"Can You Help Me?" An Experience Report of Teamwork in a Game Coding Camp for Autistic High School Students

Makayla Moster*, Ella Kokinda*, Matthew Re*, James Dominic*, Jason Lehmann†, Andrew Begel‡, and Paige Rodeghero*
*Clemson University, SC, USA. Email: {mmoster, ekokind, mjre, domini4, prodegh}@clemson.edu
‡Microsoft Research, WA, USA. Email: andrew.begel@microsoft.com
†Aspiritech, IL, USA. Email: lehmanndev@hey.com

ABSTRACT

Teamwork skills are increasingly important for students to have as they enter the workforce, especially in software development positions. However, autistic students do not get to practice teamwork since much of their education is focused on learning social skills. The hybrid mode of education comes with challenges, including communication and collaboration issues and teaming difficulties, however, this method of teaching and learning can be difficult for students with autism. In this experience report paper, we discuss our experience planning and running a hybrid camp to teach teamwork and programming to 14 autistic high school students. Overall, our camp was successful in teaching students software development skills with open source software, and, from our experience, we detail our lessons learned and provide recommendations for educators and researchers working with autistic students in a hybrid setting.

CCS CONCEPTS

• **Human-centered computing** → **Accessibility**; *Collaborative and social computing*; • **Social and professional topics** → *Computing education.*

KEYWORDS

hybrid, autism, game coding camp

ACM Reference Format:
Makayla Moster*, Ella Kokinda*, Matthew Re*, James Dominic*, Jason Lehmann†,, Andrew Begel‡, and Paige Rodeghero* . 2022. "Can You Help Me?" An Experience Report of Teamwork in a Game Coding Camp for Autistic High School Students. In *44nd International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET '22), May 21–29, 2022, Pittsburgh, PA, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3510456.3514154

ICSE-SEET '22, May 21–29, 2022, Pittsburgh, PA, USA

ACM ISBN 978-1-4503-9225-9/22/05…$15.00
https://doi.org/10.1145/3510456.3514154

1 INTRODUCTION

Teamwork skills are becoming increasingly more important for positions such as software development [30]. Many universities implement a capstone course for computer science or software engineering majors to teach students skills such as teamwork and communication [38]. Throughout these projects, students are teamed together to complete a small-scale software development project. This project gives university students an opportunity to practice and hone their soft skills, such as communication, teamwork, and collaboration, in a low-stakes environment before joining the workforce.

Unfortunately, autistic students frequently do not have an opportunity to practice teamwork skills in a technical and goal-based environment, as much of their education is devoted to learning practical social skills [13] including one-on-one with therapists, speech pathologists, and other service providers. However, when working in groups, autistic children tend to have difficulty tacitly picking up social norms, especially when collaborating remotely [14]. Remote interactions via text-oriented computer-mediated communication (CMC) provide many affordances for "increased comprehension of and control over communication" and "contact with and social support for similar others who may be geographically distant" [24]. Gaming environments have been shown to have a significant positive impact on providing a community for autistic youth to build friendships [55]. By combining a learning environment that affords psychological and social safety with specially-designed scaffolding [58], we can teach autistic youth how to negotiate new social situations involving collaborative group work and software development.

The main vehicles of our educational pedagogy are informal learning opportunities in coding camps. Over the last 10-15 years, many organizations have offered summer coding camps for children and teenagers [9] which provide volunteer opportunities for students to explore domain-specific programming activities [1, 63]. They also offer a great deal of autonomy for students to engage as deeply with the material as they wish [60]. This kind of informal learning is opportunistic, unstructured, and self-directed, and can be incredibly motivating due to its strong connection to a student's personal interests and activities [9].

We designed and implemented a two-week hybrid summer game coding camp for 14 autistic high school students to expose the students to programming while also working on teaming, communication, and collaboration skills.

Due to the COVID-19 pandemic, we offered the students and their parents multiple options for participation that accommodated for personal and medical considerations and travel restrictions. Of the 14 student participants, 8 students attended the camp in-person and 5 students attended the camp remotely. Additionally, one student participated remotely the first week and joined the in-person group during the second week. This created a unique hybrid teaching and learning environment where all of the instructors were remote and the students were both in-person and remote.

In this paper we give motivation for presenting a hybrid coding camp and background in remote and hybrid learning for students with Autism Spectrum Disorder (ASD) and coding camps. We then discuss briefly the inspiration for our camp and the informed decisions and planning that resulted in our finalized camp design. A detailed breakdown of our camp is also included. Finally, we reflect on and discuss the lessons we learned from our camp implementation and provide recommendations for both academia and industry.

2 MOTIVATION

As of 2020, the Centers for Disease Control (CDC) reported that 1 in 54 children in the USA are diagnosed with ASD [43, 59] (around 6 million children). Only 17% of them (around 1 million) enroll in four-year colleges and only 39% of those (around 390,000) graduate [12]. There are also lower rates of employment for young adults with ASD than those who are non-disabled or with other disabilities overall [7, 46, 57]. To support young adults with ASD in finding and maintaining employment in the technology industry, a program in Australia has been placing students with ASD in real-life work environments to help them gain exposure to a workplace that matches their interests [41].

Coding camps also help to foster students' interest in programming and future development jobs by providing mentoring and exposure to computing [40]. Although there are many coding camps for K-12 students, only one, Tech Kids Unlimited[1], targets the autistic student population [56]. Our work is primarily focused on teaching basic programming skills to autistic high school students [54]. However, this leaves out significant curricula on communication, social interaction, and teaming, skills which, for decades, industry experts have been telling academia that they expect new software developers to have before entering the workforce [10, 16, 18, 50, 62]. Lack of these skills is a common weakness across the general student population [20] and is known to be incredibly challenging for autistic individuals working with non-autistic colleagues [6, 29, 48, 53]. As of yet, no other coding camps offer autistic students a curriculum designed to develop these skills. This experience report details our preliminary work in this area, lessons learned based on our camp, and our recommendations for educators and researchers.

[1] https://www.techkidsunlimited.org

3 BACKGROUND

In this section, we discuss teamwork, hybrid education, hybrid camps, camps for autistic individuals, and a previous game coding camp for autistic students.

3.1 Teamwork

Teamwork is prevalent within the software development community. Most college students taking computer science classes are required to take a software development course where there is a semester-long group software development project. These courses are meant to teach the students industry expectations, collaboration, and teamwork [52]. Within college courses, most teams are collocated, but industry has seen an uptick in the number of geographically distributed teams. Within our camp, we split students into both collocated and distributed teams for the last week.

3.1.1 Collocated Teams. Collocated teams are teams in which the members are located in the same area or workplace. These are more traditional software development teams, many development methodologies, such as the agile method, have been designed for collocated teams [61]. Due to the nature of agile methods and collocated teams, team members contribute more visibly than their distributed counterparts, however social loafing is still present [23].

3.1.2 Distributed Teams. Distributed teams are teams in which the members reside in different geographical areas, areas which may be in different timezones. There has been plenty of research on distributed teams, such as conflict research [31], trust in teammates [15, 35], collaboration [19], etc. Due to the COVID-19 pandemic, most software development companies had their teams work remotely, simulating a distributed team. One research study during COVID-19 found that developers may have trouble communicating with their coworkers, struggle with brainstorming as a team, and have less awareness on other coworkers' projects [47]. Even before the pandemic, teams struggled with working differences; one study investigated social loafing behavior within distributed teams. They found that when distributed teams explicitly specify the expectations of their team members they can combat social loafing behavior [23]. Another study investigated issues with teamwork such as lack of a common language, lack of information about teammates' projects, conflicts between teammates, cultural differences, and time zone differences affected team effectiveness [21]. Due to the struggles with adopting collocated software development methods in a distributed setting, many researchers are studying how to translate these methods into viable solutions for distributed teams [36, 37, 39]. However, there are benefits to distributed working, since it can be more accessible for disabled workers due to overall flexibility [47].

3.1.3 Teamwork & Autism in Industry. Many researchers agree that autistic individuals have a tendency to pursue careers in technology [45]. One study found that common stressors in software developers include individual team interactions [33]. Annabi *et al.* found in their study that autistic individuals

have the skills necessary to succeed in software development careers, however there may be challenges with communication and interpersonal relationships that may be a source of stress for the developer [5]. We hope that by providing an early technology-based teamwork opportunity for autistic high-schoolers, we can help mitigate and reduce the stress of working as a team in their future careers.

3.2 Hybrid Education

Hybrid learning, also known as high-flex or blended learning, is a new concept in education. Due to this, there are many definitions of hybrid learning, however they all incorporate the same aspects, including high flexibility and learning content partially online and in-person [3, 22, 32, 44, 64]. Due to the COVID-19 pandemic, hybrid education has become more popular since it allows students to have a high flexibility in their learning. Since hybrid learning involves a large amount of moving parts, there can be many challenges from both the student and instructor perspectives. First, instructors must adapt their teaching approaches to an online environment while maintaining their in-person standards of instruction [51]. Next, instructors must also become knowledgeable about the technology used, inexperience may impact students' learning abilities [51]. For students, instructors must ensure that both students that are in-person and online are receiving equal amounts of instruction [51].

One study found that students who were instructed via hybrid learning had more achievements in learning than their in-person counterparts [26]. However, another study found that high school students with and without disabilities have no preference for the accessibility of blended learning versus in-person learning and they also have an overall negative opinion of hybrid learning [4]. Engagement is also difficult to measure within blended learning settings [25]. However, there is very little research on hybrid learning with autistic students, especially high school-aged students. There is also little research on hybrid educational coding camps. We hope to begin to close this gap with this experience report by inspiring other researchers to study this area.

3.3 A Game Coding Camp for Autistic Students

Begel *et. al* conducted a fully online game coding camp for autistic students during the summer of 2020 due to the COVID-19 pandemic [8]. The main goal of the camp was to build teamwork skills and practice and improve communication skills. The seven campers were rising college freshmen who enrolled in the camp due to their interest in video games. The camp utilized MakeCode Arcade as the programming language, while still focusing on game development. Findings of the camp include increased communication between team members and positive collaboration experiences within teams. The camp also taught the instructor team about educating students with autism in an online environment, including allowing students the freedom to turn their cameras and microphones on or off, involving an instructor in breakout room activities, piloting teaching material before camp to make adjustments, and the importance of scaffolding assignments [8]. These lessons helped shaped our coding camp for high school autistic students.

3.3.1 Rationale for Change. Based on feedback received from students and instructors after Begel *et. al's* online game coding camp, the instructor team decided to utilize a game development environment with richer capabilities and features [8]. The criteria for choosing a new two-dimensional (2D) game development environment included an option for the use of traditional programming languages (i.e. Java, Python, C#), integrated version control software, and the option for students to collaborate on their code together. The team investigated four different environments which included Minecraft modding[2], Unity game engine[3], Godot game engine[4], and MakeCode Arcade[5]. All tools considered are free to use. Godot game engine was the only environment that met all of the team's criteria – Godot is a free, open-source, professional development environment with documentation, it has multiple languages to choose from (GDScript, C#, C++, and visual scripting), and plugins to integrate GitHub[6] and utilize Visual Studio (VS) Code's Liveshare[7] feature for student collaboration and pair programming. This choice led the team to redesign the curriculum provided from Begel *et. al's* camp to utilize all of Godot's features [8].

4 CAMP DESIGN & OBSERVATIONS

In this section, we cover design changes from Begel *et. al's* camp and motivations for those changes for our camp, and our finalized camp design. Within the camp design we will cover the learning outcomes for the camp, students and instructors, hardware and software requirements for the camp, our teaching materials, and a timeline of topics covered during our two-week hybrid camp.

4.1 Our Game Coding Camp Design

Due to relaxed COVID-19 restrictions in June 2021, we were able to offer the camp in a hybrid format. Students and their parents could choose whether they wanted to spend the two week camp fully remote, on Clemson University's campus, or split between one week fully remote and one week in-person. Students also had the option to just attend during the first week of camp or to attend both weeks.

4.1.1 Learning Outcomes. For the camp, we focused on teaching the students a real, industry-level programming language as well as exposing them to a real team-based programming environment. We also focused time on essential professional software development skills such as collaboration, communication, and teamwork. To do this, we used the Godot game

[2]https://dev.bukkit.org
[3]https://unity.com
[4]https://godotengine.org
[5]https://arcade.makecode.com
[6]https://github.com
[7]https://code.visualstudio.com

engine, GitHub, and VS Code during camp, which allowed students to learn version control with GitHub as well as expose them to pair programming with VS Code's Liveshare feature. As a bonus, both of these tools were integrated within the Godot programming environment.

During Week 1 of camp, we focused on teaching the students the basics of the Godot environment and GDScript so each student could contribute during Week 2 when we broke the students into teams to work on their final project. Week 2 involved working in collocated and distributed teams; we teamed in-person students together and online students together to simulate real world work environments. During this teaming, we had students work together to program their games using the traditional driver-navigator roles [2, 65]. Each day, a different student was chosen to be the driver while the others in the group were navigators.

We also wanted students to produce a game that they wanted to play, so we gave the student teams the artistic freedom to create a game of their choosing. Most teams chose to produce shooters or story-based games while one team wanted to create a rogue-like dungeon crawler. Since the instructors did not fully teach how to create these game types, students needed to reference the official documentation when they were stuck working on their games. Instructors also encouraged this behavior by helping students come up with keywords to put into their search engines, which is a common occurrence for professional software developers [34].

4.1.2 Camp Instructors. A total of 10 instructors conducted the camp. Eight of those instructors have had previous programming experience, one instructor is a Computer Science Professor at Clemson University, one instructor is a researcher at Microsoft Research, two instructors are professionals in industry, and four instructors are graduate students in Computer Science and Human-Centered Computing at Clemson University. The other two instructors included a psychotherapist as well as an Applied Behavioral Analysis (ABA) therapist. Half of the instructors helped conduct Begel *et. al's* camp the previous year including the professor from Clemson University, the researcher at Microsoft Research, one graduate student, one industry professional, and the ABA therapist [8]. This year, while most of the instructors came from Clemson University, three instructors came to help from other parts of the United States. The instructors were in different timezones, with the majority being in the Eastern timezone. Our camp also had an uptick in volunteer instructors, one volunteer is a Quality Assurance (QA) lead in industry as well as an advocate for students with autism, and the other volunteer is a psychotherapist with interests in autism advocation. The camp also had in-person support provided by Clemson University's Summer Scholars Program.

4.1.3 Students. A total of 14 high school students participated in the camp. The ages of the students ranged from 14 years old to 18 years old. All of the students were male. Five students participated remotely. Eight students participated in-person. Two students chose to only attend the first week of camp. Additionally, one of the students participated remotely and then joined the in-person students for the second week of the camp. All of the students resided in the United States and came from all over the country to attend our two week camp.

4.1.4 Hardware & Software Requirements. To participate in camp, all students were required to have their own computer or laptop, headphones, camera, and microphone. No hardware was provided by the camp for the students to use. Students were also required to have the Godot programming environment installed as well as Zoom[8] and VS Code before the first day of camp. Other requirements for students involved an account for Google Drive, where all teaching materials were stored, as well as a way for students to contact and talk with their teammates during Week 2. Students were able to choose their own contact method, some teams utilized Discord[9], others texted, and some emailed each other.

Instructors were also required to bring their own equipment, including a computer or laptop, headphones, camera, and microphone for camp. Other software that was required for instructors included Microsoft Teams[10], Open Broadcasting Software[11], an email account for access to the camp's Google Drive folder, and a GitHub account.

4.1.5 Programming Environment & Add-Ons. During our initial camp design, we went through many ideas for programming environments to be used within the camp. Begel *et. al's* version of this camp utilized MakeCode Arcade, a browser-based visual programming language. After testing multiple programming environments, we decided to use Godot, a free, open-source, professional game development language. Since Godot is a professional programming language, there are many bells and whistles that go along with it. The organization of Godot's environment is node-based, there is a root node with other nodes branching from it. The branch nodes contain smaller pieces of the game such as characters and backgrounds, and are added into the root node to create a functional game. Additionally, incorporating GitHub and VS Code plugins into the environment increased the complexity of students' computer desktop and navigation through windows.

4.1.6 Teaching Materials. For the purposes of reproducibility and independent study, we have made all teaching materials available via an online appendix [49].

4.1.7 Instructor Preparation & Debrief Meetings. The instructor team virtually met 30 minutes prior to the start of camp each day via Microsoft Teams to review the schedule for the day and address any concerns that an instructor had regarding the prepared material. These meetings also allowed instructors to share the day's updated teaching material with each other as well as discuss and strategize how to best help the students create their games.

After camp each day, the instructors met for an hour to discuss what had happened that day. This allowed the instructors

[8] https://zoom.us
[9] https://discord.com
[10] https://www.microsoft.com/en-us/microsoft-teams/group-chat-software
[11] https://obsproject.com/

to correct-course when things went awry as well as discuss student behavior and intervention strategies. We also used this time to reorganize our teaching material and discuss each student team, their game, and their current status.

4.1.8 Camp Timeline. Table 1 is the timeline of teaching material for the two-week camp. Week 1 was setup for students to get to know one-another and learn the basics of Godot. Week 2 was designed to allow the student teams to work together on their final project. Each day, all instructors and students logged into Zoom from their personal computers to begin instruction. In-person students joined the Zoom call on their personal computers each day from a classroom on Clemson University's campus and were supervised by two in-person instructors as well as camp counselors from the Summer Scholars program. The Summer Scholars program utilized the non-instructional time for the in-person students each day, facilitating camp activities such as meal time, rock climbing, socializing events, campus tours, etc. Remote students logged into Zoom from their personal computers at their residence and interacted with in-person students only during the three hour instructional period.

Each day, students were given links to the day's scaffolded instructions so they could follow along and links to their daily journals where they would have activities to work on throughout the camp day and as homework. Each camp day was 3 hours long with at least two 15-minute breaks for students to stretch and grab a snack. Additionally, Zoom breakout rooms were utilized at 10-15 minute intervals during programming instruction each day to allow small, randomized student groups (Week 1) and student teams (Week 2) the chance to have one-on-one help sessions with a virtual instructor.

4.1.9 Student Feedback. At the end of camp (after the project presentations), we interviewed students about their camp experience and asked for their feedback regarding what could be changed for future iterations of the camp. We asked students what one thing they would change about the camp; most in-person students recommended to change the building layout on campus to require less walking and to improve the food quality in the dining halls. However, other students noted having technical issues such as parental computer locks, forgotten headphones, and forgotten laptop chargers. We also asked students about their teamwork within the camp and they were glad that the instructors chose the final project teams versus the students self-assembling into teams. Most students thought that working in a team during this camp was better than the other times they have worked in teams during school due to their teammates being "better than previous ones." We also asked what the most important lesson students learned during camp, and the majority responded with time management. They realized that with the time constraints of camp, there was no time to produce a full game, and most teams settled on making a demo version for their game. Finally we asked students what their most improved skill was and we got a range of responses from game creation skills such as pixel art and GoDot to interpersonal skills including communication and collaboration.

TABLE 1 **Game Coding Camp Timeline**

Day 1	Camp Introduction & Icebreakers
	Instruction: Video Game Elements
	Discussion: Other Game Elements
	Discussion: Elements of Flappy Bird
Day 2	*Instruction*: GitHub Introduction
	Discussion: GitHub & Its Uses
	Instruction: How to Design Games
	Discussion: Game Design
	Programming: Flappy Bird Character Movement
Day 3	*Programming*: Flappy Bird Character Movement
	Instruction: Sounds in Games
	Programming: Flappy Bird Game Sounds
Day 4	*Programming*: Importing Assets
	Instruction: Game Obstacles
	Programming: Flappy Bird Obstacle Movement
Day 5	*Programming*: Flappy Bird Obstacle Movement
Day 6	Intra-team Introductions
	Instruction: Pixel Art
Day 7	*Instruction*: Narrative Design
	Programming: Character Movement
	Final Project Brainstorm
Day 8	*Programming*: Shooting Projectiles
Day 9	Final Project Workday
Day 10	Final Project Presentations Post-Camp Interviews

5 LIMITATIONS

Due to the COVID-19 pandemic, we believed it was in the best interest of the health of students and instructors that we host the camp in a hybrid format. This, however created discrepancies between the in-person and online teams. For instance, where in-person students were able to work together in the evening after camp concluded, online students were limited to their regular at-home schedules. We also had a majority of instructors participate online, which limited the student-instructor interaction which can be key for success in ASD students. We hope in future iterations of the camp we will be able to host the camp exclusively in-person and provide opportunities for students to work with each other and the instructors outside of the allotted instructional time.

The team faced many limitations in the use of videoconferencing software with regard to being able to mute and unmute students. Some students had difficulties with unmuting themselves when wanting to speak, some students would not mute and leave their microphone on during lessons, causing distractions for the other students and the teaching instructor. While these problems persisted throughout both weeks of the camp, the instructors did not want to auto-mute participants and have them request to unmute themselves when they wished to speak as the team felt that was over reaching and inappropriate for the age group.

Some students were also less prepared than expected with their hardware set-ups, with some students having machines incapable of running the Godot engine alongside an active Zoom call and internet windows. Though the team provided hardware requirements when advertising the camp, we found it difficult to enforce before students arrived at camp. We also did not expect some of the students to have inexperience with the basic operations of a computer such as right clicking, finding settings, understanding file structures, and knowing what operating system they were using. We mitigated this issue by utilizing Zoom's remote control feature, allowing instructors to take control of students' computers for a short time while also helping them with more complex computer operations.

6 CHANGES & ADJUSTMENTS

In this section, we discuss the schedule adjustments that were made during the camp, how we improved the students' behavior to one-another, our observations on student-to-student and team interaction as well as how well the camp was executed and met expectations as a whole.

6.1 Schedule Adjustments & Being Over Prepared

Begel *et. al's* study utilized MakeCode Arcade as the development environment, however, for this camp camp the instructor team utilized a new environment. Therefore, the team could not use the previous previous MakeCode teaching materials and developed new material focused on the Godot environment. Due to the change in programming environment, each day's teaching material was overly packed with instruction and activities that were estimated to last for the full 3 hours allotted to camp time. However, the team quickly learned that we had not planned for the students' inexperience with computers in addition to our instructional material. Their inexperience forced the instruction team to reduce pace and hold additional one-on-one help breakout sessions with instructors. This led the instruction team to reorganize and reduce the amount of instruction during their debrief sessions each day, while increasing the amount of breakout help sessions for students to get caught up. Due to spending a lot of time on helping students navigate their operating systems, the team also had to remove the majority of their pair programming and GitHub content after the student introductions. These adjustments also caused the team to greatly reduce the amount of time allotted for students to work together on their game during Week 2 – instead of having the student teams work together for 2 whole camp days, student teams were only able to work together during the last camp day leading to unpolished final games.

6.2 Student Behavior

Other adjustments during the camp came from the students' behavior towards their peers. During Week 1, the instructor team noticed some students bullying[12] one another within the Zoom chat as well as verbally on Zoom and within the classroom. This caused the team to have to enact kindness rules for all students to follow. The lead instructors spent 30 minutes facilitating a discussion with the students around kindness and how to treat one-another. This discussion led to the instructors and students collaborating together to create a total of four guidelines for the students to follow for the rest of camp. Additionally, during the first few break sessions, instructors noted that students were not getting up and away from their computers. The instructor team felt the students needed to get away from the screen during these times, and came up with creative ways to get students out of their chairs such as having them complete 15 jumping jacks or get a snack. Instructors who were in-person also held small push-up or jumping jack competitions or short walks outside to encourage the students to get out of their chairs. Getting students up and away from their computers during breaks also improved student attitudes and attention throughout the day.

6.3 Student-to-Student & Team Interaction

Throughout camp, there were noticeable changes in how students interacted with one another and with the instructor team. Instructors noticed during the beginning of Week 1, most students were timid, not wanting to speak during breakout sessions until prompted by the instructors. Students also did not want to acknowledge that they needed help or were behind on instructions within the breakout sessions, which required instructors to ask students to share their screens to provide help. However, as the students became more accustomed to how the camp worked, students started becoming

[12] Due to it being difficult to categorize student intentions, the team chooses to utilize the term 'bullying' over 'acts of unkindness.'

more comfortable with one another, initiating conversations about the newest videogames, their pets, or outerspace facts.

During the camp, we also noticed changes in how team members interacted with each other. During the beginning of Week 2, the instructors notices that student teams were apprehensive due to the change in routine, which required instructors to facilitate team discussions by asking questions and prompting individual students to speak to their teammates. Student teams also struggled to create a cohesive game design, instead opting to stick to their individual game ideas rather than compromise and combine ideas. However, as the week progressed, students compromised on their game design and became more outspoken by initiating conversations with their teammates about ideas and aspects of their game design, sometimes even ignoring the breakout room instructor completely until they required help implementing an idea. At the end of the camp, some student teams wanted to stay connected and exchanged contact information to continue working on their game or on other projects such as modding Minecraft. Initially when we created student teams, we asked students about their previous group project experiences, where the majority of students mentioned that all previous group project experiences were unpleasant due to the quality of their former teammates. After the conclusion of the camp, we asked students if this experience was better or worse than their previous group projects, and all students responded that the experiences in this camp were better than any previously, with the majority of students stating that their camp team was better than any of their previous teams.

6.4 Further Camp Reflections

Although there were some setbacks and changes needed to be implemented on-the-fly, our coding camp was successful and, from our observations, met most of our defined learning outcomes. Due to time constraints, we were unable to focus on teaching the group version control with GitHub and also had to remove pair programming from our curriculum. We were successful in teaching students a real, industry-level game development environment and, during Week 2, students showed proficiency in teamwork and communication, development environment navigation, and game creation with the creation of their final projects. Students also showed proficiency in problem-solving skills by utilizing a search engine to research how to incorporate aspects into their games that were not part of the camp instruction. Multiple groups also used Liveshare within VS Code to program their final project together, and one advanced group utilized GitHub for version control and sharing their project with others. Student teams exceeded instructor expectations with their videogames and teamwork, many teams had multiple characters or enemies to avoid or destroy and were proud to present their hardwork at the end of the camp. Although each student group had different challenges in teamwork and collaboration, all of the feedback provided by students at the end of camp stated that students perceived this experience as their best group project so far. In addition, we found that students were inspired to continue with their game development dreams by our guest speakers.

7 LESSONS LEARNED & RECOMMENDATIONS

We have learned many lessons through the development and implementation of our camp for autistic high school students. Based upon our experience and previous work, we also provide recommendations for educators and researchers when working with autistic students in a hybrid environment. We believe that it is important to note that these recommendations can also improve education for students with other disabilities and can create a more inclusive environment for all. In the following section, we discuss our lessons learned, including what did and did not work in our camp design and our recommendations for educators.

7.1 Unanticipated Student Behavior

During the camp, students presented behaviors that the instructional team did not anticipate, such as students causing disruptions during classtime or working as a team overnight on their final project. Through these experiences, we have learned many lessons and have recommendations for others.

7.1.1 Reducing Distractions. We found that the students were easily distracted in the Zoom chat as well as during instruction. For example, students would initiate conversations about new video games or outer space and others became distracted by the messaging notifications. Allowing students to freely unmute their microphones or having unmoderated chat windows did not work within our camp. To reduce the distractions for both the students and the instructors, at the beginning of each day, we shared a blank Google Doc, called the camp conversation board, with the students to provide a space for them to have off-topic discussions. This proved to decrease the off-topic comments in the chat and student outbursts during instruction, and the students appeared to have improved focus during the instruction. We recommend for other educators experiencing the same issues to provide a space for students to write down their thoughts or hold short discussion sessions where students can talk freely.

7.1.2 Promoting Kindness. We also found that some students consistently bullied one another both verbally and using the Zoom chat. We would interrupt students to remind them to be kind to one another. Unfortunately, this reminder did not help decrease those types of interactions. About halfway through the camp, the lead instructors held an open forum with the students to discuss how students should treat one another. The instructors created a shared document to provide guidelines for how to treat one another in a learning environment. The instructors asked for students to provide input into the document and facilitated a discussion around promoting kindness within the classroom. From the discussion, four main guidelines were created within the shared document for the students to follow throughout the rest of camp.

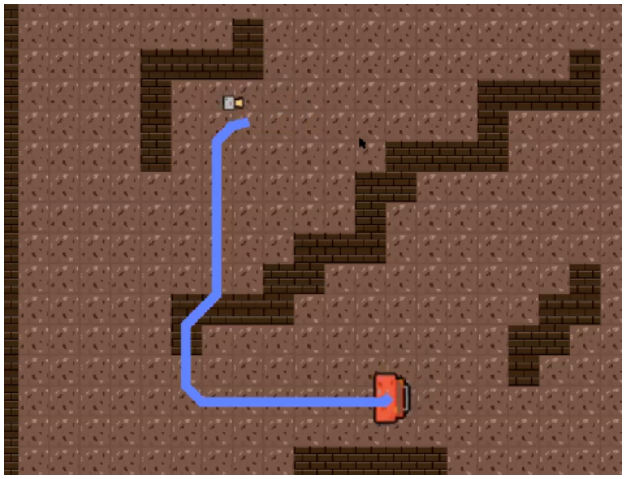

Figure 1: Team Lyra's presented game with implemented path finding for their enemy character using the blue line.

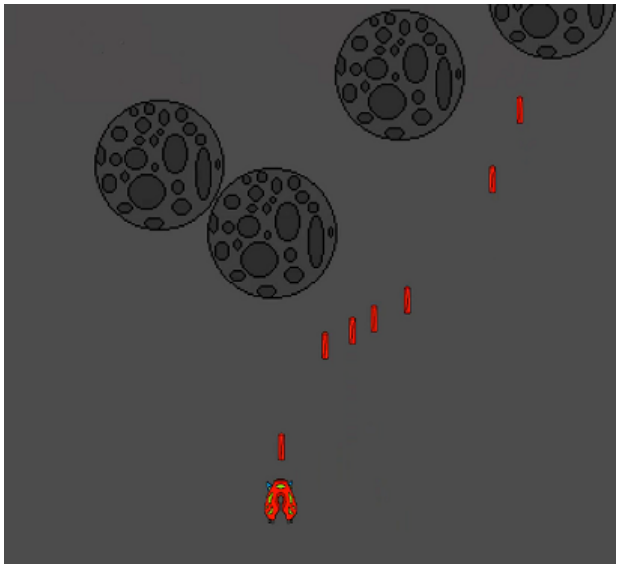

Figure 2: Team Taurus's game where one student focused solely on game programming and one focused on the art and assets for the game.

The guidelines included not speaking out of turn, raising a virtual hand to speak, utilizing the camp conversation document to have off-topic discussions, and allow one person to speak at a time. The guidelines were also referenced by the instructors throughout the rest of camp when students needed reminders. We found that hosting a discussion with the students about kindness and creating and implementing rules together worked well for our camp. Therefore, we recommend other educators hold a discussion about kindness with their students and together create rules for students to follow encompassing kind behavior.

7.1.3 Affording Extracurricular Student Work & Interaction. As part of the camp, we did not require or have preconceived notions for students to work outside of camp hours on their project. However, Team Lyra, one of the in-person teams, worked the night before the end of the camp to implement pathfinding in their game, which was unprompted and unexpected and only revealed to the camp instructors on the day of their presentation shown in Figure 1. The ability for the students to work together outside of the game coding camp hours proved to be very helpful for this team.

Additionally, two of the remote students on the same team expressed interest in staying friends and communicating via telephone to expand upon the game they created during camp, and inquired if the other would like to collaborate on Minecraft mods as one camper liked the art style of his teammate.

Overall, the hybrid design of our camp worked well. However, we would recommend for other hybrid camps utilizing the same design to incorporate additional, scheduled Zoom meetings in the evening for online students to have time to work together outside of instructional periods.

7.1.4 Participating & Distributing Work Equally. For students, we expected them to contribute and share equally in all aspects of the game development process, but some students tended to prefer one aspect over another, whether that be coding, asset development in Piskel[13], or coming up with the overall story and theme of their team's game. We had no way of mediating or showing that a student had implemented part of the coding of their game. Though this mirrors real-world game development teams by taking on specific roles, it fell short of our expectations of the group sharing in all aspects of the game development process, and we had no means of reconciling with those who only wanted to contribute to one aspect of the team's game.

An example of this was found in Team Taurus, a remote team, where they were able to produce a final game in Figure 2, but we were unable to measure the amount of overall learning between the two teammates where one focused on the artistic design and the other programmed the game.

To mitigate this issue, we recommend that instructors encourage students to share their screen during breakout room sessions. Screen sharing is a great way to build up student confidence in their own work and their ability to describe their concerns or bugs. It can be intimidating for students at first due to their fear of showing that they fell behind or needed help, however, the positives vastly outweigh the negatives. Sharing screens also allows for students to make comments on each others progress and share some of their own ideas with their peers. We implemented screen sharing during the last week of camp and found that some students were not following along during most instructional periods, causing them to require additional one-on-one sessions with instructors to get caught up to the rest of the group.

[13] https://www.piskelapp.com

7.1.5 Forming a Connection with the Students. We found that when instructors focused early on forming a personal connection with the students, the students were more confident and willing to ask the instructor questions. Since each instructor's group of students changed each day, the students became more comfortable with most instructors, leading them to become more comfortable with the group overall. Instructors formed bonds with students over video games, hobbies, pets, food, and even sleep habits, allowing students to show interest in and ask questions about the instructors' jobs and collegiate experience. From our experience, we recommend that instructors form a connection with students during early class sessions over topics that students believe are important, like videogames or hobbies.

7.2 Designing a Camp for Autistic Students

We also learned lessons from how we designed our camp, including universal design, preparation and debrief sessions, and course content. Through the design of our camp and reflecting on the lessons learned, we have produced recommendations for other educators and camp designers.

7.2.1 Implementing Universal Design. Educators need to understand that there is a growing diversity within the classrooms that requires innovative learning and teaching methodologies [11]. Learning about universal design for their classrooms can help educators to become empathetic towards these challenges and help educators to better develop inclusive teaching methodologies and materials [17]. Additionally, educators will be able to make the appropriate decisions within their classrooms regarding instructional material, educational activities, teamwork activities, and team assembly.

7.2.2 Providing Scaffolded Instructions. Like any traditional classroom setting, it is inevitable that a student occasionally falls behind. The likelihood of this occurrence is even greater when working with ASD students should the provided information begin to overwhelm them [27, 28]. Professionals believe that providing written instructions benefits autistic students enormously within classrooms [27]. For our camp, we provided written scaffolds of every lecture which consisted of pictures and step-by-step instructions on completing the day's instructions. By providing these scaffolds to the students every day, they were allowed to catch up to the lecture at their own pace without the need to interrupt the instructor or their classmates. We recommend providing students with daily scaffolded instructions in case they fall behind so students can catch up at their own pace without derailing other students.

7.2.3 Holding Preparation and Debrief Sessions. Meeting pre- and post-camp enabled the instructor team to know what needed to be accomplished each day and reflect on how the day went. Preparation sessions set the schedule of lessons, resolving any interpersonal student conflicts, and informed the other instructors of their progress and if they expected any student groups to need help. Debrief sessions after each day of the camp allow for instructors to evaluate and report on the daily student successes and challenges that could be incorporated into the next day's lessons. Due to our positive experience, we recommend to others to hold preparation and debrief sessions to ensure that all instructors are aware of any changes to the instructional material or student conflicts.

7.2.4 Implementing Common Game Mechanics & Features. During our lectures on how to pitch and describe narratives in games, we scoped only for narratives and actions motivated by storytelling, however, many of the students' favorite games were more open-ended where the mechanics or player actions create a narrative, as opposed to a story describing the game. We expected students to take inspiration for features that are regularly seen in common video games, such as shooting projectiles or receiving damage when the player collides with an obstacle. While in smaller groups, students described games that were system-driven with interconnected game mechanics that would foster a novel experience, similar to systems in games that create "random encounters" which may not be a part of the main story line, but immerse the player in the overall experience playing the game.

However, students began to run into issues when they set their sights on a feature that had a complexity far outside the planned and provided lectures and time constraints of the camp. This unfortunately made some students feel restricted on what they believed they could add to their game and resulted in some groups going too far and leaving out reasonable features from their game. To try and assist these students, we found it effective to take a survey of key features the groups wanted to implement, such as firing projectiles from a single source, and shifting lesson plans around so that more popular features were highlighted and demoed in lectures while less popular features were reserved to our previously recorded YouTube video lectures. We recommend that instructors survey students early-on in the camp about features they want to see in their final projects to rearrange the lecture material based on the most common features.

7.2.5 Taking Breaks. Breaks are an important part of the learning process. We implemented breaks each day during our camp, giving students at least two 10-15 minute breaks per day during our 3 hour camp. During our camp, we found that breaks allowed all of the students, both remote and in-person, to socialize with one another by playing short games[14]. We also encouraged students to get away from the computer screen and walk around during break time to mitigate Zoom fatigue. The in-person students were provided activities that encouraged them to get out of their seats during breaks as well. We recommend that instructors ensure that there are at least two breaks scheduled within their lesson plans each day and that there are predetermined activities for students to have an opportunity to socialize with their peers.

7.2.6 Being Prepared to Change Lesson Plans. During the camp, we found that we had to update our lesson plans significantly in order to finish the major topics. We found that the material took two to three times as long to cover than we had originally had planned, resulting in topics getting significantly cut down

[14] https://garticphone.com/, https://www.speedtyper.dev/

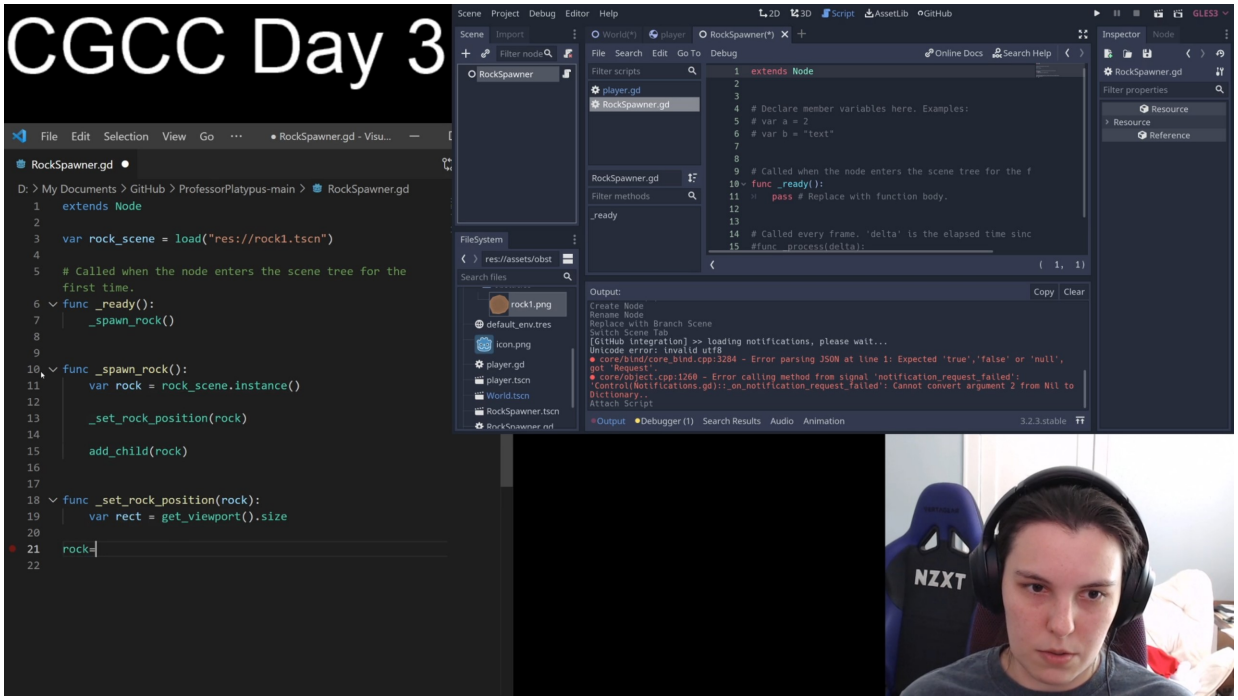

Figure 3: Example video recording layout for Day 3 with the VS Code window on the left, Godot environment on the right, and the instructor's video in the bottom right corner.

or removed entirely. We also found that instructors were not prepared for instances in which a majority of students do not need a specific lecture topic for their game design due to lack of student interest or time (e.g. projectiles or life points). We recommend that instructors be prepared for some topics to go unused and have multiple backup lectures prepared for instances where students are not interested in the material to be presented on a particular day.

7.2.7 Utilizing Simple Zoom Backgrounds. We recommend that educators create plain Zoom backgrounds (e.g., a light blue background with a simple logo) to reduce distractions within classrooms and for instructor identification purposes. Backgrounds can cause autistic individuals to be distracted and in a hybrid camp, there could be many videos being displayed, causing a multitude of distractions. Educators can also create simple backgrounds and share with hybrid students so that everyone has the same background when sharing their video. Backgrounds could also be provided to the students to maintain consistency with the educators and reduce unnecessary distractions with their own backgrounds.

7.2.8 Pre-recording Technical Lessons. After a concrete lesson plan was established, we found it useful to create a day-length pre-recorded teaching video that followed the lesson plan for a specific day. This enabled the instructors to practice their teaching and provided the camp with a hard copy of a lesson in case of technical difficulties and for student reference if they faced any challenges or fall behind during instruction.

We found it helpful to make YouTube playlists for the videos each day and within the playlists each video would cover a large topic and include timestamps for detail work in the video. Titles of the videos were descriptive and matched titles in the written lesson plans so students who were using the written instructions could also have a video companion.

For recording the videos, we recommend using the highest recording quality (1080p+), as well as a layout that enables the viewer to see the instructor's face while teaching and the environment in which they are working in, as shown in Figure 3. We also recommend recording the videos as though the instructor were teaching to a live class, for example, without removing errors while showing the full development and debugging process if mistakes were made. Showing the thought process, recapping, and think-aloud are beneficial for a deeper understanding of concepts [42]. Overall, we recommend recording video lectures for each day of instruction as a backup and to help with supplemental student instruction.

7.2.9 Accessing Technology. For in-person students, we recommend utilizing an environment such as a computer lab set up to host students with their specific computer needs during the camp. This recommendation is based on feedback from students who felt that bringing their own device and having to carry possessions around campus was burdensome and from remote instructors who were unable to help with device technical issues (i.e. forgetting a charger or mouse, random bluescreens). Due to the format of the camp, our students were

also required to have the Zoom window, the Godot environment, and an instance of VS Code running on their machines at the same time. Having to switch between multiple windows throughout the camp was a source of frustration for some of the technologically inexperienced students. Other sources of frustration from students come from Godot's unfamiliar tree-like organization structure, confusion about GitHub and its uses, and confusion about how and why to use the Liveshare feature in VS Code. Having a pre-built environment also decreases the need of addressing discrepancies and differences in UI and technical differences between Macintosh and Windows operating systems for in-person students.

8 CONCLUSION

In this paper, we presented an experience report on a hybrid summer camp for autistic high-schoolers. We provided motivation for conducting this coding camp for children with ASD. We also broke down how our camp was designed and discuss our rationale for changes made based upon a previous iteration of the camp. Next, we covered our learning outcomes for the students as well as a breakdown of how each week of the camp was structured. Finally, we discussed our lessons learned through implementing this camp and provide recommendations for educators and researchers working with autistic students in a hybrid format, as well as the limitations of the camp we experienced in our implementation.

9 ACKNOWLEDGMENTS

We thank all of the camp participants. We thank Thomas Beeson, Dr. Jeanette Ashworth, and Conner Phillis for volunteering to help with the camp. Finally, we thank our guest speakers Sofia Rodriguez, Soo Hyun (Andy) Moon, Vanessa Casteñada-Gill, and Kaylin Norman-Slack.

REFERENCES

[1] Joel C. Adams. 2010. Scratching Middle Schoolers' Creative Itch. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education* (Milwaukee, Wisconsin, USA) *(SIGCSE '10)*. Association for Computing Machinery, New York, NY, USA, 356–360. https://doi.org/10.1145/1734263.1734385

[2] Adeola Adeliyi, Michel Wermelinger, Karen Kear, and Jon Rosewell. 2021. Investigating Remote Pair Programming In Part-Time Distance Education. In *United Kingdom and Ireland Computing Education Research conference*. 1–7.

[3] Alaa AlKhaleel. 2019. The advantages of using blended learning in studying English as a foreign language at the University of Tabuk. *Modern Journal of Language Teaching Methods (MJLTM)* 9, 2 (2019), 1–7.

[4] Rebecca Alvarado-Alcantar, Randa Keeley, and Breanna Sherrow. 2018. Accessibility and usability of preferences in blended learning for students with and without disabilities in high school. *Journal of Online Learning Research* 4, 2 (2018), 173–198.

[5] Hala Annabi, Karthika Sundaresan, and Annuska Zolyomi. 2017. It's not just about attention to details: Redefining the talents autistic software developers bring to software development. In *Proceedings of the 50th Hawaii International Conference on System Sciences*.

[6] R. D. Austin and G.P. Pisano. 2017. Neurodiversity as a Competitive Advantage. *Harvard Business Review* (May–June 2017), 96–103.

[7] Petra S Barneveld, Hanna Swaab, Selene Fagel, Herman Van Engeland, and Leo MJ de Sonneville. 2014. Quality of life: A case-controlled long-term follow-up study, comparing young high-functioning adults with autism spectrum disorders with adults with other psychiatric disorders diagnosed in childhood. *Comprehensive psychiatry* 55, 2 (2014), 302–310.

[8] Andrew Begel, James Dominic, Conner Phillis, Thomas Beeson, and Paige Rodeghero. 2021. How a Remote Video Game Coding Camp Improved Autistic College Students' Self-Efficacy in Communication. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*. 142–148.

[9] Andrew Begel and Amy J. Ko. 2019. Learning Outside the Classroom. In *The Cambridge Handbook of Computing Education Research*, Sally A. Fincher and Anthony V. Robins (Eds.). Cambridge University Press, Cambridge, United Kingdom, Chapter 26, 749–772. https://doi.org/10.1017/9781108654555

[10] Andrew Begel and Beth Simon. 2008. Novice Software Developers, All over Again. In *Proceedings of the Fourth International Workshop on Computing Education Research* (Sydney, Australia) *(ICER '08)*. ACM, New York, NY, USA, 3–14. https://doi.org/10.1145/1404520.1404522

[11] Kathleen A Boothe, Marla J Lohmann, Kimberly A Donnell, and D Dean Hall. 2018. Applying the principles of universal design for learning (UDL) in the college classroom. *Journal of Special Education Apprenticeship* 7, 3 (2018), n3.

[12] Brendan Borrell. 2018. How colleges can prepare for students with autism. https://www.spectrumnews.org/features/deep-dive/colleges-can-prepare-students-autism.

[13] LouAnne Boyd, Christina McReynolds, and Karen Chanin. 2013. *The Social Compass Curriculum*. Paul H. Brookes Publishing Co., Baltimore, MD.

[14] Moira Burke, Robert Kraut, and Diane Williams. 2010. Social Use of Computer-Mediated Communication by Adults on the Autism Spectrum. In *Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work* (Savannah, Georgia, USA) *(CSCW '10)*. Association for Computing Machinery, New York, NY, USA, 425–434. https://doi.org/10.1145/1718918.1718991

[15] Fabio Calefato and Christof Ebert. 2019. Agile collaboration for distributed teams [software technology]. *IEEE Software* 36, 1 (2019), 72–78.

[16] Orges Cico, Letizia Jaccheri, Anh Nguyen-Duc, and He Zhang. 2021. Exploring the intersection between software industry and Software Engineering education - A systematic mapping of Software Engineering Trends. *Journal of Systems and Software* 172 (2021), 110736. https://doi.org/10.1016/j.jss.2020.110736

[17] Carrie Anna Courtad. 2019. Making your classroom smart: Universal design for learning and technology. In *Smart Education and e-Learning 2019*. Springer, 501–510.

[18] Michelle Craig, Phill Conrad, Dylan Lynch, Natasha Lee, and Laura Anthony. 2018. Listening to Early Career Software Developers. *J. Comput. Sci. Coll.* 33, 4 (April 2018), 138–149.

[19] Siva Dorairaj and James Noble. 2013. Agile Software Development with Distributed Teams: Agility, Distribution and Trust. In *2013 Agile Conference*. 1–10. https://doi.org/10.1109/AGILE.2013.7

[20] William Easley. 2018. Understanding The Experiences of Youth In Collaborative Technical Workplaces. In *Companion of the 2018 ACM Conference on Computer Supported Cooperative Work and Social Computing* (Jersey City, NJ, USA) *(CSCW '18)*. Association for Computing Machinery, New York, NY, USA, 113–116. https://doi.org/10.1145/3272973.3272987

[21] Auksė Endriulaitienė and Lina Cirtautienė. 2021. Team effectiveness in software development: the role of personality and work factors. *Verslas: teorija ir praktika= Business: theory & practice. Vilnius: VGTU leidykla, 2021, Vol. 22, iss. 1* (2021).

[22] R Ferdig, C Cavanaugh, and J Freidhoff. 2012. Lessons learned from blended programs: Experiences and recommendations from the field. *Vienna, VA: iNACOL* (2012).

[23] Ilenia Fronza and Xiaofeng Wang. 2021. Social loafing prevention in agile software development teams using team expectations agreements. *IET Software* 15, 3 (2021), 214–229.

[24] Kristen Gillespie-Lynch, Steven K. Kapp, Christina Shane-Simpson, David Shane Smith, and Ted Hutman. 2014. Intersections Between the Autism Spectrum and the Internet: Perceived Benefits and Preferred Functions of Computer-Mediated Communication. *Intellectual and Developmental Disabilities* 52, 6 (12 2014), 456–469. https://doi.org/10.1352/1934-9556-52.6.456 arXiv:https://meridian.allenpress.com/idd/article-pdf/52/6/456/2042724/1934-9556-52_6_456.pdf

[25] Lisa R Halverson and Charles R Graham. 2019. Learner engagement in blended learning environments: A conceptual framework. *Online Learning* 23, 2 (2019), 145–178.

[26] Kyleigh B Harrell and Jillian L Wendt. 2019. The impact of blended learning on community of inquiry and perceived learning among high school learners enrolled in a public charter school. *Journal of Research on Technology in Education* 51, 3 (2019), 259–272.

[27] Juliet E Hart and Kelly J Whalon. 2008. Promote academic engagement and communication of students with autism spectrum disorder in inclusive settings. *Intervention in School and Clinic* 44, 2 (2008), 116–120.

[28] Angela Haslam. 2013. Literacy Support For Children With Autism. (2013).

[29] Darren Hedley, Mirko Uljarević, Lauren Cameron, Santoshi Halder, Amanda Richdale, and Cheryl Dissanayake. 2017. Employment

programmes and interventions targeting adults with autism spectrum disorder: A systematic review of the literature. *Autism* 21, 8 (2017), 929–941. https://doi.org/10.1177/1362361316661855 arXiv:https://doi.org/10.1177/1362361316661855 PMID: 27542395.

[30] Anita Hidayati, Eko K Budiardjo, and Betty Purwandari. 2020. Hard and soft skills for scrum global software development teams. In *Proceedings of the 3rd International Conference on Software Engineering and Information Management*. 110–114.

[31] Pamela J Hinds and Diane E Bailey. 2003. Out of sight, out of sync: Understanding conflict in distributed teams. *Organization science* 14, 6 (2003), 615–632.

[32] Nicky Hockly. 2018. Blended learning. *Elt Journal* 72, 1 (2018), 97–101.

[33] Terrell Holmes and Hala Annabi. 2020. The dark side of software development: job stress amongst autistic software developers. In *Proceedings of the 53rd Hawaii International Conference on System Sciences*.

[34] Andre Hora. 2021. Googling for Software Development: What Developers Search For and What They Find. In *International Conference on Mining Software Repositories*. 1–12.

[35] Asim Iftikhar, Muhammad Alam, Shahrulniza Musa, and Mazliham Mohd Su'ud. 2017. Trust Development in virtual teams to implement global software development (GSD): A structured approach to overcome communication barriers. In *2017 IEEE 3rd International Conference on Engineering Technologies and Social Sciences (ICETSS)*. IEEE, 1–5.

[36] Ritu Jain and Ugrasen Suman. 2017. An Adaptive Agile Process Model for Global Software Development. *Int. J. Comput. Sci. Eng* 9, 06 (2017), 436–445.

[37] Murat Dogus Kahya and Çağla Seneler. 2018. Geographical distance challenges in distributed agile software development: Case study of a global company. In *2018 3rd International Conference on Computer Science and Engineering (UBMK)*. IEEE, 78–83.

[38] Jayden Khakurel and Jari Porras. 2020. The Effect of Real-World Capstone Project in an Acquisition of Soft Skills among Software Engineering Students. In *2020 IEEE 32nd Conference on Software Engineering Education and Training (CSEE&T)*. IEEE, 1–9.

[39] Youry Khmelevsky, Xitong Li, and Stuart Madnick. 2017. Software development using agile and scrum in distributed teams. In *2017 Annual IEEE International Systems Conference (SysCon)*. IEEE, 1–4.

[40] Andrew J Ko and Katie Davis. 2017. Computing mentorship in a software boomtown: Relationships to adolescent interest and beliefs. In *Proceedings of the 2017 ACM Conference on International Computing Education Research*. 236–244.

[41] Elinda Ai Lim Lee, Melissa H Black, Tele Tan, Torbjorn Falkmer, and Sonya Girdler. 2019. "I'm destined to ace this": Work experience placement during high school for individuals with autism spectrum disorder. *Journal of autism and developmental disorders* 49, 8 (2019), 3089–3101.

[42] Chen Li, Emily Chan, Paul Denny, Andrew Luxton-Reilly, and Ewan Tempero. 2019. Towards a framework for teaching debugging. In *Proceedings of the Twenty-First Australasian Computing Education Conference*. 79–86.

[43] M.J. Maenner, K.A. Shaw, and J. Baio. 2020. Prevalence of Autism Spectrum Disorder Among Children Aged 8 Years — Autism and Developmental Disabilities Monitoring Network, 11 Sites, United States, 2016. *MMWR Surveill Summ 2020* 69, SS-4 (2020), 1–12.

[44] Margie Martyn. 2003. The hybrid online model: Good practice. *Educause Quarterly* 26, 1 (2003), 18–23.

[45] Micah O Mazurek, Paul T Shattuck, Mary Wagner, and Benjamin P Cooper. 2012. Prevalence and correlates of screen-based media use among youths with autism spectrum disorders. *Journal of autism and developmental disorders* 42, 8 (2012), 1757–1767.

[46] Jane D McLeod, Emily Meanwell, and Amelia Hawbaker. 2019. The experiences of college students on the autism spectrum: A comparison to their neurotypical peers. *Journal of autism and developmental disorders* 49, 6 (2019), 2320–2336.

[47] Courtney Miller, Paige Rodeghero, Margaret-Anne Storey, Denae Ford, and Thomas Zimmermann. 2021. "How Was Your Weekend?" Software Development Teams Working From Home During COVID-19. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*. IEEE, 624–636.

[48] Meredith Ringel Morris, Andrew Begel, and Ben Wiedermann. 2015. Understanding the Challenges Faced by Neurodiverse Software Engineering Employees: Towards a More Inclusive and Productive Technical Workforce. In *Proceedings of the 17th International ACM SIGACCESS Conference on Computers and Accessibility* (Lisbon, Portugal) *(ASSETS '15)*. Association for Computing Machinery, New York, NY, USA, 173–184. https://doi.org/10.1145/2700648.2809841

[49] Makayla Moster, Ella Kokinda, Matthew Re, James Dominic, Jason Lehmann, Andrew Begel, and Paige Rodeghero. 2022. Lesson Materials: "Can You Help Me?" An Experience Report of Teamwork in a Game Coding Camp for Autistic High School Students. https://doi.org/10.5281/zenodo.5902445

[50] Alex Radermacher, Gursimran Walia, and Dean Knudson. 2014. Investigating the Skill Gap between Graduating Students and Industry Expectations. In *Companion Proceedings of the 36th International Conference on Software Engineering* (Hyderabad, India) *(ICSE Companion 2014)*. Association for Computing Machinery, New York, NY, USA, 291–300. https://doi.org/10.1145/2591062.2591159

[51] Annelies Raes, Loulou Detienne, Ine Windey, and Fien Depaepe. 2020. A systematic literature review on synchronous hybrid learning: Gaps identified. *Learning Environments Research* 23, 3 (2020), 269–290.

[52] Claudia Raibulet and Francesca Arcelli Fontana. 2018. Collaborative and teamwork software development in an undergraduate software engineering course. *Journal of Systems and Software* 144 (2018), 409–422.

[53] Christina Rebholz. 2012. *Life in the uncanny valley: Workplace issues for knowledge workers on the autism spectrum*. Ph. D. Dissertation. Antioch University, Seattle, WA.

[54] Ariana Riccio, Maruf Hossain, Beth Rosenberg, and Kristen Gillespie-Lynch. 2018. A Participatory Evaluation of an Innovative Technology-Based Program for Adolescents with Autism. In *Connected Learning Summit*. Cambridge, MA.

[55] Kathryn E. Ringland, Christine T. Wolf, LouAnne E. Boyd, Mark S. Baldwin, and Gillian R. Hayes. 2016. Would You Be Mine: Appropriating Minecraft as an Assistive Technology for Youth with Autism. In *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility* (Reno, Nevada, USA) *(ASSETS '16)*. Association for Computing Machinery, New York, NY, USA, 33–41. https://doi.org/10.1145/2982142.2982172

[56] Beth Rosenberg. 2016. Tech Kids Unlimited. In *2016 Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*. IEEE Computer Society, 1–2.

[57] Anne M Roux, Paul T Shattuck, Benjamin P Cooper, Kristy A Anderson, Mary Wagner, and Sarah C Narendorf. 2013. Postsecondary employment experiences among young adults with an autism spectrum disorder. *Journal of the American Academy of Child & Adolescent Psychiatry* 52, 9 (2013), 931–939.

[58] Marlene Scardamalia and Carl Bereiter. 1993. Computer Support for Knowledge-Building Communities. *The Journal of the Learning Sciences* 3, 3 (1993), 265–283. http://www.jstor.org/stable/1466822

[59] K.A. Shaw, M.J. Maenner, and J. Baio. 2020. Early Identification of Autism Spectrum Disorder Among Children Aged 4 Years — Early Autism and Developmental Disabilities Monitoring Network, Six Sites, United States, 2016. *MMWR Surveill Summ 2020* 69, SS-3 (2020), 1–11.

[60] Jefferson Silva, Igor Wiese, Daniel M. German, Christoph Treude, Marco Aurélio Gerosa, and Igor Steinmacher. 2020. A Theory of the Engagement in Open Source Projects via Summer of Code Programs. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (Virtual Event, USA) *(ESEC/FSE 2020)*. Association for Computing Machinery, New York, NY, USA, 421–431. https://doi.org/10.1145/3368089.3409724

[61] Manuel Stadler, Raoul Vallon, Martin Pazderka, and Thomas Grechenig. 2019. Agile distributed software development in nine central European teams: Challenges, benefits, and recommendations. *International Journal of Computer Science & Information Technology (IJCSIT) Vol* 11 (2019).

[62] Sander Valstar. 2019. Closing the Academia-Industry Gap in Undergraduate CS. In *Proceedings of the 2019 ACM Conference on International Computing Education Research* (Toronto ON, Canada) *(ICER '19)*. Association for Computing Machinery, New York, NY, USA, 357–358. https://doi.org/10.1145/3291279.3339440

[63] Heidi C. Webb and Mary Beth Rosson. 2011. Exploring Careers While Learning Alice 3D: A Summer Camp for Middle School Girls. In *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education* (Dallas, TX, USA) *(SIGCSE '11)*. Association for Computing Machinery, New York, NY, USA, 377–382. https://doi.org/10.1145/1953163.1953275

[64] Jun Xiao, Hong-Zheng Sun-Lin, Tzu-Han Lin, Mengyuan Li, Zhimin Pan, and Hsu-Chen Cheng. 2020. What makes learners a good fit for hybrid learning? Learning competences as predictors of experience and satisfaction in hybrid learning space. *British Journal of Educational Technology* 51, 4 (2020), 1203–1219.

[65] Baichang Zhong, Qiyun Wang, Jie Chen, and Yi Li. 2017. Investigating the period of switching roles in pair programming in a primary school. *Journal of Educational Technology & Society* 20, 3 (2017), 220–233.