2016 Angela helped introduce students to the computational thinking involved in cornrow braiding by using physical braiding techniques on mannequin heads (see Figure 4). At a workshop with an Upstate college, Angela helped students learn about the computational thinking of good braid designs using materials from her salon (e.g. mannequins, combs, etc.). This was followed by students making designs in Cornrow Curves, voting on the designs that best fit the criteria for good braids, and then 3D printing (see Figure 5) the winning design. The majority of students' designs were later 2D printed and displayed in Angela's salon.

Figure 4. Angela introduces a group of students to braiding techniques using CS and math vocabulary.

Figure 5. 3D printed mannequin head with the winning cornrow design.

Once we eliminate the confusion with aesthetic abstraction, it should be clear that CS does not have to give up any functional abstraction when contextualized by the computational thinking embedded in cornrow braiding. The added value in implementation is that the diversified interface also diversifies the entry points for people to engage CS. From the educational side, students that might normally be alienated from computing can find engagement; from the beautician side, computing concepts can enrich their discourse and lower barriers to cosmetology students' success in CS and beyond.

5. Conclusion

I envision the relationship between the two physical pidgin trading zones of salon and classroom as a kind of feedback loop through a shared commons; what Eglash et al. (2017) call "generative justice." There are of course shallow versions of culture-based education: the coconut example referred to previously, changing skin color of characters in educational games, etc. In such cases, the cultural materials are typically there to lead students away from culture, to the "real" math or computing. Generative models like ethnocomputing, in contrast, have "recursive depth" (Banks 2016) in which value circulation can lead to innovations in synthesis: in the case of cornrows the move from single braids to a "braid block" that can be dragged and dropped into a visual script; exploration of increasingly complex styles; and 3D printing.

The traditional diversity model that uses the metaphor of a "leaky-pipeline"—where underrepresented students leak out at different points on their ways to the CS workforce—fails to account for much of the problems with CS that produce underrepresentation in the first place; not least of which includes processes of assimilation imposed by positions of neutrality that alienate students from school structure and curriculum. As opposed to the linear pipeline model that extracts value from underrepresented communities for a global workforce, generative justice replaces extraction with recursion by identifying and strengthening opportunities for CS value, in its many forms, to circulate among the underrepresented communities where it was produced.

The pidgin trading zones that Angela helped to create exemplify this recursive metric: students may find their way to CS and cosmetology through the technology classroom that uses cornrow braiding or the salon that highlights computational thinking, increasing the opportunity for recursivity between the two. The hypothesis is that these ethnocomputing inspired pidgins will be "generative contexts" that attract a greater diversity of underrepresented students to CS while also keeping the value that these students produce in their local communities (Lyles et al. 2016). Lyles et al. (2016) explain that technology education that accounts for local knowledge can help foster the pre-conditions for building sustainable pathways for circulating unalienated and anti-authoritarian forms of value between educational and community settings. If this hypothesis is correct then it may be possible for a creole to emerge from that context. This highlights the strength of focusing on pidgin trading zones in CS education research: they help to create these conditions for an emergent, as opposed to imposed, creole alongside other pidgins in a trading zone polyglot. This decreases the risk of assimilation while also working to make the CS discipline more diverse from multiple directions.

References:

Abbate, J. (2012). *Recoding gender: Women's changing participation in computing*. Cambridge: MIT Press.

Adelt, U. (2010). *Blues music in the sixties: A story in black and white*. New Brunswick: Rutgers University Press.

Alba, R., & Nee, V. (2017). Rethinking assimilation theory for a new era of immigration. In Z. Valdez (Ed.), *Beyond black and white: A reader on Contemporary Race Relations* (pp. 44-70). Thousand Oaks: Sage Publications.

Anderson, S. E. (1997). Worldmath curriculum: Fighting Eurocentrism in mathematics. In A. Powell & M. Frankenstein (Eds.), *Ethnomathematics: Challenging Eurocentrism in mathematics education* (pp. 291-306). Albany: State University of New York.

Banks, D. A. (2016). Anti-authoritarian metrics: Recursivity as a strategy for post capitalism. *Teknokultura*, *13*(2), 405-438.

Barron, B., Gomez, K., Pinkard, N., and Martin, C. K. (2014). *The digital youth network: Cultivating digital media citizenship in urban communities*. Cambridge: MIT Press.

Bennett, A. G. (2016). Ethnocomputational creativity in STEAM education: A cultural framework for generative justice. *Teknokultura*, *13*(2), 587-612.

Bonilla-Silva, E. (2006). *Racism without racists: Color-blind racism and the persistence of racial inequality in the United States*. Lanham: Rowman & Littlefield.

Bourdieu, P. (1983). The field of cultural production, or: The economic world reversed. *Poetics*, *12*(4-5), 311-356.

Bright, A. (2016). The problem with word problems. *Rethinking Schools* 30(4).

Delpit, L. D. (2012). *"Multiplication is for white people": Raising expectations for other people's children*. New York: The New Press.

Dillon, E.C., Gilbert, J.E., Jackson, J.F.L., & Charleston, L.J., (2015). Expanding the pipeline, the state of African-Americans in computer science: The Need to increase representation. *Computing Research News*, 27(8): 2-6.

Downey, G. L., & Lucena, J. C. (1997). Engineering selves: Hiring in to a contested field of education. In G. L. Downey and J. Dumit (Eds.), *Cyborgs and citadels: Anthropological interventions in emerging science and technologies* (pp. 117 141). Santa Fe: The of American Research Press.

Dutton, T. (1983). Birds of a feather: a pair of rare pidgins from the Gulf of Papua. In E. Woolford and W. Washabaugh (Eds), *The social context of creolization* (pp. 77 105). Ann Arbor: Karoma Pub.

Eglash, R. (1999). *African fractals: Modern computing and indigenous design.* New Brunswick: Rutgers University Press.

Eglash, R. (2007). Broken metaphor: The master-slave analogy in technical literature. *Technology and Culture*, *48*(2), 360-369.

Eglash, R., Bennett, A., O'Donnell, C., Jennings, S. and Cintorino, M. (2006). Culturally situated design tools: Ethnocomputing from field sit to classroom. *American Anthropologist* 108 (2): 347-362.

Eglash, R., Babbitt, W., Bennett, A., Bennett, K., Callahan, B., Davis, J., Drazan, J., Hathaway, C., Hughes, D., Krishnamoorthy, M., Lachney, M., Mascarenhas, M.,

Sawyer, S., and Tully, K. (2017). Culturally situated design tools: Generative justice as a foundation for STEM diversity. In Y. Rankin and J. Thomas (Eds.), *Moving students of color from consumers to producers of technology* (pp. 132-151). Hershey: IGI Global.

Fincher, S., and Petre, M. (Eds.). (2004). *Computer science education research*. New York: Routledge.

Galison, P. (1997). *Image and logic: A material culture of microphysics*. Chicago: University of Chicago Press.

Galison, P. (2010). Trading with the Enemy. In M. E. Gorman (Ed.), *Trading zones and interactional expertise: Creating new kinds of collaboration* (pp. 25-52). Cambridge: MIT Press.

Gill, T. M. (2015). *Beauty shop politics: African American women's activism in the beauty industry*. Champaign: University of Illinois Press.

Goldberg, D. A. M. (2004). The scratch is hip-hop: Appropriating the phonographic medium. In R. Eglash, J. Crossiant, G. Di Chiro, R. Fouché (Eds.), *Appropriating technology: Vernacular science and social power* (pp. 107-144). Minneapolis: University of Minnesota Press.

George, N. (2003). *The death of rhythm and blues*. London: Penguin.

Harel, I. (1991). *Children designers: Interdisciplinary constructions for learning and knowing mathematics in a computer-rich school*. New York: Ablex Publishing.

Heitzeg, N. A. (2016). *The school-to-prison pipeline: Education, discipline, and racialized double standards*. Santa Barbra: ABC-CLIO.

Jennings, M. M. Rain-forest algebra and MTV geometry. *The Textbook Letter* November-December. http://www.textbookleague.org/75math.htm

Kafai, Y. B. (1995). *Minds in play: Computer game design as a context for children's learning*. New York: Routledge.

Kafai, Y. B. and Burke, Q. (2014). *Connected code: Why children need to learn programming*. Cambridge: MIT Press.

Kelan, E. K. (2009) "Gender fatigue: The ideological dilemma of gender neutrality and discrimination in organizations." *Canadian Journal of Administrative Sciences/Revue Canadienne des Sciences de l'Administration* 26 (3): 197-210.

Kuhn, T. S. (2012). *The structure of scientific revolutions*. Chicago: University of Chicago Press.

Lachney, M. (2016). Culturally responsive computing as brokerage: toward asset building with education-based social movements. *Learning, Media and Technology,* 1-20. DOI: 10.1080/17439884.2016.1211679

Lachney, M., & Nieusma, D. (2015). Engineering Bait-and-Switch: K-12 Recruitment Strategies Meet University Curricula and Culture. In *Proceedings of the 2015 American Society for Engineering Education annual conference & exposition*.

Lachney, M., Bennett, A., Appiah, J., and Eglash, R. (2016). Modeling in ethnocomputing: Replacing bi-directional flows with recursive emergence. *International Journal for Research in Mathematics Education* 6 (1): 219-243.

Lachney, M., Babbitt, W., and Eglash, R. (2016). Software design in the 'construction genre' of learning technology: Content aware versus content agonistic." *Computational Culture: A Journal of Software Studies 5*.

Ladson-Billings, G., & Tate, W. F. (2017). Toward a critical race theory of education. In A. D. Dixson, C. K. R. Anderson, and J. K. Donnor (Eds.), *Critical race theory in education: All God's children got a song* (pp. 11-30). New York: Routledge.

Lyles, D., Lachney, M., Foster, E., & Zatz, Z. (2016). Generative Contexts: Generating value between community and educational settings. *Teknokultura*, *13*(2), 613-637.

National Equity Atlas. (2016) "School Poverty United States" http://nationalequityatlas.org/indicators/School_poverty/Over_time%3A35536/Uited_States/false/School_type%3APrimary_schools

Margolis, J., and Fisher, A. (2003). *Unlocking the clubhouse: Women in computing*. Cambridge: MIT Press.

Margolis, J., Estrella, R., Goode, J., Holme, J. J., and Nao, K. (2008). *Stuck in the shallow end: Education, race, and computing*. Cambridge: MIT Press.

McLeod, K. (1999). Authenticity within hip-hop and other cultures threatened with assimilation. *Journal of Communication*, *49*(4), 134-150.

Mos Def. (1999). *Black on both sides.* Rawkus and Columbia Records.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books.

Papert, S. (1993). *The children's machine: Rethinking school in the age of the computer*. New York: Basic Books.

Parker, J. (2017). Writing and unwriting race: Using hip-hop in writing and literature classrooms. In F. Condon and V. A. Young (Eds.), *Performing antiracist pedagogy in rhetoric, writing, and communication*, (pp. 195-209). Fort Collins: Colorado StateUniversity Open Press.

Reardon, S. F., & Robinson, J. P. (2008). Patterns and trends in racial/ethnic and socioeconomic academic achievement gaps. In H. F. Ladd and E. B. Fiske (Eds), *Handbook of research in education finance and policy* (pp. 497-516). New York: Routledge.

Resnick, M. (1997) *Turtles, termites, and traffic jams.* Cambridge: MIT Press.

Rothstein, R. (2013). For public schools, segregation then, segregation since. *Economic Policy Institute. Retrieved July*, *31*, 2014.

Scott, K. A., Sheridan, K. M, and Clark, K. (2015), Culturally responsive computing: A theory revisited. *Learning, Media and Technology* 40 (4): 412-436.

Solomon, C. (1986). *Computer environments for children: A reflection on theories of learning and education.* Cambridge: MIT Press

Spring, J. (2016). *Deculturalization and the struggle for equality: A brief history of the education of dominated cultures in the United States*. New York: Routledge.

Svedin, M., & Bälter, O. (2016). Gender neutrality improved completion rate for all. *Computer Science Education*, *26*(2-3), 192-207.

Wampler, D., & Clark, T. (2010). Guest editors' introduction: Multiparadigm programming. *IEEE Software*, *27*(5), 20-24.

Wingfield, A. H. (2009). *Doing business with beauty: Black women, hair salons, and the racial enclave economy*. Lanham: Rowman & Littlefield.