Gaming Together, Coding Together: Collaborative Pathways to Computational Learning

Brianna Dym
brianna.dym@maine.edu
University of Maine
Orono, Maine, USA

Cole Rockwood
cole.rockwood@colorado.edu
University of Colorado, Boulder
Boulder, Colorado, USA

Casey Fiesler
casey.fiesler@colorado.edu
University of Colorado, Boulder
Boulder, Colorado, USA

ABSTRACT

Collaborative, playful learning represents an important avenue to mastering a range of skills within computer science education. This research presents findings from interviews with 9 members of an online community that started out as a gaming league and transitioned into a game development team. Community members learned programming skills to contribute their ideas to the game and participate in activities based around game development. Drawing on these experiences, we identify key elements from informal learning that can improve computer science education: 1) playful projects can help learners overcome barriers to participating in computer science; 2) community closeness facilitates a collaborative learning environment to support developing expertise in specific computational skills. We consider these findings in the context of learning as an everyday social practice, and discuss means of developing playful learning communities in computer science classrooms.

CCS CONCEPTS

• **Social and professional topics → Informal education**; **Computer science education**.

KEYWORDS

Computing Education, Video Games, Communities of Practice, Informal Learning

ACM Reference Format:
Brianna Dym, Cole Rockwood, and Casey Fiesler. 2023. Gaming Together, Coding Together: Collaborative Pathways to Computational Learning. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2023), March 15–18, 2023, Toronto, ON, Canada.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3545945.3569833

1 INTRODUCTION

Collaborative learning represents an increasingly adopted educational tool in computer science classes [27]. Additionally, online communities outside of the computer science classroom offer opportunities for learners to accomplish impressive computational feats [11, 31]. Not only do people in these communities commit to learning complex skills, they do so without the support of formal education. These informal and collaborative learning experiences can impart confidence in technical subjects that might otherwise feel too intimidating to a learner. This research is part of a broader body of work that seeks to draw lessons for facilitating learning in the computer science classroom from in-the-wild collaborative computing spaces.

In advancing that work, our research draws insights from a specific case study. We conducted interviews with a community of fan game developers that taught themselves all the skills necessary to build a video game from the ground up. These nine developers transitioned from a community playing video games together to building their own video games. Participants were motivated to build games by playing together, both in terms of gaming and other communal activities like role-playing and sharing comics and stories. What started as one person tinkering with some game assets expanded to a whole community of learners.

This article presents findings on informal computational learning through play in online communities, then discusses how to potentially design playful and collaborative course work in computer science education spaces inspired by this example. The lessons drawn from this community demonstrate how classrooms can leverage collaboration, expertise, and creativity to support students from underrepresented and nontraditional backgrounds in becoming more confident in computer science overall.

2 BACKGROUND

Informal learning represents a rich site of inquiry for improving computer science education. Prior work includes examining informal spaces including Scratch communities [14, 43], hackathon spaces [21, 31], game jams [29, 35], or even online communities that don't center computational work. Additionally, transformative fandom, the community surrounding media fans and the transformative works they create based on that media, represents an online community where people have learned a range of computational skills to participate in community activities [7, 11]. Organized events like hackathons and game jams also teach valuable collaboration skills [31, 35]. In particular, Nandi and Mandernach noted the benefits of learning interpersonal and collaborative skills with a team of programmers as a major outcome of participating in hackathons [31].

Mizuko Ito argues for the importance of networked informal learning spaces [18], particularly "affinity spaces" as described by Gee [13]. Ito describes connection and shared interest being an important motivation for learning, especially in the context of marginalized communities such as LGBTQIA+ people and women of color, who may not feel comfortable forming social connections

SIGCSE 2023, March 15–18, 2023, Toronto, ON, Canada.

ACM ISBN 978-1-4503-9431-4/23/03…$15.00
https://doi.org/10.1145/3545945.3569833

in offline learning spaces. Informal learning spaces such as online communities offer a space for young people to connect on their terms and through their own interests, Ito argues, and should be considered important spaces to integrate into formal education practice.

Learning theorists Jean Lave and Etienne Wenger-Trayner highlight the importance of learning from peers in their discussion of communities of practice [26], where observing and interacting with a community facilitates learning a skill. Both hackathons and informal online developer communities can be viewed as communities of practice. Within transformative fandom, a number of community members learned programming skills in order to help build the fanfiction platform Archive of Our Own, and prior work has identified legitimate peripheral participation within their learning processes [11, 26]. That is, volunteers were given meaningful tasks on the *periphery* of larger programming tasks. Volunteers started as testers or as tag wranglers (e.g. cleaning data), then would move on to tasks like debugging or assisting in smaller development tasks. Eventually, many of those volunteers became fully competent programmers [11].

In this study, participants made fan games, which is an independent, standalone creation that takes inspiration from an original video game. Fan games are made using a mix of borrowed or duplicated assets from an original game paired with original code and assets developed by a fan community from the ground up. They use the likeness, images, and mechanics of a popular original media property to design a new video game that both plays with and critiques the original media property. Because fan games remix original media concepts into something new, they are considered a type of fanwork [4].

Like other fanworks, fan games occupy a legally gray space in terms of copyright [22]. Hosting and distributing fan games can be challenging when there is no support for making derivative works from the media property owner. For this reason, fan games are typically made freely available to communities as part of a gift economy [15].

Much of the user-generated content in fandom relies on technical knowledge to some degree. Archive of Our Own, for example, is built and maintained almost entirely by women who initially had to teach one another the programming skills needed to maintain the platform [11]. The technical work carried out in fan communities is part of its everyday practices [7]. In describing how people come to identify as part of a community, Wenger-Trayner centers the means of practice as building identity [42]. That is, people carry out daily rituals and routines that form the practice of being or doing whatever they have committed to. Through iterating on those practices, the person learns how to be a member of their community as well as the social and technical skills required to participate. Within the context of a fan community, people carry out a range of everyday practices, both social and technical, to maintain the community.

Fan communities like the one studied here have previously been characterized as communities of practice thanks to the amount of expertise required to operate within a fan community [17]. The expertise required to participate in certain fan spaces can make those practices exclusionary, whether for better or worse [17]. In this study, we examine communal practices as a means of learning through near-peer connections [26]. Peer and collaborative instruction have shown to provide strong educational benefits to students in large-enrollment computer science classrooms [16]. However, collaborative learning environments also run the risk of students newer to computer science lacking the scaffolding needed to truly learn the material, and instead rely on more experienced group members to do the labor [3].

In the computer science classroom, video game development has been explored as a means of drawing in drawing in students from underrepresented groups. Researchers have leveraged video game testing to successfully teach computer science to underrepresented groups in computing [6]. Researchers have also successfully deployed a video game development project to engage students in learning specific computer science concepts [33]. Educators have even used game design as an interdisciplinary means for teaching computer science to a wide range of students [40]. Video games integrate boundaries and rules to transform unstructured play into a game [39]. Within the undergraduate classroom, video games can be used to increase student interest and engagement with introductory programming concepts [5, 24, 30].

Rather than investigating means of implementing game development in the context of formal computer science education, this work examines in-the-wild collaborative, peer-supported learning to draw potential lessons learned back to the classroom. In particular, our interviews surface mundane, routine practices within the community to understand everyday learning practices. In our analysis, we focus on two findings: (1) how play formed an important motivator for overcoming barriers to learning computer science; and (2) how community closeness facilitated a collaborative learning environment to support developing expertise in specific computational skills.

3 METHODS

This research is part of a broader investigation into fan communities as non-traditional spaces for learning computational skills. What we present here is a case study that draws on nine interviews with participants from the same community. The participants all belong to a close-knit fan game community where people create and share their own games and other fanworks related to a specific original media property. In the course of recruiting participants for a separate study about computational fan projects, we learned about P1's fan game. When we reached out to P1 for an interview, she agreed to speak with us. She then shared our call for participants for our IRB-approved interview study with everyone else who worked on fan games in her community. In total, we recruited nine participants, six of whom work on the main fan game the community is known for and three work on a different fan game hosted in the community's forums. The close-knit nature of participants all working on the same or adjacent projects makes for a useful case study on collaborative learning within informal computing spaces.

Out of the nine participants, three identified as women, five as men, and one as nonbinary. A majority of participants were white and live in the U.S. Five participants identified with the LGBTQIA+ community in some way. Participants were all in their 20s, ranging from 20 to 28. All the participants joined the fan game project without being experts in computer science. Furthermore, none had more than a semester of educational training in programming or computer science concepts. P2 had attempted to learn programming before starting work on the game, but had decided that it was not for her after experiencing an introductory course. Working on the game change her mind, and encouraged her to apply to computer science programs later. P4 had recently enrolled in a computer science program at the time of the interview, but found it frustrating compared to his experiences working on the fan game. P3 was working toward a PhD in mathematics, but his work with the game had come first. All participants identified working on fan games as their first successful engagement with programming, with no single participant beginning work on the game with significant programming experience. We describe their characteristics and experiences in aggregate here to preserve the confidentiality of their identities.

We conducted semi-structured interviews [34] that lasted between 30 and 90 minutes, with an average interview duration of 45 minutes. We communicated with participants over voice, video, or text chat depending on their preference. We recorded and transcribed audio from interviews or copied chat logs that were then modified to form a transcript. We then conducted a thematic analysis [1] on the transcripts. The first and second authors read through the transcripts for an initial round of open coding, then met and discussed emerging themes. After forming a code book of themes relevant to our research goal (methods for broadening participation in computing by leveraging transformative works), we then conducted a second round of coding for those themes specifically. The themes we identified in the initial round focused not necessarily on learning, but on the pull that this community had on its members. At this part of our analysis we also compiled narrative profiles of each participants' journey in getting involved with making fan games, summarizing from participant quotes.

From this analysis, we identified two emergent overarching themes connected to making games as play: (1) participants transitioned from other forms of play (e.g. role-play and drawing comics) to developing their own games as an extension of their play; and (2) participants used building fan games as a means of being close with their friends and community. These themes offer insight into how we as educators might facilitate stronger collaborative and playful learning experiences in the computer science classroom, especially for long-term small group work. In the following section, we share the experiences of our participants and how they relate to these themes.

3.1 Ethical Considerations

Because our participants were making fan games using the likeness, images, and mechanics of an original media property, their works fall into a legally gray area. Participants did not directly make money off their fan games and instead made them freely available for others to enjoy, just like other transformative works. However, participants still fear copyright claim notices from the original media property owners and maintain some level of secrecy around their community. Secrecy norms are a common means of protecting fanworks from legal repercussions [10]. To respect and preserve that secrecy, we have removed any mention of the original media property from participant data and do not disclose which media property their fan game borrows from.

4 FINDINGS

Our participants come from a community that started out as an online forum for fans of a popular video game franchise to organize online matches against one another and role-play as characters from their game. Role-playing a character in coordination with a community can be a useful tool for bridging games, play, and an educational subject [38]. Within the context of our interview participants, it was a creative catalyst that sparked the initial drive to delve into game development. While P1 first created the game, all other participants were recruited as "bug fixers" and developed basic programming skills through learning how to troubleshoot errors in the game code. Over time, participants specialized into unique roles, ranging from providing art, level design, tweaks to the game's combat, and more. Participants were drawn to the fan game as a creative work and were motivated to learn whatever skills needed as a means of being further involved in the community. Participants were not concerned with how they learned, but how their involvement in developing the game brought them closer to the overall community. These findings explore how the playful community participants were part of motivated their learning opportunities.

4.1 Learning Motivated Through Play

The forum was initially built by P1, who wanted to make a gathering place for friends to play online together. However, it was not until an inciting incident that the community began developing the fan game. One night, when the forum's moderators were offline, a troll infiltrated the community. While no harm came to anyone, the event left a lasting impression. A narrative for the role-play took shape around the troll's antics. Community members began illustrating comics summarizing the events, and those comics eventually inspired the fan game. The lead developer for the game, P1, says the game started as scraps of conversations and sketches between community members. The game was a narrative, a story told between community members that was later applied to a programming tool to express a creative vision. This process is similar to how creative writing and storytelling are leveraged as a means to introduce students to programming [2, 19, 20].

> *"And so later on when people were talking like, 'what would [our role-play world] look like if it were a [location in the main franchise]?' What would it look like if it were a region? And we naturally defaulted to the comic and story that users came up with where it was supposed to be an apocalyptic city. When I was drawing a detailed version of the region map, I had no intention to make a game out of it."* (P1).

The game started with P1 downloading RPG Maker (a beginner's game development tool) and a package of assets that replicated the look and feel of the main franchise. P1 started putting the game together on her own. Eventually other participants got involved as part of the development team. Most participants started out as members of P1's online forum because they were fans of the larger video game franchise. Later, they became fans of P1's fan game and would ask if there were ways they could support its development. P3, for example, had played the fan game as a teenager and wanted to be more involved in the development process.

> *"At the time, I had played the game to the point of completion. I enjoyed it very much, obviously...[And the development team was looking for help with] things like fixing small bugs that fell through the cracks. So they put out a call on the forum, looking for anyone that was interested. And I had no experience whatsoever. I learned Ruby, the game's programming language, on the job, getting help from the people that worked on it already and having to get familiar with Ruby from the code itself."* (P3).

Starting out as a complete novice, P3 received guidance from P1 and other community members, first learning how to find and resolve bugs in the game's code. P1 taught P3 how to debug problems with the game. P3 fell into a rhythm of working on bugs in his own time, then approaching P1 with questions whenever he got stuck. Combined with looking up further resources, talking with other developers, and applying his own expertise in mathematics, P3 became a formidable game developer (at the time of interviews, P3 was in graduate school for a mathematics program). Eventually, P3 learned enough to program the combat AI for the fan game, taking pseudocode statements from P1 and programming them into the game itself. The combat AI now consistently beats P3, who is a self-described casual player of the main franchise. P2, who at the time of interviews was enrolled in a graduate computer science program, talked about how working on the fan game was her first major introduction to computer science:

> *"In 2015 I was bored. I ended up searching for different fan games [to play]. And then I found [this fan game] and decided to play it...And then [P1] and I started talking and that was how I ended up getting involved in the community...When I first got started, I didn't know how to program in Ruby. And I didn't think that I was very good at programming in general. After I started working on optimizations, I ended up getting really involved and learned a ton of extra stuff about Ruby and programming in general."* (P2).

Like P3, P2 started out doing bug fixes to help P1 with updates to the game. While working on the game started out as a means to be more involved with a community, she also grew more confident in her skills as a programmer. She said, *"It's frankly saving me a lot this term"* in reference to her graduate computer science classes. Before working as a developer on the fan game, P2 had frustrating and discouraging experiences with formal computer science education. It was after trying out programming for the fan game that she decided to give computer science another try.

Participants 2-9 became developers on these fan games after first being members of P1's role-playing forum and playing her fan game. P7, for example, decided to try making his own fan game because he wanted to tell a story that felt best suited for a video game. After teaching himself how to use the development software that P1 uses (RPG Maker) and downloading the third-party modding kit for the main franchise, he had a working version of his own game and, more importantly, his own story. He posted the game files for download on P1's forums. P7 took a break from developing the game, but found a message from P9, who was interested in collaborating on the game, two years later.

> *"When I picked up the project again last year, it was mostly because one of the fans was really interested in working on [the fan game] and bringing it back. They just offered to do art first and I was like, 'Look, I don't really know what I want to do. I don't know if I'm going to start working on it again.' And then she went ahead and started doing all this artwork. So I was like, well, let's actually do it then."* (P7).

> *"The moment I got the response I immediately started working on [the fan game] because I was so excited. We spoke back and forth on the forums until P7 eventually got a discord. And we've been chatting on discord ever since in a group chat."* (P9).

After P9 got involved in the game, she asked P7 if a friend of hers, P8, could join the development team.

> *"I think my drive for getting involved was the fact that P9, who's my best friend, got involved in [P7's fan game]. And we were just joking around. She was showing me a lot of stuff she was working on and over time as she was showing me more of how her contribution was improving the game's quality. I wanted to get involved as well."* (P8).

For P8 and P9, working on video games was a passion they wanted to pursue professionally. Working on the fan game was a fun way to practice skills they wanted to apply toward an industry position, but were not necessarily learning about in their programs. While P8 and P9 volunteered because working in game design was their dream, other participants viewed the fan games as a fun, creative outlet, not unlike sitting down and writing fanfiction.

> *"[For school], you have to keep working on [your programming] even if you don't succeed. For [P1's fan game], I could step away for a few hours. I don't feel guilty about it. Like I'm not wasting time...Another big difference is that nobody's expecting anything from me. I was just doing it. Nobody knew I was doing it. So I had time to do the things I wanted to do and no pressure from anybody to do it."* (P4).

> *"[P1] taught all of us how the animation editor worked. Anything she didn't teach us, I would learn myself through trial and error. She was kind of hands off and let us do what we wanted–but of course would give us notes after showing her the animations."* (P5).

Making the games became the primary way that participants engage with their community. Most participants did not play the games anymore but talked about maybe playing through once they had finished building it. Instead, participants found the act of making the games themselves as a form of play and spending time with their friends. Learning was intrinsically motivated, with a few pointers coming from P1 as the project lead, but with most participants learning through their own curiosity and time spent playing with the game development tools. In the following section, we explore how fostering community closeness was an important part of participants' practice as learners.

4.2 Community Closeness Fostering Learning

Participants shared a sense of community closeness with the other developers. Some participants became friends while working on the fan games. Others got involved in working on the fan games because of their existing friendships. As discussed above, P1 began working on the fan game as an extension of her community's playful discussion. This playfulness extended to the fan game as a project, allowing P1 to recruit help over the years. After searching for fan games related to the main franchise, P2 played and enjoyed P1's game. After becoming more active in the community, she joined the development team.

Similarly, P3 joined the team after playing the fan game and wanting to get more involved in the community. Now, P1, P2, and P3 are all close friends and are moving on to work on an original game development project together. P6 started out as friends with P1 and then joined to help perform bug tests. Now, P6 also writes scripts for the game's animations.

> *"I don't work too closely with the team as a whole...I'm doing a lot of informal alpha testing and bug reporting but [P1] and I were already friends, and the team has a few other friends so bug reporting can always be a good time."* (P6).

In the interview, P6 shared a screenshot of the team joking about a bug someone found while testing the game. The team communicates regularly over Discord, sharing lighthearted moments in the development process. Participants worked on these fan games because they were social projects that helped them connect with friends and other members of the community. Their work is voluntary and is a means to contributing in a playful way, much like how other game modders have previously viewed their work [32]. Reflecting on the game's progress, P1 admitted that the community support was what made making the game possible:

> *"I feel very lucky to have attracted the community that we have. And I say this, you know, now it's 2020 and we've been here for, for so many years since the game has been out. But even back [when the fan game started], I feel like I've really lucked into having such a supportive community that has enabled this whole [project]."* (P1).

P5 described similar feelings of camaraderie, which contributed to the playful nature of making the game, *"It's so fun making something and having the team who I practically view as a family."* While fostering feelings of such closeness and support may not be achievable in the timeframe of a one-semester course, there are certain ways that this community functions as co-learners that offer important lessons for how we might structure learning in the computer science classroom. In the following section, we discuss motivating factors in learning to code and then outline guiding principles for designing in-class assignments that model the fan game community.

5 DISCUSSION

If we look at the common experiences across this community, we see that participants primarily learned to code over time (that is, over the course of several years) and through their own exploration. While P1 acted as the project lead and gave basic directions, she also gave participants space to figure out their own work processes. Participants were motivated to learn what they needed to know not because they wanted to learn to code, but because they wanted to contribute to a creative project. In the following sections, we discuss how learning occurred within this community and how we might replicate this style of learning in the computer science classroom.

5.1 Motivation and Learning in a Fan Community

Many creativity-focused computer science education environments such as EarSketch or Scratch are based in part on the importance of "personally meaningful" experiences for learners [8, 28]. EarSketch, in which learners use code to remix music, has been found to be engaging in part because music can be so culturally relevant and meaningful [8]. Scratch similarly is often leveraged to allow learners to choose to work on topics that are of most interest to them [37]. In the context of these types of creative computing environments, prior work has also pointed to the potential for leveraging fandom for interest-based motivators in computer science education [9]. Moreover, we already know from prior work that people in fan communities are often highly motivated to learn complex technical processes in order to contribute to projects they are excited about or to help their community [7]. Our findings here imply that both being a *fan* of something and being part of a community around it can be a powerful motivator for computational learning.

However, learning does not simply happen because a person is motivated. There are specific processes in place within a community that facilitate learning. According to Jean Lave, knowledge is constructed through cultural and social processes, and is also transformed through those processes (p. 32 [25]). During interviews, participants explained how they learned to contribute to the fan game through situated learning that was tied to specific social processes within the community [26].

That is, members of the community started by learning to first fix bugs on the game and would approach a peer with questions as they encountered more and more complex problems. Through the practice of fixing bugs, participants eventually broadened their practice to include other problems with the fan game, such as redesigning the game's combat AI, animating new combat moves, and constructing maps and dungeons for the player to move through.

The community learned the skills necessary to complete these tasks not because they wanted to learn a specific skill, but because they wanted to be involved in that community. Technical practices, too, are social practices with culturally produced knowledge. They do not live within the hardware of platforms but in the everyday lives of community members. P3 drew on the expertise of his fellow community members to initially learn the basics of game development, but soon after sought out further resources to bring his own ideas to fruition. P3 did not dwell on the process of learning to re-code the combat AI in his interview. Instead, he talked about how P1 introduced him to working on the game. It was that moment of mentorship and sense of community that stood out to P3.

Fostering a sense of community and peer mentorship can be challenging in a computer science classroom, especially when certain collaborative interactions don't always lead to equal learning opportunities [3]. However, the learning experiences described by our participants point to an opportunity in the computer science classroom. Long-term collaboration across the course of an entire semester, or even across multiple semesters, could help students foster a sense of community closeness. Collaborative assignments that center playfulness as opposed to subject matter expertise can help foster that community closeness within the classroom. With these suggestions in mind, we next discuss further how to apply these informal learning practices to the computer science classroom.

5.2 Computational Learning as Everyday Social Practice

Recent scholarship from Etienne Wenger-Trayner describes a new challenge to learning in the age of the internet [42]. With so many people getting online and generating content, accessing curriculum is hardly a barrier to learning. There are forums, YouTube tutorials, scanned textbooks, and active discussions from educators about best practices in the classroom. Wenger-Trayner argues, instead, that identity represents a challenge in learning. That is, who identifies as part of a community of learners, and how they relate to the practices in a community, impacts a person's ability to learn [42].

In computer science, stereotype threat plays a role in limiting how people identify as learners in the computer science community [23]. Learning is "the becoming of a person who inhabits the landscape" of a specific body of knowledge [42]. The act of inhabiting that landscape involves everyday practices that reinforce one's identity within an area of expertise.

Our participants centered their identity on being a fan of a game first. Over time, their identity changed to centering the community of people surrounding a fan game. They saw their computational practices as a pathway to being more involved in their community, as an opportunity to foster more community closeness. Working on fan games reinforced a sense of membership, thus familiarizing the individual with a group (p. 149, [41]). Identity became a negotiated experience for the fan community, as Wenger-Trayner posits [41], a lifestyle that is constructed through community experiences.

To model this style of learning in the classroom, we might consider designing large-scale collaborative projects. The teacher or a more advanced student has the opportunity to facilitate learning through a project manager role, much like the one that P1 adopted. Individual students ought to be given the opportunity to find an area of "expertise," a role where they understand how their specific contribution to the group project fits in.

In the fan game developer group, participants moved from fixing bugs to finding a niche expertise based on their own interests. For P3, it was working on the game's combat. For P6, it was art. P8 wanted to design maps for the game. P9 redesigned the game's sprites. By adopting an expertise, participants were able to focus their creative energy toward a specific task while learning the basics needed to work on the overall project.

In a classroom, this project might look like developing and "launching" an application, platform, or video game. Over the course of a semester, students can have certain benchmarks to meet at checkpoints, certain competencies they need to demonstrate through presenting different pieces of their work. If students are building a video game together, they might need to submit a storyboard, proposed interaction design, and plan for specific game features they want to show off. The next check-in could showcase one specific feature, with later check-ins adding features to the game over time. In this process, students would have a scaffolded project with small, achievable benchmarks that build up to a large showcase. Collaborative learning has already been demonstrated to be an extremely effective method of helping advanced CS students perform better in the classroom [27]. Additionally, research from Timmerman and colleagues has demonstrated exciting potential benefits for department-wide and multi-semester collaborations with a broader community for computer science students [36]. The findings in this research suggest how collaborative learning can benefit novice learners and draw in people who might not otherwise identify or fit in with perceptions of computer science.

Only three of the nine participants from this study were heterosexual white men. The majority of participants belonged to underrepresented groups in computing as women and/or as part of the LGBTQIA+ community, a group that is increasingly noted to withdraw from or avoid STEM disciplines [12]. Participants learned computing to be part of the community, not to become specialists in computer science. Drawing from this perspective, we might pursue blended courses that target students in different disciplines, especially from creative backgrounds. Even if students in these blended courses do not change their major to computer science, they still have the opportunity to learn computational skills and potentially change broader perceptions toward computer science.

6 LIMITATIONS AND CONCLUSION

At its core, computer science can be a very playful discipline that empowers people to express themselves in a unique medium. Our participant group in this study was small–only nine individuals. However, their experiences remind us that computer science education still struggles to replicate a sense of community closeness and meaningfulness within the classroom. In future work, we might explore how to better foster that sense of community and its impacts on learning. In the networked world we live in, people are able to access resources that tell them exactly how to write in the programming language they want to learn. What does this mean for the future of higher education, and why should a student pay to learn something in a classroom that they could instead observe online? As educators, we have the opportunity to not just curate curriculum but also facilitate a community of learners through playful and meaningful activities. Whether that be through making games, designing tools for play, or other creative projects, playful collaboration can inspire people to learn in the pursuit of being part of a community.

ACKNOWLEDGMENTS

Special thanks to the Internet Rules Lab for providing valuable feedback to early drafts of our paper, and a very special thank you to our participants for letting us into their community and sharing their stories. Funding for this work was provided by NSF award 1936741.

REFERENCES

[1] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative research in psychology* 3, 2 (2006), 77–101.

[2] Quinn Burke and Yasmin B Kafai. 2012. The writers' workshop for youth programmers: digital storytelling with scratch in middle school classrooms. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education*. 433–438.

[3] Leslie Cintron, Yunjeong Chang, James Cohoon, Luther Tychonievich, Brittany Halsey, Devon Yi, and Genevieve Schmitt. 2019. Exploring underrepresented student motivation and perceptions of collaborative learning-enhanced CS undergraduate introductory courses. In *2019 IEEE Frontiers in Education Conference (FIE)*. IEEE, 1–6.

[4] Francesca Coppa. 2014. Fuck yeah, fandom is beautiful. *The Journal of Fandom Studies* 2, 1 (2014), 73–82.

[5] Paul E Dickson. 2015. Using unity to teach game development: When you've never written a game. In *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*. 75–80.

[6] Betsy DiSalvo, Mark Guzdial, Charles Meadows, Ken Perry, Tom McKlin, and Amy Bruckman. 2013. Workifying games: Successfully engaging African American gamers with computer science. In *Proceeding of the 44th ACM technical symposium on Computer science education*. 317–322.

[7] Brianna Dym, Namita Pasupuleti, Cole Rockwood, and Casey Fiesler. 2021. " You don't do your hobby as a job": Stereotypes of Computational Labor and their Implications for CS Education. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*. 823–829.

[8] Shelly Engelman, Brian Magerko, Tom McKlin, Morgan Miller, Doug Edwards, and Jason Freeman. 2017. Creativity in authentic STEAM education with EarSketch. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*. 183–188.

[9] Casey Fiesler. 2013. The chilling tale of copyright law in online creative communities. *XRDS: Crossroads, The ACM Magazine for Students* 19, 4 (2013), 26–29.

[10] Casey Fiesler and Amy S Bruckman. 2019. Creativity, Copyright, and Close-Knit Communities: A Case Study of Social Norm Formation and Enforcement. *Proceedings of the ACM on Human-Computer Interaction* 3, GROUP (2019), 1–24.

[11] Casey Fiesler, Shannon Morrison, R Benjamin Shapiro, and Amy S Bruckman. 2017. Growing their own: Legitimate peripheral participation for computational learning in an online fandom community. In *Proceedings of the 2017 ACM conference on computer supported cooperative work and social computing*. 1375–1386.

[12] Jonathan B Freeman. 2020. Measuring and resolving LGBTQ disparities in STEM. *Policy Insights from the Behavioral and Brain Sciences* 7, 2 (2020), 141–148.

[13] James Paul Gee. 2005. Semiotic social spaces and affinity spaces. *Beyond communities of practice language power and social context* 214232 (2005).

[14] Ben U Gelman, Chris Beckley, Aditya Johri, Carlotta Domeniconi, and Seungwon Yang. 2016. Online urbanism: Interest-based subcultures as drivers of informal learning in an online community. In *Proceedings of the Third ACM Conference on Learning @ Scale*. 21–30.

[15] Karen Hellekson. 2009. A fannish field of value: Online fan gift culture. *Cinema journal* 48, 4 (2009), 113–118.

[16] Geoffrey L Herman and Sushmita Azad. 2020. A comparison of peer instruction and collaborative problem solving in a computer architecture course. In *Proceedings of SIGCSE '20*. 461–467.

[17] Matt Hills. 2015. The expertise of digital fandom as a 'community of practice': Exploring the narrative universe of Doctor Who. *Convergence* 21, 3 (2015), 360–374.

[18] Mizuko Ito, Crystle Martin, Rachel Cody Pfister, Matthew H Rafalow, Katie Salen, and Amanda Wortman. 2018. *Affinity online: How connection and shared interest fuel learning*. Vol. 2. NYU Press.

[19] Mary Elizabeth Jones, Melanie Kisthardt, and Marie A Cooper. 2011. Interdisciplinary teaching: Introductory programming via creative writing. In *Proceedings of SIGCSE '11*. 523–528.

[20] Ari Korhonen, Marianna Vivitsou, et al. 2019. Digital storytelling and group work. In *Annual Conference on Innovation and Technology in Computer Science Education*.

[21] Brittany Ann Kos. 2019. Understanding female-focused hackathon participants' collaboration styles and event goals. In *Proceedings of the International Conference on Game Jams, Hackathons and Game Creation Events 2019*. 1–4.

[22] Mark Kretzschmar and Mel Stanfill. 2019. Mods as lightning rods: A typology of video game mods, intellectual property, and social benefit/harm. *Social & legal studies* 28, 4 (2019), 517–536.

[23] Amruth N Kumar. 2012. A study of stereotype threat in computer science. In *Proceedings of the 17th ACM annual conference on Innovation and technology in computer science education*. 273–278.

[24] Stan Kurkovsky. 2013. Mobile game development: improving student engagement and motivation in introductory computing courses. *Computer Science Education* 23, 2 (2013), 138–157.

[25] Jean Lave. 2019. *Learning and everyday life*. Cambridge University Press.

[26] Jean Lave and Etienne Wenger-Trayner. 1991. *Situated learning: Legitimate peripheral participation*. Cambridge university press.

[27] Xinyue Lin, James Connors, Chang Lim, and John R Hott. 2021. How Do Students Collaborate? Analyzing Group Choice in a Collaborative Learning Environment. In *Proceedings of SIGCSE '21*. 212–218.

[28] John Maloney, Mitchel Resnick, Natalie Rusk, Brian Silverman, and Evelyn Eastmond. 2010. The scratch programming language and environment. *ACM Transactions on Computing Education (TOCE)* 10, 4 (2010), 1–15.

[29] Mikko Meriläinen. 2019. First-timer learning experiences in Global Game Jam. *International Journal of Game-Based Learning (IJGBL)* 9, 1 (2019), 30–41.

[30] Briana B Morrison and Jon A Preston. 2009. Engagement: Gaming throughout the curriculum. *ACM SIGCSE Bulletin* 41, 1 (2009), 342–346.

[31] Arnab Nandi and Meris Mandernach. 2016. Hackathons as an informal learning platform. In *Proceedings of SIGCSE '16*. 346–351.

[32] Hector Postigo. 2007. Of mods and modders: Chasing down the value of fan-based digital game modifications. *Games and Culture* 2, 4 (2007), 300–313.

[33] Arnold Rosenbloom. 2022. A Full Stack MMOG as a Two Part Web Development Assignment. In *Proceedings of the 27th ACM Conference on on Innovation and Technology in Computer Science Education Vol. 2*. 603–604.

[34] Irving Seidman. 2006. *Interviewing as qualitative research: A guide for researchers in education and the social sciences*. Teachers college press.

[35] Kiyoshi Shin, Kosuke Kaneko, Yu Matsui, Koji Mikami, Masaru Nagaku, Toshifumi Nakabayashi, Kenji Ono, and Shinji R Yamane. 2012. Localizing global game jam: Designing game development for collaborative learning in the social context. In *International Conference on Advances in Computer Entertainment Technology*. Springer, 117–132.

[36] Kathleen Timmerman and Michael Goldweber. 2022. Department-wide Multi-semester Community Engaged Learning Initiative to Overcome Common Barriers to Service-Learning Implementation. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 1*. 808–814.

[37] Moran Tsur and Natalie Rusk. 2018. Scratch microworlds: designing project-based introductions to coding. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*. 894–899.

[38] Aaron Vanek and Andrew Peterson. 2016. Live action role-playing (LARP): Insight into an underutilized educational tool. *Learning, education and games* 219 (2016).

[39] Bo Kampmann Walther. 2003. Playing and gaming. *Game studies* 3, 1 (2003), 1–20.

[40] Richert Wang and Vincent Olivieri. 2018. Sound design for video games: An interdisciplinary course for computer science and art students. In *Proceedings of SIGCSE '18*. 981–986.

[41] Etienne Wenger-Trayner. 1999. *Communities of practice: Learning, meaning, and identity*. Cambridge university press.

[42] Etienne Wenger-Trayner, Mark Fenton-O'Creevy, Steven Hutchinson, Chris Kubiak, and Beverly Wenger-Trayner. 2014. *Learning in landscapes of practice: Boundaries, identity, and knowledgeability in practice-based learning*. Routledge.

[43] Seungwon Yang, Carlotta Domeniconi, Matt Revelle, Mack Sweeney, Ben U Gelman, Chris Beckley, and Aditya Johri. 2015. Uncovering trajectories of informal learning in large online communities of creators. In *Proceedings of the Second (2015) ACM Conference on Learning @ Scale*. 131–140.