Check for updates

The House of Computing: Integrating Counternarratives into Computer Systems Education

Mara Kirdani-Ryan
Paul G. Allen School of Computer Science & Engineering
University of Washington
Seattle, WA, USA
marakr@uw.edu

Amy J. Ko
The Information School
University of Washington
Seattle, WA, USA
ajko@uw.edu

ABSTRACT

Social upheaval through widespread disinformation, aggressive automation, and algorithmic oppression have led to an increasing focus on the ethical considerations of technologists. In response, researchers and educators have looked to integrate ethics into Computer Science curricula, either by creating ethics-exclusive courses or embedding ethics into existing computing topics. Regardless of approach, few ethics integrations seek to explicitly center counternarratives, narratives opposing dominant narratives within computing, as a method of instruction. Given an existing teaching opportunity, our prior experience with computer systems education, and a lack of existing ethics integrations into computer systems, we integrated counternarratives into an introductory systems course. We framed this integration through the *House of Computing (HoC)*, a structural metaphor that frames the computing discipline as an object for critique. Throughout the course, we presented counternarratives alongside technical content. We assessed student understanding of counternarratives through "floorplans": metaphorical representations of course units, or floors within the *HoC*. Through an analysis of students' first floorplan, we found that nearly every student expressed existing or newfound awareness of structural problems within computing, though the novelty of the floorplans concept challenged students. Based on this experience, we offer recommendations for instructors looking to teach computer systems critically or integrate counternarratives into other computing courses.

CCS CONCEPTS

• **Social and professional topics → Computing education**.

KEYWORDS

Ethics, Counternarratives, Computer Systems

ACM Reference Format:
Mara Kirdani-Ryan and Amy J. Ko. 2022. The House of Computing: Integrating Counternarratives into Computer Systems Education. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2022), March 3–5, 2022, Providence, RI, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/https://doi.org/10.1145/3478431.3499394

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
SIGCSE 2022, March 3–5, 2022, Providence, RI, USA.
© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-9070-5/22/03…$15.00
https://doi.org/https://doi.org/10.1145/3478431.3499394

1 INTRODUCTION

Social upheaval through widespread disinformation [16], aggressive automation [13], and algorithmic oppression [4] have led to an increasing focus on the ethical considerations made by technologists. Computing accreditation requirements mandate that graduates of computing programs have an understanding of legal and ethical principles [1], but recent actions taken by computing professionals and technology companies [9, 22, 36, 45] demonstrate that fulfilling these requirements is insufficient to change the ethical practice of these actors. Critiques of technologists specifically, and the field of computing broadly [44], advocate for more comprehensive integration of ethics into computing curricula, whereby students would learn to consider the ethical implications of their work as a component of their engineering, programming, and design practice [11, 25].

While critiques and accreditation standards give little guidance for educators, prior work offers two primary approaches for addressing students' ethics education. One approach creates ethics-exclusive courses: spaces for students' socio-technical learning separate from their technical learning, most commonly covering law, privacy, European philosophy, and inequality [15]. These courses offer opportunities for in-depth and comprehensive approaches to ethics education and avoid conundrums where ethics inclusions are contingent on "if time allows" [18]. Prior work, however, argues that standalone courses allow students to view their ethical considerations as separate from their engineering work, rather than as an integrated component of their practice [10, 14, 20, 43]. As an alternative, a second approach embeds students' ethics education into existing technical coursework. This approach spans several decades of scholarship [20, 31], and its popularity has increased recently, with embeddings within artificial intelligence [18], data science [3], machine learning [37], human-computer interaction [38], and introductory programming courses [14, 34], among others. While both approaches have their benefits, we center ethics embeddings in this work because they require less administrative intervention and offer the opportunity to critique computing in context.

Within courses that embed ethics, instructors have taken several approaches: adding short ethics modules taught by subject matter experts [19], inviting guest speakers to present ethics content alongside technical content [38], and modifying assignments to focus on ethical considerations, especially with reflections [14, 37]. Existing integrations provide valuable space for students to consider their roles and decisions as individuals, but, as Vakil notes [42], instructors also need to make space for students to adopt a critical focus on the structures and systems that they exist within. Approaches without a structural focus also risk centering dominant narratives that utilize individualism as a tool to silence, supplant, and distort the narratives of marginalized groups and individuals [39].

As a vehicle for structurally focused embeddings, counternarratives look to give epistemological weight and theoretical grounding to narratives that run counter to existing dominant narratives [39]. As an example, one might consider the argument advocating for ethics integrations into computing curricula as a counternarrative, one that runs counter to dominant narratives that frame computing as objective, apolitical, and unbiased, with little need for ethics education [28]. Pedagogically, counternarratives have been used in K–12 and teacher education contexts to surface the structural nature of individual problems, examine underlying motivations and factors that have lead to these problems, and support students in social interventions to address problems structurally [30].

Prior ethics embeddings have aligned with counternarrative pedagogies; any embedding that seeks to center ethics implicitly employs counternarrative methods, countering dominant CS narratives that frame ethics as "off-topic". However, while students that learn counternarratives implicitly may be well positioned to change their own behavior or incorporate ethics into their decision making, they might be unprepared to name or advocate against dominant narratives within computing. One's ethical decision-making does not exist in a vacuum, rather it tends to be situated in a broader context of one's work, one's positionality, and one's relationship with structures of power and oppression. Following counternarrative scholarship [30], this work looks not only to educate students in naming dominant narratives, but also to resist and advocate against them, thus an explicit approach is necessary.

Given this explicit approach, many existing ethics embeddings could be modified to center counternarratives. For instance, embeddings in artificial intelligence (AI) could draw from counternarratives that highlight the racist behavior of so-called "race-neutral" technologies [4]. Embeddings in Human-Computer Interaction (HCI) might draw from counternarratives detailing manipulative interaction design [47], disability studies [26, 40], or diverse genders in research [23]. However, while prior work has established a growing list of potential embeddings, courses that center computer systems remain notably absent.

Unlike AI and HCI, computer systems are an explicit requirement of many computing degrees [1]. Additionally, an introductory course in computer systems that focuses on interactions between architectural systems design and software abstraction might be the lowest level of the computational stack around which students develop proficiency. Given the importance of computer systems in CS degree programs, the lack of ethics embeddings within computer systems, and the first author's prior experience researching computer systems and counternarratives, we felt that explicitly centering counternarratives into a required, introductory systems course could create a foothold for multiple avenues of future work.

In this work, we look to take an explicit approach towards integrating counternarratives into an introductory computer systems course. We describe our process for framing and presenting counternarratives, establish a counternarrative assessment, and report student and instructor reflections on the course integration.

2 COURSE DESIGN

We embedded ethics into an introduction to low-level software (computer systems) course that emphasizes the architectural interface between hardware and software, available for students to take after their introductory programming sequence, and completed at a variety of different points along students' degree timelines.

The course takes "a programmer's perspective" [8], though, unlike the Carnegie Mellon course that ours is based on [7, 35], this course is intended to be students' first exposure to the C programming language. As C's many historic quirks tend to lead to an unsafe and challenging programming experience for newcomers [2], our course aims to place a minimal programming burden upon students by modifying labs from *Computer Systems: A Programmer's Perspective* [8] to make them more accessible to students without a strong C background. Lectures in summer course offerings have an additional 10 minutes, which, along with a slight increase to the pace of lecture and the removal of a few specialized but inessential topics, left enough time to accommodate counternarratives.

2.1 Counternarratives in Systems Education

Our integration of counternarratives involved three changes: (1) an overarching course metaphor, the *House of Computing*, (2) relevant socio-technical content and counternarratives presented alongside technical material in most lectures, and (3) assessments of counternarrative understanding, *floorplans*. We discuss each below.

2.1.1 The House of Computing. Effective pedagogical use of counternarratives requires establishing links between individual problems in society and their structural manifestations. Often, the connection between individuals and their surrounding structures needs to be made explicitly visible; dominant narratives tend to conceal the existence of structures, especially those that seek to oppress, under the guise of objectivity [30]. For instance, dominant narratives surrounding work and poverty in the United States tend to emphasize self-reliance and personal responsibility, with little space given to the myriad of structural factors (e.g., 50 years of wage stagnation) that affect one's reliance on social services.

We expected some students to come to this course with an understanding of broad structural oppression, perhaps along axes of gender, race, and class, but we anticipated that most would lack an understanding of structural oppression within computing, as prior courses primarily taught dominant, technical narratives. Dominant computing culture emphasizes values inherited from industrial society such as efficiency, automation, and individualism over alternative priorities such as inclusion and justice, and is frequently presented without question within computing education. In addition to students lacking exposure to counternarratives, we were concerned that students who had completed much of their coursework might have already internalized these dominant narratives, leaving little space for any discussion of counternarratives. Thus, we looked to both motivate the inclusion of counternarratives as well as establish individual-structural connections.

Countering narratives that frame computing as objective, apolitical, unbiased and valueless, we utilized the *House of Computing (HoC)* to frame computing as a structural object warranting critique. Our syllabus began with a metaphor:

> *Let's imagine that computing is a house, maybe one that your parents lived in, maybe one that your grandparents lived in as well. This house was built quite a long time ago, somewhere in the 1940s, and has been lived in by many, many people since. There's a foundation that's been built a few times, there's many, many floors, there's lots of furniture and decoration, some that's stayed around since the house was built.*

The *HoC* represents the discipline of computing, a structural edifice built slowly over time through the labor of individuals and groups. We emphasized that the *HoC*, like many other older houses, has a variety of structural features that were created by people who, at the time, thought that their addition would be an improvement. Today, some features have aged beautifully, some were trendy and fell out of style, some have become dingy, and others are unequivocally unsafe, especially for those with greater accessibility needs.

Each floor within the *HoC* builds on the foundation of the floors below it, much like how abstractions within computing machines build on each other. We divided our course into three units — Data Representation, Programs, and Scale & Coherence — and treated each as a floor within the house. Within our metaphor, many floors exist beyond the ones explored in our course: lower floors might house spaces for computer architecture, digital logic, and transistor design, while higher floors might house more familiar high-level languages like Java or Python. By framing course material as explicitly structural and learning as an exploratory process through this structure, we hoped to frame this course as an opportunity to question structural assumptions and pose problems, rather than implicitly accept the validity of existing structures.

In addition to establishing connections between individual and structural problems, pedagogical approaches that incorporate counternarratives should emphasize the malleability of structures [30]. The *HoC* has features that need fixing or remodeling, but structural repair requires more finesse than simple demolition. We argued that, for students seeking to remodel, it's critically important to understand the existing structure and motivations for creating features that might now be considered obsolete. Without understanding the existing structure from a socio-technical perspective, remodeling projects might unintentionally destroy load-bearing walls and cause widespread collapse. For students that don't intend to remodel, we emphasized that the *HoC* remains inaccessible from many decades-old design choices, and that we should strive for a *HoC* that all students could access and feel a sense of belonging.

2.1.2 Socio-Technical Content. Prior work recognizes that students are shaped by their perception of professional practice [14, 41], so we sought to give counternarratives sufficient weight, relative to existing technical content, by integrating them into as many lectures as possible. Aiming for counternarratives with a connection to existing technical content, we examined established course topics through our *HoC* metaphor, drawing upon our existing knowledge of counternarratives within computing, and researching additional content as needed. Rather than describe this inclusion to students as counternarratives, we framed additions as socio-technical content that linked technical structures and social underpinnings.

Table 1 details the counternarratives that we chose alongside corresponding technical topics. For each course topic, we include the link between topic and counternarrative that we established when constructing this course. For instance, arrays in C famously lack bounds-checking, and dominant narratives tend to put the onus on the individual to remember to use library functions that dictate strict bounds checking. This individualistic framing matches dominant narratives surrounding accessibility and one's ability to use technology without causing harm, thus we drew from established work in disability studies that posed accessibility issues as structural problems rather than individual failings [32].

Our course met for lecture three times per week for 8 weeks, with the last 15 minutes of nearly every lecture devoted to socio-technical content and counternarratives, with three slots devoted to in-class critique on students' floorplans. Counternarratives were presented as lecture, supplemented with small group discussions and pre-lecture readings when appropriate. We chose to place counternarratives last in lecture so that we could present dominant narratives for contrast before delving into socio-technical topics, though we often referred to socio-technical content throughout. This approach allowed us to establish technical legitimacy among more technically-minded students, ensured that technical material for the course was adequately covered, and assured students that this course offering would be as technically focused as prior offerings. We considered counternarrative-exclusive lectures, but we were concerned that technically-minded students might skip, whereas our structure might force some degree of engagement.

2.1.3 Floor-plans. Prior offerings of this course during the COVID-19 pandemic had opted to replace exams with Unit Summaries, in part due to the infeasibility of administering exams remotely. The intent of Unit Summaries was to give space for students to create a personal artifact while engaging in the reviewing and summarizing that would typically occur with exam review. For our offering, we looked to assess students' understanding of counternarratives and opted to remain within the *HoC* metaphor by assigning *floorplans* in lieu of Unit Summaries or exams.

For each unit, or "floor" within the *HoC*, we asked students to create a representation of the spaces that the course visited throughout the unit. Our definition of a floorplan was intentionally vague: students could submit schematics, sketches, narratives, or other formats that felt accessible and expressive to them. Floorplans needed to include representations of socio-technical content; we argued that those with career success within the *HoC* (for instance, programming language designers) often have deep understandings of both social concerns and technical concerns. We also emphasized that established scientists and engineers go beyond descriptions of "what is" by supplementing with context when appropriate. Likewise, floorplans should include descriptions of why features were included within the *HoC* ("what was"), as well as what students would change about this space ("what could be").

As prior work emphasizes the importance of creative expression as necessary for feeling that life is worth living [46] and resisting and acting against oppression [17, 21], we wanted to make sure that floorplans offered space for students' creativity. Thus, no examples

Course Topic	Bridge Concept	Counternarratives
Accessing Memory	Course Foundations	The first programmers & computers as people, and the racist, sexist motivations for automating devalued labor [4, 27]
Introduction to the C language, Pointer arithmetic	History of C	C as rugged, minimalistic & individualistic, mirroring the spirit of the frontier from 1970s (space) and 1870s (manifest destiny)
Signed & Unsigned integers	Historic Tabulation	Babbage's analytical engine automating away jobs, inspired by Gaspard De Prony and Adam Smith
Bitwise & Boolean Operators	The influence of Navya Logic on George Boole [6]	The duality of computing's intellectual insulation, both as isolating and as a refuge, especially with autistic people [24]
Floating Point	Shame from comparing floats for equality	The exclusionary role of knowledge policing, community legitimacy, and shame within CS, comparing "man cards" and "CS cards"
x86 Programming I	Contrasting CISC and RISC ISAs	Arguments have ideologically underpinnings, with a focus on 1980s CPU advertisements and motivations for RISC ISAs [33]
x86 Programming II	Processor Market Domination	The growth of monopolies, within, and beyond computing, is a result of neoliberalistic anti-trust policies [12]
x86 Procedures	—	Textbooks and course goals have ideological underpinnings and design goals that can be examined through critical reading
Arrays in C	Lack of array bounds-checking	C's inaccessibility when viewed from a structural lens of access and ability, rather than an individual one [32]
Buffer Overflows	Lack of array bounds-checking	Technology, specifically programming languages, can be inaccessible, expanding to race and technology and racist technologies [4]
Direct Mapped Caches	Performance motivations for caches	Metrics, especially efficiency, are an ideological choice that defines success and shapes structures as a result
Associative Caches and Locality	Assuming code with "Good Locality"	Optimizing for the average case can be problematic by erasing diversity, drawing from critiques of machine learning [24]
Optimizing Code for Caches	Objectivity in CS	Positivist epistemologies and objectivity claims can cause harm when computing and science interact with human diversity
System Control Flow & Processes	Historic Operating Systems	The first operating systems as people, whose jobs were automated
Virtual Memory	Computing at a Global Scale	"Utopian" societal visions, especially from tech leaders, emphasize an all-encompassing scale and a technocratic, oppressive society [47]
Memory Bugs	Debugging challenges	Debugging is often disembodied and intellectualized, are there alternatives?
Java and C	—	Students' career practice within elite institutions centers prestige [5]
Course Wrap-Up	Finality	How can we act to achieve alternative futures? [17, 30]

Table 1: Connections between course topics and counternarratives, ordered by their appearance in our course.

of existing floorplans were provided to students; we felt that examples might short-circuit students' creative process. Students also submitted reflections on learning the unit's content. We evaluated students along four axes: (1) how completely the floorplan represented the unit's content, though students could justify omissions, (2) the cohesiveness of the floorplan representation relative to the unit's content, emphasizing the importance of prototyping and iteration when designing a floorplan, (3) the clarity of the floorplan and the metaphors that a student chose, and (4) the degree that students were creative, incorporating their own experiences and creating a piece that was unique and personal to them. For each, we evaluated using a 3 point standards-based grading scale [29], averaging and rounding up to produce a grade.

3 EXPERIENCES AND REFLECTIONS

3.1 Student Experiences

Most interesting to us was students' metacognitive awareness towards structural problems [30], especially among students that had not previously connected CS to widespread structural inequity or were unaware of structural inequity at all. In analyzing students' floorplan reflections and mid-quarter surveys, we found that nearly every student expressed either some existing metacognitive awareness situated within CS, or some newfound awareness that resulted from our intervention. One in particular realized that:

> *A computer scientist is not always an objective individual working with universal principals, which one might gather from the word 'scientist'. Rather, computer scientists must be mindful of their role in historical systems [. . .] and view their work through an informed lens.*

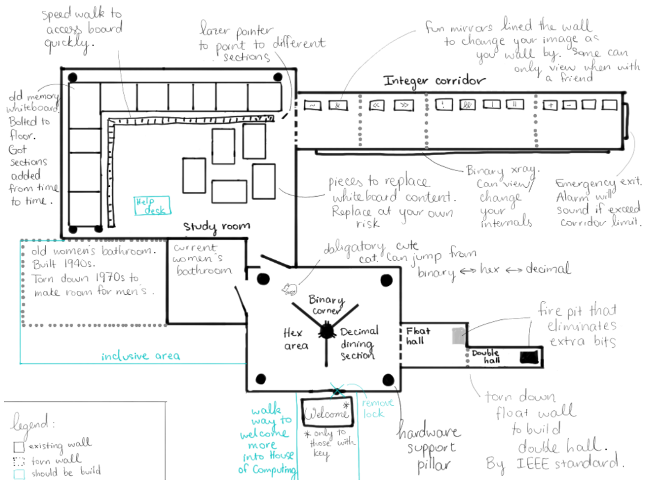

Figure 1: One student's floorplan for Unit 1

Several students expressed a change in how they viewed themselves and CS. One said that as a *"CS-minded person who believes efficiency more than anything, this unit alters my mind"*, the awareness of *"cultural values which implicitly affect how creators make certain choices"* was *"priceless; it changes my perspective on viewing many things"*. Another found:

> *'Priorities are baked in'. Even as a self-proclaimed skeptic, I have seen this ring true.*

Some students were less enthusiastic, but noted some degree of change:

> *I don't think the socio-technical content has really changed my life in big ways just yet, but I find myself thinking about what I've learned throughout the day.*

However, a few students didn't display this awareness. One reflection viewed historic counternarratives and present-day computing as separate:

> *I've gained a better understanding of what computer science used to be, compared to how it is now.*

Another ignored socio-technical content when responding to a scaffolding prompt that asked if their idea of what it means to be a computer scientist had changed:

> *Disagree, I don't think we've learned much about computing yet, but this unit did get me curious about how computers work at the fundamental level.*

In their mid-quarter feedback, a few students wanted to keep socio-technical content, but worried about their ability to keep up with course technical content:

> *The socio-technical content is really interesting but it does kind of take time away from what a really difficult set of technical content. I'd keep it, but it also means making sure the technical content isn't rushed.*

Students offered several options: one wanted socio-technical content to be an optional recording alongside lecture, another suggested devoting one lecture per week to socio-technical content:

> *I find the constant flipping between the two rather jarring during lectures sometimes as it might often detract from the new content that I am trying to digest or follow the thoughts forming in my head.*

Other students advocated eliminating socio-technical content:

> *Everything I'm learning in this course is excellent, but the socio-technical content is boring [and] unnecessary.*

While floorplans are by no means a perfect assessment of students' socio-technical understanding, they acted as an artifact representing students' relationships with course material. Looking to assess students' experience with the course redesign and focusing on students' first floorplan, the first author graded floorplans along the rubric presented in Section 2.1.3, then performed a second round of analysis, grouping for themes independent of the rubric. Two students submitted floorplans following requirements from a prior course offering, so we omitted them from our analysis.

At a high level, student floorplans either utilized metaphors to connect socio-technical and technical material, or they avoided using metaphors. For students that utilized metaphors (25 of 32 students), their floorplans varied around how well-suited their representations matched the course material (cohesion) and the uniqueness of their expressions (creativity). Students with strong cohesive and creative representations (12 total) chose a variety of forms: narratives that described walking through a house, pamphlets for gentlemen's clubs, pop-up rooms, and a variety of visual forms. Most chose to stay within a conventional "floor", connecting, for instance, operators that transformed binary representations to kitchen knives that "transformed" ingredients into different forms, and the English-centric ASCII encoding to multiple residents that struggled to communicate across different bit representations. Students that met expectations for creativity and cohesion (5 total) surfaced metaphors that mostly fit, but likely would've benefited from iteration and feedback, for instance, a multi-course meal where utensils were pointers. Other students (8 total) had metaphors that seemed somewhat haphazard, with strained connections to course material or other metaphors, for instance, conflating variable assignment to bringing in new items from outside and uniquely naming them.

Students that chose to not utilize metaphors (7 total) created floorplans analogous to a typical unit summary. Of these students, 3 created mind maps and 4 summarized the material linearly, with one student submitting a linear summary alongside personal reflections from posing socio-technical material as critical questions. We find it worth noting that the only students that didn't demonstrate awareness of structural problems were those that summarized the unit material linearly, without personal reflections (3).

Several students preferred floorplans to traditional exams:

> *Brainstorming and designing the project was extremely refreshing as opposed to the traditional review for exam.*

However, several students struggled with the unfamiliar format of an open-ended, creative assignment. In their mid-quarter feedback, one student offered:

> *The floor-plan part of the first unit summary was my biggest nightmare come to life. I still enjoyed it, don't get me wrong, but it was a lot tougher to focus on.*

Others were concerned about the correctness of their solution:

> *It took me quite a long time as I was stuck for a while trying to figure out what to do for the project.*

Some students procrastinated, due to the unfamiliar format:

> *I was extremely intimidated at the vague instructions for the floorplan and ended up putting it off.*

For some students, the vagueness proved beneficial:

> *I realized that innovation means working with little to no instruction and relying on knowledge and creativity to create a final product.*

While for others, it was a waste:

> *I wasn't sure what to do, and I spent a lot of time doing a thing that did not help me learn.*

3.2 Instructor Reflections

When designing this integration, our goal was a somewhat extreme embedding, integrating counternarratives within every lecture, with some spanning several lectures. We might have taken a more restrained approach by incorporating curricular perspectives and integrating counternarratives that fit best within our course, leaving the rest for future courses (though we argue that some especially crucial counternarratives should span multiple courses). While our approach was ambitious and should provide scaffolding for future instructors, student concerns about lecture pacing indicate that more cuts to course content were needed. As suggested, an approach that devoted one lecture per week to counternarratives might offer more reasonable pacing, but we worry that students may devalue or skip socio-technical lectures. We note that this was the first author's first time instructing a course, and more experienced instructors might find a better balance.

Throughout the course, we stressed that we were willing to meet with students who questioned the validity of presented counternarratives or the legitimacy of their adoption, but no students chose to engage in this capacity. We suspect that the enthusiasm for counternarratives that students with prior experience brought to the classroom bolstered the material's legitimacy and led students critical of our approach to perceive themselves as a minority. Lecture attendance dropped throughout the term; we wonder if students critical of counternarratives elected to not attend. For those that chose to attend, we noted that students seemed more comfortable with counternarratives, and themes surrounding dominant computing culture became more familiar. Some counternarrative content was more polished than others; for less-polished material students seemed less engaged but never combative.

Regarding floorplans, we recognized that an unfamiliar assignment format and open-ended rubric requirements might unsettle students. Our goal was an assignment that felt challenging, but accessible and supported. While we didn't provide example floorplans, several students found in-class critique to be especially valuable, both as an incentive to start their floorplan and an opportunity to integrate direction from others. However, some students still rushed to piece together their floorplan, and perhaps attaching a superficial grade to students' check-in would have aided motivation.

Students varied widely in their comfort with creative expression, and there were many additional opportunities to add scaffolding for students with less prior experience. In general, we graded floorplans leniently in an attempt to alleviate students' concerns around grades given an unfamiliar assignment and a unique socio-technical context, but these concerns persisted for several students. We could have offered a framework for approaching creative work, as well as more explicit instruction in iterative design. We also could have further emphasized that non-visual formats, especially narratives, were welcome representations for floorplans, as several students expressed concern around the additional cognitive load of learning a new expressive medium on top of an already challenging assignment. More iteration on our part is needed to balance assessment feedback with student anxieties around a new assignment.

4 FUTURE WORK

For instructors looking to embed counternarratives within CS classrooms, we offer a few parting words of advice. First, we found that some of the most expressive and powerful counternarratives were personal. Every instructor, even those with primarily dominant identities, experiences oppression at the hands of dominant narratives and structures. In our case, body scanners within airport security assume binary, cisgendered bodies and algorithmically optimize for efficiency based on that assumption, at the expense of non-normative bodies being labeled as anomalous. The first author on this work, who identifies as transgender, incorporated their airport experience into a counternarrative around common case optimizations within CS. Many complex experiences exist, but for us, a close connection to one's history and one's experience was a compelling guide towards constructing counternarratives.

Second, the task of choosing counternarratives to incorporate is a creative and challenging one, requiring mental and emotional space, and ideally would be completed before the course began. At the onset of our course, some counternarratives felt familiar, so constructing a framing for students was relatively simple. Others felt less so, and constructing a framing required several hours of research. For instructors looking to avoid overburdening themselves, choosing several counternarratives and noting the accessibility of each before the course began would be a splendid start. This would also allow cuts to existing technical material to be more conscious than ours, hopefully allowing for a more coherent course plan.

Finally, we feel that many spaces within CS would benefit from an embedded counternarratives approach. Much of computer systems and CS broadly involves teaching about concrete, human-made, historic structures alongside their modern-day counterparts. Few would argue the infallibility of these structures, and so a holistic approach could frame those structures as present but not permanent. Teaching these structures could be a praxis between constructing and critiquing these structures, perhaps utilizing the metaphoric framing of the *House of Computing*. This framing could make space for students to not only be critical of technical structures, but also socio-technical structures and oppressive structures, especially when instructors are willing to co-create that context with students.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 2031265, 1836813, 1703304, and 1539179, as well as support from Google, Microsoft, and Adobe.

REFERENCES

[1] Accreditation Board for Engineering and Technology Inc. 2016. *Criteria for Accrediting Computing Programs*. Technical Report. Retrieved July 31, 2021 from https://www.abet.org/accreditation/accreditation-criteria/criteria-for-accrediting-computing-programs-2016-2017/

[2] Rob Arnold. 2010. *C0, an Imperative Programming Language for Novice Computer Scientists*. Ph.D. Dissertation. Carnegie Mellon University.

[3] Jo Bates, David Cameron, Alessandro Checco, Paul Clough, Frank Hopfgartner, Suvodeep Mazumdar, Laura Sbaffi, Peter Stordy, and Antonio de la Vega de León. 2020. Integrating FATE/Critical Data Studies into Data Science Curricula: Where Are We Going and How Do We Get There?. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. Association for Computing Machinery, Barcelona, Spain, 425–435. https://doi.org/10.1145/3351095.3372832

[4] Ruha Benjamin. 2019. *Race after Technology: Abolitionist Tools for the New Jim Code*. Polity, Medford, MA.

[5] Amy J. Binder, Daniel B. Davis, and Nick Bloom. 2016. Career Funneling: How Elite Students Learn to Define and Desire "Prestigious" Jobs. *Sociology of Education* 89, 1 (Jan. 2016), 20–39. https://doi.org/10.1177/0038040715610883

[6] Mary Everest Boole. 1909. Indian Thought and Western Science in the Nineteenth Century.

[7] Randal E Bryant and David R O'Hallaron. 2001. Introducing Computer Systems from a Programmer's Perspective. In *Proceedings of the 32nd ACM Technical Symposium on Computer Science Education*. ACM, New York, NY, USA., Charlotte NC USA, 90–94. https://doi.org/10.1145/364447.364549

[8] Randal E. Bryant and David R. O'Hallaron. 2016. *Computer Systems: A Programmer's Perspective* (3rd edition ed.). Pearson.

[9] Carole Cadwalladr and Emma Graham-Harrison. 2018. Revealed: 50 Million Facebook Profiles Harvested for Cambridge Analytica in Major Data Breach. Retrieved August 10, 2021 from http://www.theguardian.com/news/2018/mar/17/cambridge-analytica-facebook-influence-us-election

[10] Erin A. Cech. 2014. Culture of Disengagement in Engineering Education? *Science, Technology, & Human Values* 39, 1 (Jan. 2014), 42–72. https://doi.org/10.1177/0162243913504305

[11] Sasha Costanza-Chock. 2020. *Design Justice: Community-Led Practices to Build the Worlds We Need*. The MIT Press, Cambridge, MA.

[12] Cory Doctorow. 2020. Early-Onset Oppenheimers. https://www.youtube.com/watch?v=Ep78A-jtcrE

[13] Virginia Eubanks. 2017. *Automating Inequality: How High-Tech Tools Profile, Police, and Punish the Poor* (first ed.). St. Martin's Press, New York, NY.

[14] Casey Fiesler, Mikhaila Friske, Natalie Garrett, Felix Muzny, Jessie J. Smith, and Jason Zietz. 2021. Integrating Ethics into Introductory Programming Classes. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*. ACM, Virtual Event USA, 1027–1033. https://doi.org/10.1145/3408877.3432510

[15] Casey Fiesler, Natalie Garrett, and Nathan Beard. 2020. What Do We Teach When We Teach Tech Ethics?: A Syllabi Analysis. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. ACM, Portland OR USA, 289–295. https://doi.org/10.1145/3328778.3366825

[16] Center for Combating Digital Hate. 2021. *The Disinformation Dozen*. Technical Report. Retrieved August 10, 2021 from https://s3.documentcloud.org/documents/21011322/disinfo-dozen.pdf

[17] Paulo Freire, Donaldo P. Macedo, and Ira Shor. 2018. *Pedagogy of the Oppressed* (50th anniversary ed.). Bloomsbury Academic, New York.

[18] Natalie Garrett, Nathan Beard, and Casey Fiesler. 2020. More Than "If Time Allows": The Role of Ethics in AI Education. In *Proceedings of the ACM Conference on AI, Ethics, and Society*. ACM, New York NY USA, 272–278. https://doi.org/10.1145/3375627.3375868

[19] Barbara J. Grosz, David Gray Grant, Kate Vredenburgh, Jeff Behrends, Lily Hu, Alison Simmons, and Jim Waldo. 2019. Embedded EthiCS: Integrating Ethics across CS Education. *Commun. ACM* 62, 8 (July 2019), 54–61. https://doi.org/10.1145/3330794

[20] Anna Lauren Hoffmann and Katherine Alejandra Cross. 2021. *Teaching Data Ethics: Foundations and Possibilities from Engineering and Computer Science Ethics Education*. Technical Report. University of Washington ResearchWorks Archive. http://hdl.handle.net/1773/46921

[21] Dorothy C. Holland (Ed.). 1998. *Identity and Agency in Cultural Worlds*. Harvard University Press, Cambridge, Mass.

[22] Lauren K. Gurley. 2020. Amazon Reportedly Has Pinkerton Agents Surveil Workers Who Try To Form Unions. *NPR* (Nov. 2020). Retrieved August 10, 2021 from https://www.npr.org/2020/11/30/940196997/amazon-reportedly-has-pinkerton-agents-surveil-workers-who-try-to-form-unions

[23] Os Keyes, Chandler May, and Annabelle Carrell. 2021. You Keep Using That Word: Ways of Thinking about Gender in Computing Research. *Proceedings of the ACM Conference on Human Factors in Computing Systems* 5 (April 2021), 1–23. https://doi.org/10.1145/3449113

[24] Amy J. Ko. 2021. Deconstructing CS Culture. https://www.youtube.com/watch?v=c8TQ29I8lK4

[25] Amy J. Ko, Alannah Oleson, Mara Kirdani-Ryan, Yim Register, Benjamin Xie, Mina Tari, Matthew Davidson, Stefania Druga, and Dastyni Loksa. 2020. It Is Time for More Critical CS Education. *Commun. ACM* 63, 11 (Oct. 2020), 31–33. https://doi.org/10.1145/3424000

[26] Jennifer Mankoff, Gillian R. Hayes, and Devva Kasnitz. 2010. Disability Studies as a Source of Critical Inquiry for the Field of Assistive Technology. In *Proceedings of the 12th International ACM Conference on Computers and Accessibility*. ACM Press, Orlando, Florida, USA, 3. https://doi.org/10.1145/1878803.1878807

[27] Jane Margolis and Allan Fisher. 2002. *Unlocking the Clubhouse: Women in Computing*. The MIT Press, Cambridge, Massachusetts.

[28] C. Dianne Martin. 1997. The Case for Integrating Ethical and Social Impact into the Computer Science Curriculum. In *The Supplemental Proceedings of the Conference on Integrating Technology into Computer Science Education: Working Group Reports and Supplemental Proceedings*. ACM Press, Uppsala, Sweden, 114–120. https://doi.org/10.1145/266057.266131

[29] Robert J. Marzano. 2010. *Formative Assessment & Standards-Based Grading*. Marzano Research Laboratory, Bloomington, IN.

[30] Richard Miller, Katrina Liu, and Arnetha F. Ball. 2020. Critical Counter-Narrative as Transformative Methodology for Educational Equity. *Review of Research in Education* 44, 1 (March 2020), 269–300. https://doi.org/10.3102/0091732X20908501

[31] Christy Moore, Hillary Hart, D'Arcy Randall, and Steven P. Nichols. 2006. PRiME: Integrating Professional Responsibility into the Engineering Curriculum. *Science and Engineering Ethics* 12, 2 (June 2006), 273–289. https://doi.org/10.1007/s11948-006-0027-6

[32] Mike Oliver. 1990. The Individual and Social Models of Disability. *Joint Workshop of the Living Options Group and the Research Unit of the Royal College of Physicians* (1990), 7.

[33] David A. Patterson and David R. Ditzel. 1980. The Case for the Reduced Instruction Set Computer. *ACM SIGARCH Computer Architecture News* 8, 6 (Oct. 1980), 25–33. https://doi.org/10.1145/641914.641917

[34] Evan Peck. 2019. The Ethical Engine: Integrating Ethical Design into Intro to Computer Science. https://medium.com/bucknell-hci/the-ethical-engine-integrating-ethical-design-into-intro-to-computer-science-4f9874e756af

[35] Brian P. Railing and Randal E. Bryant. 2018. Implementing Malloc: Students and Systems Programming. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*. ACM, Baltimore Maryland USA, 104–109. https://doi.org/10.1145/3159450.3159597

[36] Michael Sainato. 2020. 'I'm Not a Robot': Amazon Workers Condemn Unsafe, Grueling Conditions at Warehouse. Retrieved August 10, 2021 from http://www.theguardian.com/technology/2020/feb/05/amazon-workers-protest-unsafe-grueling-conditions-warehouse

[37] Jeffrey Saltz, Michael Skirpan, Casey Fiesler, Micha Gorelick, Tom Yeh, Robert Heckman, Neil Dewar, and Nathan Beard. 2019. Integrating Ethics within Machine Learning Courses. *ACM Transactions on Computing Education* 19, 4 (Nov. 2019), 1–26. https://doi.org/10.1145/3341164

[38] Michael Skirpan, Nathan Beard, Srinjita Bhaduri, Casey Fiesler, and Tom Yeh. 2018. Ethics Education in Context: A Case Study of Novel Ethics Activities for the CS Classroom. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*. ACM, Baltimore Maryland USA, 940–945. https://doi.org/10.1145/3159450.3159573

[39] Daniel G. Solórzano and Tara J. Yosso. 2002. Critical Race Methodology: Counter-Storytelling as an Analytical Framework for Education Research. *Qualitative Inquiry* 8, 1 (Feb. 2002), 23–44. https://doi.org/10.1177/107780040200800103

[40] Katta Spiel, Kathrin Gerling, Cynthia L. Bennett, Emeline Brulé, Rua M. Williams, Jennifer Rode, and Jennifer Mankoff. 2020. Nothing About Us Without Us: Investigating the Role of Critical Disability Studies in HCI. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. ACM, Honolulu HI USA, 1–8. https://doi.org/10.1145/3334480.3375150

[41] Reed Stevens, Kevin O'Connor, Lari Garrison, Andrew Jocuns, and Daniel M Amos. 2008. Becoming an Engineer: Toward a Three Dimensional View of Engineering Learning. *Journal of Engineering Education* 97, 3 (2008), 355–368. https://doi.org/10.1002/j.2168-9830.2008.tb00984.x

[42] Sepehr Vakil. 2018. Ethics, Identity, and Political Vision: Toward a Justice-Centered Approach to Equity in Computer Science Education. *Harvard Educational Review* 88, 1 (2018).

[43] Sepehr Vakil and Rick Ayers. 2019. The Racial Politics of STEM Education in the USA: Interrogations and Explorations. *Race Ethnicity and Education* 22, 4 (July 2019), 449–458. https://doi.org/10.1080/13613324.2019.1592831

[44] Moshe Y. Vardi. 2018. Are We Having an Ethical Crisis in Computing? *Commun. ACM* 62, 1 (Dec. 2018). https://doi.org/10.1145/3292822

[45] Daisuke Wakabayashi, Erin Griffith, Amie Tsang, and Kate Conger. 2018. Google Walkout: Employees Stage Protest Over Handling of Sexual Harassment. *The New York Times* (Nov. 2018). Retrieved August 10, 2021 from https://www.nytimes.com/2018/11/01/technology/google-walkout-sexual-harassment.html

[46] Donald W. Winnicott and Robert Rodman. 2010. *Playing and Reality* (reprint ed.). Routledge, London.

[47] Shoshana Zuboff. 2019. *The Age of Surveillance Capitalism: The Fight for a Human Future at the New Frontier of Power* (first ed.). PublicAffairs, New York.