How a Remote Video Game Coding Camp Improved Autistic College Students' Self-Efficacy in Communication

Andrew Begel
Microsoft Research
andrew.begel@microsoft.com

James Dominic
Clemson University
domini4@clemson.edu

Conner Phillis
KeyMark, Inc.
conner.phillis@keymarkinc.com

Thomas Beeson
Clemson University
tbeeson@clemson.edu

Paige Rodeghero
Clemson University
prodegh@clemson.edu

ABSTRACT

Communication and teamwork are essential skills for software developers. However, these skills are often difficult to learn for students with autism spectrum disorder (ASD). We designed, developed, and ran a 13-day, remote video game coding camp for incoming college first-year students with ASD. We developed instructional materials to teach computer programming, video game design, and communication and teaming skills. Students used the MakeCode Arcade development environment to build their games and Zoom to remotely collaborate with their teammates. In summative interviews, students reported improved programming skills, increased confidence in communication, and better experiences working with others. We also found that students valued the opportunity to practice teaming, such as being more vocal in expressing ideas to their peers and working out differences of opinion with their teammates. Two students reported the remote learning environment decreased their anxiety and stress, both are frequent challenges for autistic people. We plan to rerun the camp next year with materials that we have made available online.

CCS CONCEPTS

• **Human-centered computing → Accessibility design and evaluation methods**; • **Social and professional topics → Informal education**.

KEYWORDS

video games, autism, coding camp

ACM Reference Format:
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 (SIGCSE '21), March 13–20, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3408877.3432516

SIGCSE '21, March 13–20, 2021, Virtual Event, USA

ACM ISBN 978-1-4503-8062-1/21/03…$15.00
https://doi.org/10.1145/3408877.3432516

1 INTRODUCTION

Autism spectrum disorder (ASD) is a life-long neuro-developmental condition that has a significant impact on an autistic person's daily life.[1] It is often characterized by particular cognitive styles, communication behaviors, social interactions, and repetitive behaviors that vary widely across autistic people but are often conspicuous to neurotypical (i.e., non-autistic) people [3]. As of 2016, the CDC reports that 1 in 54 children in the USA are diagnosed with ASD [4]. Only 17% of autistic people enroll in four-year colleges and only 39% of those graduate [7].

Due to COVID-19, many universities transitioned courses from in-person to online. This change to online education may be especially difficult for those incoming first-year students who are yet to build relationships with their peers. It is even more difficult for those with autism who may struggle to build friendships face-to-face. Fortunately, some universities offer summer online courses that provide incoming autistic first years with an opportunity to get a taste of university studies and provide an opportunity to form relationships with their peers.

During Summer 2020, in collaboration with Clemson University's Spectrum Program, we developed a remote game software development camp. This 13-day camp met every weekday afternoon on Zoom. The instructors included a CS professor, an autism researcher, an ABA therapist and coordinator of the Spectrum Program, a CS PhD student, and a professional software engineer. Our goal was to introduce and teach game design and development to the incoming first-year autistic students. Students learned how to build PacMan and worked in teams to design and develop their own game using MakeCode Arcade. The camp specifically emphasized communication, and to that end, provided extensive scaffolding to help the students strengthen collaboration skills, which are essential for college courses and software development. After the camp, we interviewed the students and found that the opportunities we provided for students to practice communication and collaboration were recognized and appreciated. The students reported that communication was their most-improved skill, which validates our approach. In addition, several pedagogical techniques we employed helped students feel less anxious and stressed, two challenges that frequently impact autistic people.

We plan to rerun the camp next year and offer our teaching materials online at https://doi.org/10.6084/m9.figshare.12890630.v1.

[1] It is important to note that in the autism community, some people prefer people-first language. However, there are others who have embraced the term "autistic" as their preferred label. In this paper, we use this latter terminology [16].

2 RELATED WORK

Autistic students can often find it challenging to adapt to university instruction due to the reduction in structure and support employed in secondary school education. University faculty members can find it difficult to connect with and teach their autistic students without support from specialists. Egan spread awareness about the challenges autistic people and professors encounter in computer science courses [11] and discussed challenges with social interactions, struggling with change, motor skill issues, attention, and emotional issues. She provides a guide to address some of these issues. Ribu's literature review on psychology and sociology studies on autistic people [21] reveals her experiences teaching autistic students and her suggestions for pedagogical techniques. We employed remote pair programming as a central pedagogy for the students' group project, which was intended to help them practice communication, collaboration, and teamwork skills [9].

Autistic students are often said to be attracted to software development because of the repetition and precision that are required [10]. Autistic developers have many talents that are helpful for writing and maintaining source code [2]. Even though many autistic people struggle with communication, they bring out-of-the-box thinking to their teams. They can be extremely detailed-oriented, focused, and committed to providing high-quality work [14, 15]. Unfortunately, some autistic people have high levels of anxiety that hinder communication and collaboration [6, 17, 24]. Zolyomi *et al.* also studied the stress that autistic people feel when communicating with others via video conferencing, a challenge of particular relevance to our camp [30]. From this work, we applied techniques to reduce our students' anxiety and cognitive load.

We found this attraction to coding among the students in our camp, and in particular, an interest in video games. Recently, Tsikinas and Xinogalos studied the effects of serious games and found them to be promising for learning for autistic people [26]. Additionally, Bossavit and Parsons studied two autistic teenagers collaborating to develop a collaborative game [8]. Unfortunately, their students did not collaborate and ended up building a collaborative game that required no player-player collaboration. The teenagers primarily interacted with the adult instructors, not one another.

Software engineering careers demand strong teamwork skills, which can suffer due to poor communication and collaboration [25]. Zolyomi *et al.* interviewed autistic team members (both students and employees) to understand their teamwork technology needs [31]. They found that increased Social Translucence [13] could help autistic people become more aware and accountable to a team's social cues. Eiselt and Carter found that programming coursework can assist in strengthening social skills and computational thinking skills for autistic students [12]. In addition, Munoz *et al.* used game programming as a motivation to assist autistic adolescents in the development of computational thinking skills [19]. Yakubova and Taber-Doughty found that programming can help in strengthening autistic students' problem-solving skills, another essential set of skills for software developers [29]. Furthermore, we are starting to see the development of tools to assist autistic developers and teach autistic children programming. For example, Zubair *et al.* designed and developed visual representations of programming concepts to teach autistic children with cognitive impairments [32].

Outside of schools, many organizations offer coding camps in the summer months for children and teenagers [5]. Webb and Rosson ran a week-long coding camp with Alice to encourage women to tell stories through computer programming [28]. Adams ran coding camps for middle school girls using block-based programming (Alice 2.0 and Scratch) [1]. He discussed the potential for using block-based programming for the outreach of those with disabilities. Others have also used block-based coding for outreach and teaching those with disabilities [20]. We have not found any reports of remote coding camps specifically targeted at autistic students.

3 METHODOLOGY

In this section, we discuss our camp design methodology, pedagogy, research study methodology, and introduce the students.

3.1 Camp Design

Our camp design was inspired by the authors' previous experience with game programming, Ian Schreiber's blog on game programming [22], and a consultation with Dr. Paul Gestiwicki, a computer science professor focusing on game development and design at Ball State U. We taught students video game development using MakeCode Arcade [18], a block-based programming environment. We piloted the first two days of our syllabus with a neurotypical high school student and used that feedback to tweak our instructional design before starting the camp.

3.1.1 Technical Content. We had two technical learning goals for the camp. First, students should be able to understand the game design process and game structures, including game design, character archetypes, storylines, narratives, goals, rules, game mechanics, and artificial intelligence. Second, students should be able to design and develop their own video games from scratch.

To accomplish these goals, the camp was divided into two sections: the first was devoted to instruction on the foundations of game design, and the second to a team project for students to create their own games. We played two games and walked the students through their story, game play, mechanics, and rules. We also introduced different game genres. Next, we devoted three days to demonstrate step-by-step how to build PacMan in MakeCode Arcade.[2] We also deconstructed a pre-existing game from the MakeCode Arcade home page, discussing how game design elements are implemented. Next, we taught paper prototyping and had students design prototypes for a hypothetical music application. We invited guest speakers to lecture on game-relevant enrichment activities, such as networking, 3D animation, and the ethics of in-app game purchases. In addition, two video gaming experts were invited to speak on maker activities and to demonstrate a wide variety of MakeCode Arcade video games. Over the last seven days of the camp, students worked in teams on a final project to design and develop their own games.

3.1.2 Communication Goals. We tailored the camp to address the communication needs of autistic students, as described by Zolyomi *et al.* [31]. With a programming curriculum aimed at teamwork, we followed Tuckman and Jensen's team formation model: forming,

[2]Incidentally, we did not anticipate how difficult PacMan is to code. Ghost behaviors are much more complicated than they first appear.

storming, norming, and performing [27]. Since our camp was so short, we eliminated the forming phase, and simply assigned students to teams on Day 2. To enhance social connection and minimize the anxiety that comes from working with new people, we kept student teams fixed throughout the 13-day camp. Since these autistic students were new to collaboration and teamwork, we scaffolded the storming and norming phases. First, student teams worked through a number of prescribed small-group, think/pair/share exercises, which required minimal negotiation other than sharing ideas and deciding on a sharing speaker. Next, when teams began to work on their project, each team member was asked to interview the others according to an interview script (available in our replication package). This helped them to get to know one another socially, outside of the context of game design and development. A second scaffolded interview between students was added to help students establish good working styles and norms.

Our communication goals were designed to help students practice social skills and teamwork, which are vital for computer science careers and often challenging for autistic people. These goals included how to communicate and explain your actions to your teammates, how to collaborate on a team project, and how to reflect on your interactions with other students. Students worked together in breakout rooms for 2–3 hours per day, dividing their work into 10-minute sessions interrupted by a scaffolded standup meeting. Students also practiced driver-navigator pair programming methods, which encouraged frequent communication and made collaboration easier to observe. In partial support of developing their teamwork skills, we encouraged the students to check in with one another periodically since more attention shown to the partner will make them feel more welcome and included. In addition, we advised our students to communicate proactively, as often as they could stand it.

On the last day of camp, student teams presented their games to the other students, instructors, parents, and several university representatives. This raised the stakes for the students because their project would have an audience who would judge them for what they presented. We scaffolded the project presentations to have students introduce themselves, give an elevator pitch for their game (which they had come up with at the beginning of the project phase), and demo their game while a teammate narrated. Additional prompts were provided to students in advance to provide a structure to their personal reflections on their project and the coding camp. Finally, after each team presented their game, to help students learn to give and receive criticism, all of the other students were required to say one nice thing they liked about the game and offer one suggestion to improve it.

3.2 Pedagogy

The initial camp design began in February of 2020. We planed to conduct the camp in person; however, due to COVID-19's emergence in America in March, we changed the camp to use synchronous online instruction. We chose to use Zoom's teleconferencing software because it provided us the ability to make breakout rooms where each team could collaborate together on their project. It also supported A/V-recording and transcription of each session (including breakout rooms), which made it possible for us to study the impact of our pedagogy on the students.

Next, we chose the programming language from a corpus of 14 game programming environments. Each of us chose two environments in which to program a simple arcade game and then evaluated how difficult it was for us and how difficult it might be for students to create their own games. We ended up choosing between Scratch and MakeCode Arcade, which have similar functionality for block-based programming environments. Ultimately, we chose MakeCode Arcade, which is a free, open-source, extensible, web-based computing education platform from Microsoft [23] because of its deep integration with GitHub would better support team collaboration, which was a core component of our camp. Initially, we were a bit worried that the students, especially those that had taken AP Computer Science, might think a block-based programming environment was beneath them, but almost everyone engaged wholeheartedly during camp. We did learn on the last day that 3 students would have preferred to use a more complex text-based programming language for game development.

During the 13 days of camp, we employed a variety of pedagogical techniques to facilitate learning, provide structure to activities, and provide formative assessment. During the first section of the camp, our primary focus was on education: introducing the students to game design and structure, providing an overview of the MakeCode platform, and paper prototyping. Direct instruction with active learning was utilized to introduce content and an "I do it/You do it/Let's talk about it" process employed to support hands-on learning in a remote environment. This process was used to divide up direct instruction into 10-minute increments, enabling us to maintain student attention in an online camp more easily.

For the next three days, we modeled the process of designing a PacMan-style arcade game through step-by-step instruction. After each step, in which students were shown how to write some code, students were instructed to write similar code (with some opportunity to make enhancements). Sample code was provided for students who fell behind or missed some of the instructional time. Initially, we intended to have each student demonstrate their work after each step, but found they kept up easily, making the sharing unnecessary. We also taught the students paper prototyping to teach meta-cognitive skills, such as planning before coding, which is an important skill for software development.

The second section of the camp was devoted to student projects. Two instructors brainstormed in front of the students to come up with ideas for a game, modeling the instructive and reflective processes we hoped the students would pick up in order to conceive of their own game. We also provided a scaffold to help students do their own game design brainstorming.

Students were given time during every day of the camp to work together on their projects. However, due to lack of support for simultaneous editing in MakeCode, we asked students to take turns working either in the role of driver (i.e., controlling the MakeCode environment) or that of the navigator (i.e., telling the driver what to do). Students were instructed to switch roles about every 10 minutes, which provided a structure for the team sessions, equalized workload, provided opportunities to assess and solidify progress, afforded possibilities for communication, and promoted engagement.

Each handoff was accompanied by a standup meeting following a Scrum pattern from Agile methods, in which each student talked about what they had been working on, what they wanted to work on

Table 1: Student Demographics. All names have been converted to pseudonyms to protect identity. Bob's teammate did not consent for his data to be used in this report.

Alias	Team	Gender	CS Classes	PL Exp.
Luke	A	Male	None	Minecraft
Alice	A	Female	AP CS, Foundations	Java, JS, Scratch
Bob	B	Male	AP	C++, Java, JS
Katie	C	Female	AP CS	Java, JS, HTML
Scott	C	Male	AP CS, Camp	Java, JS, HTML
Martha	C	Female	Self-taught	HTML

next, and what they were stuck on. In order to technically facilitate each driver-navigator handoff, we introduced the concept of source code control. We set up a GitHub Classroom to enable each team to clone their own game repository and simultaneously set project permissions to enable them (and us) to view and edit the project. The students were asked to upload their projects to GitHub when they switched drivers and at the end of each day.

Finally, the instructors helped students look for good stopping opportunities to end each day, usually after finishing a subtask. This helped students to see how to plan a complete workday. Ultimately, each student team decided for themselves when they would stop, but appreciated the value of ending on a high note each day.

After each day, the instructors discussed each student's progress. In the first week of the camp, as we were getting to know each student as individuals, we often simply shared observations of their behavior. In the second half of the camp, while students were working on team projects, we discussed successful and problematic communication and collaboration behaviors and suggested interventions to address student challenges.

3.3 Students

Seven autistic, university-level, rising first-years enrolled in this camp due to their interest in game design and development. Figure 1 presents the demographics of the students along with their reported backgrounds with computer programming.[3] Half were women, which we find significant, given that the diagnosis rate for autism is 4:1 male to female [4]. At the end of the camp, 4 students expressed an interest in becoming video game designers, while Scott wanted to be a computer scientist and Luke wanted to be an inventor.

3.4 Research Study Methodology

We designed this camp to help us understand the impact of explicit communication learning goals in a programming camp for autistic students. Our study protocol was approved by Clemson's IRB. All survey and interview questions can be found in our teaching materials package online.

To help us dial in an appropriate difficulty level for the camp, we surveyed students' computing experiences before camp began. During the camp, we asked students to fill out daily surveys after each camp session. These consisted of 15 true/false questions related to social collaboration, followed by a few open-ended questions asking them to reflect on that day's syllabus topics. The short answer questions enabled us to identify students' collaboration experiences over the 13 days of the camp. In addition, for 30 minutes prior to and after each day's sessions, the instructors met to discuss the progress of each student and team. On the final day of the camp, we interviewed all of the students individually and together to learn about their impressions of the camp experience, its impact on their perception of computer science, gaming, social skills, and their future careers, and to gather suggestions for next year's camp. All camp sessions were audio/video-recorded and transcribed.

For the findings we report in this paper, we conducted a qualitative analysis of the daily survey long answers, daily instructor observations, and final student interviews. All of the instructors coded the text into one or more of our three communication goals: communication, collaboration, and reflection. In addition, we created a quantitative measure of collaboration from our daily survey true/false questions, which we call the Social Collaboration Index (SCI). The SCI is the sum of true answers to these questions for each student for each day of the camp. See Section 4.2 and Figure 1 for a discussion of the SCI scores for each student team.

4 FINDINGS

In this section, we discuss findings from audio/video recordings, daily surveys, daily reflections, and our summative interviews. They are organized around our communication learning goals, since these are the most salient factors to evaluate the coding camp's effects on communication self-efficacy for autistic students.

4.1 Communication

From the first day of the camp, we set high expectations for social interaction, both with the instructors and among the students.

Team A: Luke and Alice's team worked well together and were good communicators from Day 1. Luke would get Alice's attention by saying, "Look, I need help," and she would respond immediately. They bounced ideas back and forth, easily able to settle on one together, no matter from whom the idea originated. Although Luke initially expressed a desire to impose his own ideas on his team, he learned to relax. "I . . . forced myself to . . . make time to let . . . my group member speak and . . . made myself . . . go against my better nature, which would be would be to talk the entire time."

Team B: At the outset, we were worried about Bob's team because they were particularly quiet. If not prompted by the instructor, they would just sit silently without doing anything together. Later on, Bob learned to narrate what he was thinking while he worked, which facilitated their social interaction. Bob continued to narrate to his teammate even when the code became too complex to explain. This had the effect of maintaining his teammate's attention on their shared work, keeping them in sync.

Team C: On the third team, Scott often spoke off-topic, but was repeatedly encouraged by his teammates, Katie and Martha, to contribute to whatever discussion they were having at the time. Katie had particularly good skills in verbally expressing her vision of the project while she was playing the navigator role in her interactions with her team, even though it can be difficult to verbally direct a driver to control something as visual as a painting program. We felt that sometimes Katie could dominate her team's discussion. We attempted to facilitate a more balanced approach in which she would moderate how much she spoke relative to the others

[3] Only 6 out of 7 students consented to have their data reported in this paper.

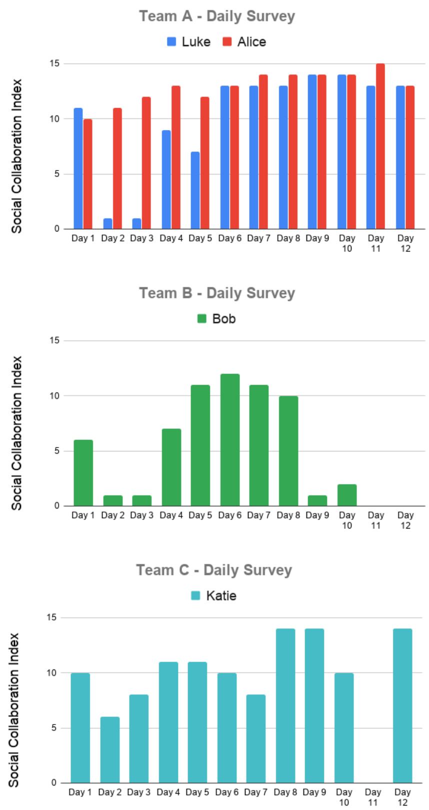

Figure 1: Social Collaboration Index for each student team over 12 days. Martha and Scott (Team C) did not complete the survey, and Bob's teammate in Team B did not give consent. Bob did not fill out the survey on Day 11 and was absent on Day 12. Katie was absent on Day 11.

on her team, but she resisted. In the latter days of the camp, she acknowledged to us that she had a tendency to take charge and issue commands to others in team projects and that she was "definitely trying to let other people have input and not just keep talking."

We felt that most of the students improved their communication skills over the 13 days of the camp. At our summative interview, 5 of 6 students reported that their most improved skill was communication. Luke said, "I never really tried to be in a group where I'd like I've been intentional about like the way I talk. I was kind of just kind of like I had no filter. And it [was] really intense to put a filter on myself." Alice reflected, "Throughout this course, I've been . . . speaking more, I think, than I usually do, and I'm talking a lot with my teammate to work out things."

4.2 Collaboration

Collaboration is an essential skill for future computer science students to learn to succeed in a career in programming.

Team A: Luke and Alice's team collaborated really well together. Whenever Luke would work on something outside of camp, he took the time to explain what he did to Alice before together incorporating it into the project. However, sometimes Luke would overthink something he worked on (he was often tripped up by perfectionism) and Alice would bring him back to reality. Luke and Alice's team regularly swapped the driver role while pair programming, though if Luke ever took over the driver's seat too often, Alice was not shy about reminding him to release control. Even though Luke was faster at manipulating the programming environment UI than Alice, he demonstrated enough patience when she was driving to wait for her to finish working. Similarly, even if Alice had concluded to herself that Luke's ideas were unworkable, she let him try his ideas so he could see for himself that they would not bear fruit.

Team B: Bob's team's collaboration was uneven. Bob took on a leadership role, often directing the team's efforts. As their project complexity increased, their team split into two roles. Bob focused on the implementation details while his partner remained attentive to Bob's verbal externalization of his thought process. Despite the asymmetry, Bob and his teammate split the driving and navigation roles evenly, swapping roles every 10 minutes.

Team C: For the most part, Katie, Martha, and Scott's team ignored our scaffolding and resisted our imposition of a periodic standup meeting. Their collaboration style was also uneven, with Katie often directing the group's activity, while Martha or Scott did the driving. Throughout the project, Scott appeared reluctant to participate meaningfully with his two teammates. Martha and Katie were often in sync with one another, leaving Scott out, except those rare times he contributed an idea. However, Scott did find his passion when choosing sprites for each character in the game. This enthusiasm sparked a great brainstorming conversation among the entire team that led to their main project concept.

To quantify the effects of our communication skills pedagogy on the students, we calculated the Social Collaboration Index (SCI), which is shown in Figure 1. It is possible to see relatively good collaboration scores (generally over 10 out of 15) in Team A, especially during the project phase. Bob started out with good collaboration scores, but as his project proceeded to increase in complexity, his communication with his partner became more one-directional, leaving less opportunity for meaningful collaboration. Katie reported positive collaboration during the project phase, but without her two teammates reporting, we are unable to corroborate her perception.

4.3 Reflection

Metacognitive skills such as reflection are often difficult for students to learn, especially without explicit modeling. As each team worked, they exhibited some early reflection abilities. We expect these will develop further as they progress through their time in university.

Team A: At times, Luke could be very particular about how something was to be implemented. If he was not driving, he would micromanage Alice to do things exactly as he wanted. However, later on, he appeared to realize that this upset Alice, and would walk back his demands and ask her what she thought about the idea. This would give Alice the opportunity to contribute her own thoughts (which always turned out to be valuable). Towards the end of the final week, we reminded Luke and Alice that they needed to finalize their game's implementation prior to the project presentation. They worked together to identify missing implementation details (e.g. scoring) that needed to be added, as well as find things to skip.

Team B: Often, once Bob got an idea in his head, it was impossible to get him to pause and reflect on the best way forward.

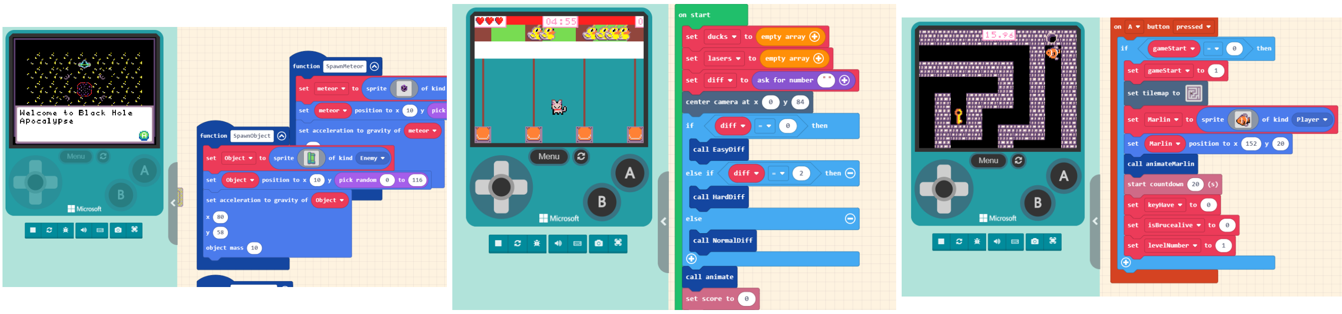

Figure 2: Screen snapshots of MakeCode Arcade video games developed by student teams.

As instructors, we felt frustrated that our camp's remote setting reduced our abilities to interrupt students to provide guidance.

Team C: Each student in Team C divided up the project by topic: e.g., collisions, sprites, and maze levels. We asked them to record their individual thoughts in a shared Google Doc, but they again preferred to limit note-taking and convey all their plans verbally. We intervened with the team to model a good planning process for the final presentation. Scott had designed the game's maze, but neglected to indicate where in the maze the 'keys' should be. We made a deliberate effort to have Katie ask Scott where he intended to put the keys instead of letting Katie choose their location. Scott was glad to be asked because he had stopped paying attention, and used that opportunity to reengage with the team. When Katie was absent one day, Martha focused on the project, but Scott failed to pay attention. We interrupted and asked them "What are we doing now? Let's discuss a plan before we continue." This got them to focus on planning, but it only worked for a short time. As they continued swapping driver-navigator roles, they often worked alone, without explaining to the other what they were doing.

4.4 Communication Self-Efficacy

After the final presentations, we asked students what they learned. About his presentation, Luke said, "I love the experience. . . . I love expressing ideas and stuff. And so I especially enjoy just talking about something with never really [having] done something like this before." His confidence in his CS abilities increased as well, "This kind of gave me the confidence to be able to know that I can potentially do stuff like this in the future." Bob noted he learned a lesson about teamwork, "I have like all my head be able to work that out and make like to pass on knowledge to others and you know help get everybody on the same page." Alice said about working out problems, "I feel like . . . you gotta stay calm when you're working out a problem because . . . if I get worried that if I mess something up . . . figuring out bugs helped me kind of like work around that and . . . keep a level head." Martha experienced the most transformation. She had always been shy and reluctant to work with others. However, she learned that "people actually do want to hear my ideas. They do want to hear my input, and sometimes they do think it's good, and they do think it's constructive."

5 DISCUSSION

We found that the students were all highly motivated to design and develop their games. Screenshots of each team's games are shown in Figure 2. Students expressed pride in their work, whose distinctiveness was partly due to the open-ended nature of the project. Apart from the instruction to build a single-player game, they had complete creative freedom.

We learned many lessons around remote education, as this was our first time intentionally designing material for remote instruction. First, we learned that one benefit of remote instruction for autistic students included the ability to mute their video and microphone. This allowed them to adjust their engagement with the camp so that they felt as comfortable as possible. Two students did not turn on their video cameras for the majority of the camp. However, Martha turned on her camera for the first time in a breakout because she felt comfortable in a smaller group. Bob stated that he ". . . honestly prefer[red] . . . online courses. . . . I can . . . relax more. . . . You're . . . not staying . . . at a desk watching the instructor. Here, you could just watch from . . . anywhere."

Second, we learned that it is important to send an instructor into every breakout room because students do not always follow the instruction or stay fully focused. The presence of the instructor helped the students stay on task, and allowed them to ask questions when they needed help quickly. The students reported that they did enjoy the camp's remote operation.

Third, we learned lessons that changed our pedagogy and our assumptions about how the camp would operate. First, we appreciated the importance of piloting our material before the camp started to receive feedback and make adjustments. Second, we originally scaffolded 10-minute cycles for the "I do it, you do it, let's talk about it" methodology to keep students engaged. We found they completed the activities while we taught instead of waiting for us.

6 CONCLUSION

We have presented our design for a video game coding camp for autistic first-year undergraduate students. We illustrated the camp's learning goals and offered narrative and quantitative evidence of its impact on its students' communication self-efficacy. Our experiences have taught us lessons about the importance of focusing heavily on communication and collaboration skills to ensure that our autistic students learn skills that will serve them well in university and beyond. We hope others can learn from us and bring valuable educational experiences to their own autistic students.

7 ACKNOWLEDGEMENTS

We thank our students, the Microsoft MakeCode Team for their assistance with technical issues, and Peli de Halleux and Richard Knoll for giving guest talks to the campers.

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] 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*. HICSS '17, Hawaii, HI, USA, 5501–5510.

[3] American Psychiatric Association. 2013. Diagnostic and Statistical Manual of Mental Disorders (5th ed.).

[4] J. Baio, L. Wiggins, DL Christensen, and et al. 2014. Prevalence of Autism Spectrum Disorder Among Children Aged 8 Years — Autism and Developmental Disabilities Monitoring Network, 11 Sites, United States, 2014. *MMWR Surveill Summ 2018* 67, SS-6 (2014), 1–23. https://doi.org/10.15585/mmwr.ss6706a1

[5] 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

[6] Scott Bellini. 2006. The development of social anxiety in adolescents with autism spectrum disorders. *Focus on autism and other developmental disabilities* 21, 3 (2006), 138–145.

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

[8] Benoît Bossavit and Sarah Parsons. 2017. From start to finish: teenagers on the autism spectrum developing their own collaborative game. *Journal of Enabling Technologies* 11 (2017), 31–42.

[9] Bernardo José da Silva Estácio and Rafael Prikladnicki. 2015. Distributed Pair Programming: A Systematic Literature Review. *Information and Software Technology* 63 (2015), 1 – 10. https://doi.org/10.1016/j.infsof.2015.02.011

[10] Zoe Dayan. 2017. 3 Reasons Autistic Children Excel at Computer Coding. https://www.codemonkey.com/blog/3-reasons-autistic-children-excel-at-computer-programming.

[11] Mary Anne L. Egan. 2005. Students with Asperger's Syndrome in the CS Classroom. In *Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education* (St. Louis, Missouri, USA) *(SIGCSE '05)*. Association for Computing Machinery, New York, NY, USA, 27–30. https://doi.org/10.1145/1047344.1047369

[12] K. Eiselt and P. Carter. 2018. Integrating Social Skills Practice with Computer Programming for Students on the Autism Spectrum. In *2018 IEEE Frontiers in Education Conference (FIE)*. IEEE, San Jose, CA, 1–5.

[13] Thomas Erickson and Wendy A. Kellogg. 2000. Social Translucence: An Approach to Designing Systems That Support Social Processes. *ACM Trans. Comput.-Hum. Interact.* 7, 1 (March 2000), 59–83. https://doi.org/10.1145/344949.345004

[14] Kristen Felicetti. 2016. These major tech companies are making autism hiring a priority. https://www.monster.com/career-advice/article/autism-hiring-initiatives-tech. Accessed: 2020-08-27.

[15] Sarah Goff-Dupont. 2019. Can the future of tech include Autism Spectrum Disorder? https://www.atlassian.com/blog/teamwork/can-future-tech-include-autism-spectrum-disorder

[16] Lorcan Kenny, Caroline Hattersley, Bonnie Molins, Carole Buckley, Carol Povey, and Elizabeth Pellicano. 2015. Which terms should be used to describe autism? Perspectives from the UK autism community. *Autism* 20, 4 (2015), 442–462.

[17] Sanna Kuusikko, Rachel Pollock-Wurman, Katja Jussila, Alice S Carter, Marja-Leena Mattila, Hanna Ebeling, David L Pauls, and Irma Moilanen. 2008. Social anxiety in high-functioning children and adolescents with autism and Asperger syndrome. *Journal of autism and developmental disorders* 38, 9 (2008), 1697–1709.

[18] Microsoft. 2020. MakeCode Arcade. https://arcade.makecode.com/. Accessed: 2020-08-27.

[19] Roberto Munoz, Rodolfo Villarroel, Thiago S. Barcelos, Fabián Riquelme, Ángeles Quezada, and Patricia Bustos-Valenzuela. 2018. Developing Computational Thinking Skills in Adolescents With Autism Spectrum Disorder Through Digital Game Programming. *IEEE Access* 6 (2018), 63880–63889.

[20] Vivek Paramasivam, Justin Huang, Sarah Elliott, and Maya Cakmak. 2017. Computer Science Outreach with End-User Robot-Programming Tools. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (Seattle, Washington, USA) *(SIGCSE '17)*. Association for Computing Machinery, New York, NY, USA, 447–452. https://doi.org/10.1145/3017680.3017796

[21] Kirsten Ribu. 2010. Teaching computer science to students with asperger's syndrome. In *Proceedings from NIK-2010: The Norwegian Informatics Conference*. NIKT, Bergen, Norway, 99–111.

[22] Ian Schreiber. 2011. Game Design Concepts. https://gamedesignconcepts.wordpress.com/. Accessed: 2020-08-27.

[23] Teddy Seyed, Peli de Halleux, Michal Moskal, James Devine, Joe Finney, Steve Hodges, and Thomas Ball. 2019. MakerArcade: Using Gaming and Physical Computing for Playful Making, Learning, and Creativity. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI EA '19)*. Association for Computing Machinery, New York, NY, USA, 1–6. https://doi.org/10.1145/3290607.3312809

[24] Debbie Spain, Jacqueline Sin, Kai B Linder, Johanna McMahon, and Francesca Happé. 2018. Social anxiety in autism spectrum disorder: A systematic review. *Research in Autism Spectrum Disorders* 52 (2018), 51–68.

[25] Stephanie Teasley, Lisa Covi, Mayuram S Krishnan, and Judith S Olson. 2000. How does radical collocation help a team succeed?. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*. ACM, Philadelphia, PA, 339–346.

[26] Stavros Tsikinas and Stelios Xinogalos. 2019. Studying the effects of computer serious games on people with intellectual disabilities or autism spectrum disorder: A systematic literature review. *Journal of Computer Assisted Learning* 35, 1 (2019), 61–73.

[27] Bruce W. Tuckman and Mary Ann C. Jensen. 1977. Stages of Small-Group Development Revisited. *Group & Organization Studies* 2, 4 (1977), 419–427. https://doi.org/10.1177/105960117700200404

[28] 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

[29] Gulnoza Yakubova and Teresa Taber-Doughty. 2017. Improving problem-solving performance of students with autism spectrum disorders. *Focus on Autism and Other Developmental Disabilities* 32, 1 (2017), 3–17.

[30] Annuska Zolyomi, Andrew Begel, Jennifer Frances Waldern, John Tang, Michael Barnett, Edward Cutrell, Daniel McDuff, Sean Andrist, and Meredith Ringel Morris. 2019. Managing Stress: The Needs of Autistic Adults in Video Calling. *Proc. ACM Hum.-Comput. Interact.* 3, CSCW, Article 134 (Nov. 2019), 29 pages. https://doi.org/10.1145/3359236

[31] Annuska Zolyomi, Anne Spencer Ross, Arpita Bhattacharya, Lauren Milne, and Sean A. Munson. 2018. Values, Identity, and Social Translucence: Neurodiverse Student Teams in Higher Education. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3173574.3174073

[32] Misbahu S. Zubair, David J. Brown, Matthew Ian Bates, and Thomas Hughes-Roberts. 2019. Designing Accessible Visual Programming Tools for Children with Autism Spectrum Condition. *CoRR* abs/1911.07624 (2019). arXiv:1911.07624 http://arxiv.org/abs/1911.07624