TwoTorials: A Remote Cooperative Tutorial System for 3D Design Software

Anonymous Author(s)

Abstract

Step-by-step tutorials have emerged as a key means for learning complex software, but they are typically designed for individuals learning independently. In contrast, cooperative learning, where learners can help each other as they work, is a fundamental pedagogical technique with many established benefits. To extend these benefits to learning 3D-design software, this work investigates the design of remote cooperative software tutorial systems. We first conduct an observational study of dyads of participants working on 3D-design tutorials, which reveals a range of potential benefits, challenges, and strategies for cooperation. Our findings inform the design of *TwoTorials*, a cooperative step-by-step tutorial system that helps pairs of remote users establish shared 3D context, maintain awareness of each other's activities, and coordinate their efforts. A user study reveals several benefits to this approach, including enhanced cooperation between learners, reduced effort and mental demand, increased awareness of peer activities, and higher subjective engagement with the tutorial.

Keywords: Software learning, 3D modeling, remote learning

1 Introduction

Users starting out in 3D design software face a range of learnability challenges [36], which have motivated the development of a variety of innovative software learning systems (e.g., [13, 36, 51, 59, 78]). In particular, step-by-step tutorials have emerged as a key means for learning complex software, and tutorials of this type exist for nearly all popular applications. In some ways, these tutorials replicate the experience of working on non-trivial projects using the software, with the tutorial providing a clear goal and scaffolding the user's skills and abilities [56]. However, this format of tutorials is primarily designed for individuals learning independently, so users cannot benefit from over-the-shoulder learning [88] and other advantages that come from learning alongside other people, such as occurs in workplace settings. This is unfortunate, because education research has established cooperative learning as a fundamental pedagogical technique [22] with benefits in terms of learner motivation [18, 75], retention [5, 80], and effective knowledge gain and transfer [50, 64].

In this paper, we are interested in how the benefits of cooperative learning can be made available on-demand to remote learners of 3D-design software, both to address some of the above challenges with tutorials, and to extend the benefits of over-the-shoulder learning to users who would not be able to benefit from it otherwise (e.g., users who are learning at home, either informally or through online courses). To this end, we developed *TwoTorials* (Figure 1) a tutorial system that allows pairs of remote users to complete a tutorial in parallel, with mechanisms to facilitate beneficial learning interactions. Through the design and development of TwoTorials, and two studies, our work addresses the following research questions: (1) *What are the salient components of cooperative learning for step-by-step 3D-design tutorials, including potential benefits, challenges, and common strategies*? and (2) *What are the appropriate design principles and interface features to support these components*?

To address these questions, we first conducted an observational study of four dyads completing step-by-step tutorials in Tinkercad, a popular 3D solid modeling application. The results of this study revealed potential benefits, challenges, and coordination strategies between users cooperatively completing step-by-step tutorials, such as the need to rapidly establish shared context to support their communication, and a hesitance to help one another if not explicitly asked.

Based on this initial study, we derived a set of five design principles for cooperative software tutorial systems and instantiated these in our TwoTorials prototype. The system provides mechanisms to help establish shared context, synchronize user progress, and facilitate non-disruptive communication between the peers.

To evaluate TwoTorials, we ran a second user study with six dyads of participants, comparing a baseline system with minimal coordination features to the TwoTorials system. Our findings, based on a within-subjects mixed-methods user study, indicate that TwoTorials helped participants to complete tutorials faster, significantly reduced their effort and mental demand, and helped them to maintain a higher level of awareness of each other's progress.

Building on previous work in software learnability and cooperative learning, and our interest in fostering peer help for 3D design software tutorials, this work makes three main contributions. First, we contribute a deeper understanding of the potential benefits, challenges, and common behaviors surrounding cooperative learning of 3D design software. Second, based on these findings, we present a set of design principles for remote cooperative tutorial systems, and instantiate these principles in what we believe to be the first cooperative software tutorial system. Finally, a user study contributes an understanding of the benefits of such system for learning feature-rich software, and points to directions for further work, including generalizations to larger peer groups and other software domains.

2 Related Work

This work is related to prior research on software learning and tutorial systems, cooperative learning and distributed teamwork, and cooperative interfaces for multiplayer games. We review each of these areas below.

2.1 Software Learning and Tutorial Systems

Early research on software learning established a tendency for learners to abandon printed manuals and other learning materials that take time away from their primary task [8, 23, 36]. This has led to a rich body of research on systems and tools to support learning of software applications [13, 25, 78]. In particular, prior work has demonstrated the benefits of step-by-step tutorials and

gamified tutorial systems [23, 59], as well as systems that allow users to learn in the context of realistic tasks [28, 35, 78].

A number of research projects have explored methods for harnessing community-created content, or improvements to learning content contributed by other learners, such as improved workflows [11, 37, 58], multimedia demonstrations of tutorial steps [13], or comments on tutorial content [7]. Recent research has also proposed new approaches for how groups learn 3D design [25, 48]. For example, Maestro [25] enables facilitators of 3D modelling workshops to track the progress of their classrooms in real-time, and provides simple mechanisms for provide help to students when needed.

While the above approaches appear to be valuable, research in the education community has demonstrated a range of benefits to active cooperative learning approaches, in which learners are able to directly interact with one another [14, 15, 52] (discussed in detail in the next section). To provide such an active learning experience, some work on software tutorial systems has integrated elements from games [21, 57, 59]. For example, CADament [59] enables users to learn 3D design software skills by observing the workflows of opponents in a competitive multiplayer learning game. The system enables competitors to engage in "over the shoulder" learning, but it is not focused on creating an environment where learners can work on tasks together to help and benefit from each other. Currently, there exists no step-by-step tutorials that explicitly support remote cooperation. To fill this gap, we build on this body of prior work but focus on supporting active cooperative approaches for learning 3D design software with other users, and propose the first known step-by-step tutorial system specifically designed to support remote peer learning.

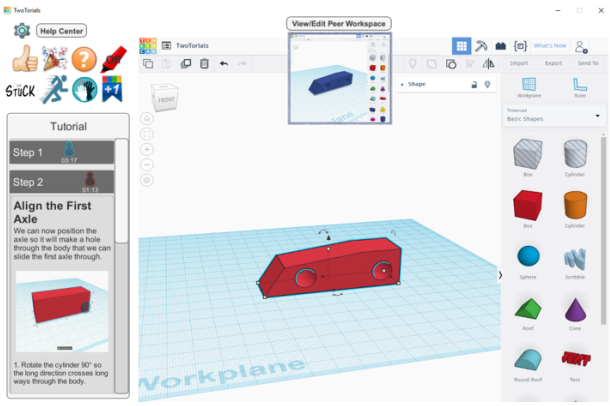

Figure 1: The *TwoTorials* system.

2.2 Remote and Cooperative Learning

Remote learning is becoming increasingly prevalent in our world. The global coronavirus (COVID-19)[1] pandemic has demonstrated the need to establish effective remote learning environments [3, 10, 70], and is forcing educators and learners to rethink pedagogical methods and approaches [6, 20, 81]. A key question raised in this context is how to enable remote learning that preserves the social aspects of in-person learning, giving learners opportunities to engage and interact with each other, which has been shown to aid in motivation and creating positive learning experiences [16, 31, 47, 76, 91]. In the present work, we aim to support remote synchronous coordinated learning for the specific learning resource of step-by-step tutorials.

[1] The 2019/2020 global novel coronavirus (COVID-19) pandemic [10]

Collaborative learning is an educational approach in which learners work together to solve a problem or complete a task, recognizing that learning is a naturally social activity [22, 65]. There are many approaches to foster collaborative learning, and significant literature showing its effectiveness both in co-located and remote learning environments [22, 44, 72]. Cooperative learning is a particular type of collaborative learning in which a set of processes help people interact together to accomplish specific goals, helping themselves and others to learn [64, 65, 85]. In terms of specific benefits, social activities have been shown to increase the motivation to learn from others, and to result in effective knowledge gain and transfer [16, 18, 75]. Critically, this kind of social learning does not need to consist of continuous interaction between the learners [64, 73] – simply being able to work together in a social environment provides the opportunity for both passive and active learning. For example, "over-the-shoulder learning" can occur from observing another learner while they are completing a task, or by actively engaging in completing the task together [18, 88]. When compared to individual and competitive learning, cooperative learning has been demonstrated to be particularly effective for sustaining learner motivation [63, 71].

In terms of particular mechanisms for enabling remote cooperative learning, prior work has shown that cooperative learning is most effective when learners organize their activities, synchronize their effort, and maintain shared situational awareness [29, 50, 82]. Research on distributed teams has also shown the importance of awareness mechanisms, to enable team members to inform one another of their status [40, 92]. Although team or group awareness can be easily maintained in co-located collaborative environments, it can be difficult in remote collaboration [40]. Thus, groupware research has focused on interfaces and techniques that facilitate communication, increase group awareness, and enable cooperation, such as capturing eye gaze [18, 19] and other awareness cues [12, 92]. A full literature review of groupware research is beyond the scope of this paper but we point readers to existing surveys on the topic [26, 40, 79].

2.3 Cooperative Gaming

Games are a prominent example of systems that cultivate cooperative behavior [68, 84, 87]. Cooperative games provide different interfaces and mechanics to facilitate multiplayer interaction [1, 45, 55]. In cooperative games, mutual understanding of the objectives between the players are essential to their success. Players must maintain awareness of each other and establish a common ground for communication [9]. Cooperative games also provide players with a verity of explicit and implicit communication mechanics, including awareness cues and cooperative communication mechanics [2, 12, 67, 87, 92]. These mechanics help teams communicate with each other and maintain high level of awareness. In this work, we draw on the body of past work on multiplayer games and gamification to develop interfaces to foster effective cooperative learning for a qualitatively different domain – step-by-step tutorials for 3D design software.

In summary, our current work contributes to the understanding of cooperative learning for the domain of step-by-step tutorials for 3D design software, and the TwoTorials prototype system provides specific mechanisms to enable shared awareness and support cooperative learning in this domain. To the best of our knowledge this work represents the first application of a cooperative active learning approach to step-by-step software tutorial systems.

3 Observational Lab Study

To inform the design of our cooperative step-by-step tutorial system, we conducted an observational study with pairs of participants. Our main goal was to understand how peers cooperate with each other to complete this type of tutorial, the challenges they face, how they synchronize their work, and how they encourage and support each other.

3.1 Study Procedure

Each pair of participants completed two step-by-step tutorials for Tinkercad drawn from those provided in-product in the software (Balloon Powered Car and Roman Dome), each lasting ~30 minutes, followed by an individual survey and a short semi-structured interview.

We intentionally tested a range of different cooperative tutorial setups (Figure 2), to gain a broad set of insights into the benefits and challenges that arise in different forms of collaboration. These included co-located vs. distributed setups (simulated by a partition between the participants, which permitted them to talk to each other, but still required the use of screen sharing to view each other's workspaces), and separate vs. shared workspace setups (i.e., whether both participants were working on one project together, or working on the same project in parallel). Our decision to use a partition to simulate a distributed condition was designed to reduce the complexity of the study setup, and is an approach that has been used in prior work [18]. In this study, each dyad of participants completed two tutorials across one of the axes of the four setups shown in Figure 2, enabling them to comment in greater detail on the effect of that axis.

The experimenter took observations and provided participants with assistance with technical difficulties but did not help the participants with completing the tutorial instructions. Video and audio recording of the study and post-study interviews were transcribed and analyzed for common themes. Each study session lasted ~60 minutes total.

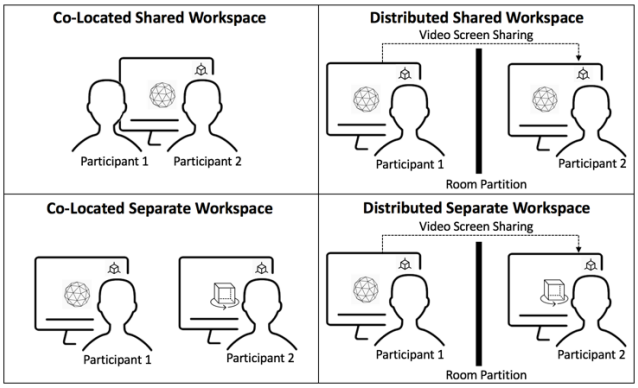

Figure 2: The four cooperative tutorial setups that were tested.

3.2 Analysis

Interview transcripts and observations were analyzed using methods drawn from grounded theory [33]. Specifically, open coding was used to label transcript data, and emerging themes and patterns were identified by the first author and then shared and discussed with the broader research team. The themes that emerged relate to the potential benefits to users from cooperative leaning of 3D design software, challenges experienced by peers when learning 3D design software together, and common strategies used to cooperatively learn.

3.3 Participants

Four dyads (8 participants total (6 male, 2 female), mean age 38.4 years (SD 10.3)) were recruited via an email to employees of a large software company. As dyads volunteered together, they are best considered as coworkers or friends. Two dyads were all male, and two were mixed (one male, one female). All participants reported having completed a bachelor's degree. All participants were screened for prior experience with 3D modeling software; 1/8 participants had no experience, 3/8 participants had minimal experience, 3/8 participants had some experience, and 1/8 participants had extensive experience. The most common 3D modeling applications used previously by participants were Maya, Blender, and Alias. Only one participant had prior experience with TinkerCAD. Each participant received a $25 gift card as compensation for their participation.

3.4 Results

We begin by discussing our observations of the tradeoffs of separate vs. shared workspaces and co-located vs. distributed workspaces, and then discuss our findings on the benefits, challenges, and strategies used by participants to cooperatively complete step-by-step tutorials.

3.4.1 Separate vs. Shared Workspace

Neither the separate nor shared workspace setups were revealed to be clearly superior for enabling cooperative learning, with both showing advantages and disadvantages. Having a shared workspace forced the peers to collaborate, which was beneficial, but created a situation where the participant that was not 'driving' the system could become frustrated, and feel like they were missing out on learning:

> When I was just watching it was frustrating to not be able to take actions myself. We were trying to figure out how the interface works, and I want to be able to create my own objects to explore the manipulators and what is possible. (P5)

Conversely, participants reported that working on separate workspaces created a feeling of working in parallel on separate tasks:

> We had the video sharing, but we were both doing our own thing, so we only looked at each other's views to make sure our work looked somewhat similar. It didn't really seem like a cooperative effort [in the distributed separate workspace condition], more like we were just doing the same thing at the same time. (P3)

This observation is consistent with prior work on personalizable groupware that can support individual and group activities [34, 55]. A design approach that emerged from this observation was the idea of a hybrid system, which would allow each of the peers to benefit from actively working on the tutorial individually, while encouraging cooperation and peer help.

3.4.2 Co-located vs. Distributed

Contrasting the co-located and distributed setups revealed a range of challenges to coordinating effort when participants were distributed. When co-located, it was much easier for participants to make spatial references to parts of the 3D environment and to assist one another by looking at each other's screens, pointing at parts of their peer's screen, or even taking over the mouse of their peer to rotate the camera or make simple changes to 3D objects.

Consistent with prior work on collaborative remote physical tasks [29, 54], cooperative help-giving and receiving in a 3D design tutorial was much more difficult when participants were distributed. Participants were not always able to clearly understand each other due to a mismatch in their respective views of the 3D environment, and providing verbal instructions became complex without the ability to ground the instructions in spatial references, or to make direct changes:

> Explaining how I want my partner to try using the manipulator with words is much slower than just being able to do it myself. (P4)

These observations suggest that additional coordination mechanisms are needed to enable productive cooperative learning in distributed setups.

3.4.3 Benefits of Cooperative Learning of 3D Design Software

In terms of the benefits to cooperative learning of 3D design software, participants reported having an overall positive experience, and suggested that this approach allowed them to gain additional insights beyond the tutorial content:

> When we both were doing the tutorial, it just felt that, wow, that moved on very quickly, and [I] actually still learned something, and it might not [have] been what was intended to be learned through the steps, but, like, the other person's insights. (P2)

Participants also pointed out the benefit of being able to quickly detect errors and identify if they were misunderstanding the tutorial instructions:

> I think working together has a lot of advantages. You can detect errors very quickly and keep on making progress. (P5)

Participants also indicated that cooperatively working on a tutorial helped them to accelerate their learning and sustain motivation to learn the tutorial content:

> It accelerated the learning since it was a shared experience and we could communicate what our successes and failures were to each other. (P8)

Overall, these observations point to several potential benefits of cooperative learning of 3D design software, which are worthy of further investigation.

3.4.4 Strategies for Cooperation

Our observations and interviews indicated several common strategies that peers used to cooperate with one another. During help-seeking instances, we observed that participants in distributed setups started by establishing common ground and shared 3D context with their peer as a first step when providing assistance. For example, they would ask questions such as, "which view are you on – top, side, bottom?" And then they would change to that view and proceed to make recommendations and provide help:

> Got to first understand the language and perspective and then give feedback after. (P4)

This strategy was more common when participants were distributed. Related to this theme, we observed that peers would frequently communicate which step in the tutorial they were on, or signal to their peer that they are moving on to the next step, as suggested by the following quote in response to a question on what techniques or practices P4 was using to work together and synchronize their efforts with their peer:

> Make sure to communicate that we were on the same step and sub-step. (P4)

This strategy suggests that mechanisms for establishing shared context between learners could be beneficial, particularly if they can support the sharing of 3D viewpoints and the step of the tutorial a learner is currently on.

A final beneficial strategy we observed was that looking at their peer's workspace provided learners with insights into their own work and how it could be improved. This over-the-shoulder learning [88] was observed in multiple instances where peers would spend time observing each other completing a step of the tutorial and then attempt it themselves.

> I can look at the other person's work and say, mine doesn't look anything like that, and then you know there is a problem. (P1)

In terms of design guidance, providing explicit support for this learning strategy could be valuable.

3.4.5 Challenges in Cooperative Learning of 3D Design Software

Finally, we observed several challenges that can come from working cooperatively on step-by-step tutorials for 3D design software. For example, participants struggled to maintain awareness of each other's activities, due to a lack of shared context:

> Sometimes it is difficult when you can not see what I'm seeing, you would be like, oh it is like this, and I'm like no it is not, we both are seeing different things, and we are arguing about nothing, that was frustrating. (P5)

While maintaining awareness is a known issue for groupware applications [27, 40–42, 92], these challenges appeared to be exacerbated from working in a 3D environment, where each peer had a different camera orientation on the scene, making it difficult to establish shared context when help is needed:

> At some times, we both were oriented in different ways, and I'm like something is wrong here, and she is like, oh no we just need to adjust the orientations to match. (P1)

Another challenge reported by participants was that it was difficult to determine when feedback or help was needed or would be welcomed by their peer. This was especially prominent in the distributed setups, where awareness of activities between peers was less strong. We believe that the domain of 3D design software exacerbates this problem, because the editing history of a model, or any mistakes made on previous steps, are not obvious to a peer observing the model "over the shoulder" as the user works on it.

Finally, while not necessarily a challenge specific to 3D design software, peers found it difficult to synchronize their progress in the step-by-step tutorials. In the distributed setups, participants would verbally share which step they were on in the tutorial, to help each other maintain constant awareness of each other's progress, and to signal if one of them was falling behind and might need help. This was less of a problem in the co-located

setups, where participants could glance at their peer's screen to get the same information:

> I found that my partner jumped to the next step. I needed to confirm what step he was on so that I could help with the next step (P4)

Past work has suggested that this kind of communication overhead can be distracting [62], which can detract from the learning process. Thus, it may be beneficial to design features to reduce this "orienting communication" overhead.

4 Design Principles

The results of our observational study complement prior literature and provide an understanding of the main challenges and breakdowns faced by peers learning 3D design software through step-by-step tutorials. In particular, our findings are consistent with known issues surrounding control ownership, and collocated use of groupware systems, but also reveal important and unique insights specifically related to both cooperative use of step-by-step tutorials, and learning challenges for 3D software. Pulling together the observations and insights from the study, we suggest a set of five design principles for cooperative step-by-step tutorial systems for 3D design software:

4.1 Help Establish Shared 3D Context (D1)

The system should assist with establishing and maintaining shared 3D context between peers, to make giving and receiving help easier. The need to establish and maintain shared 3D context has been identified in prior work [24, 30, 90], but it presents a particular challenge for step-by-step tutorials, where each user may have a different camera position and orientation on a separate 3D workspace, whose 3D content may be at a different stage of the tutorial than their peer. This makes it difficult to establish the context necessary to reference 3D objects or meaningfully discuss their orientation.

4.2 Balance Independent Action with Encouraging Collaboration (D2)

Two competing challenges we observed were peers becoming frustrated with not being able to 'drive' in the shared workspace condition, and peers not engaging with each other in the separate workspace condition. Prior work has shown that giving users the power over navigation, manipulation, and representation within shared workspaces supports collaboration, but has its tradeoffs [39]. The system should balance the need for independence, while also encouraging collaboration between the peers, to create a beneficial cooperative learning experience where both users are engaged with the tutorial task.

4.3 High-Level Awareness of Progress (D3)

The system should provide learners with high-level awareness of where their peer is in the tutorial steps and in the 3D workspace, and help make learners aware of any challenges or setbacks faced by their peer. This is particularly relevant for 3D environments, where it is more difficult to maintain awareness and establish mutual orientation and view between peers [24, 30].

4.4 Non-disruptive Communication Mechanisms (D4)

The system should provide non-disruptive communication modalities that simplify and complement the beneficial cooperative learning practices that we observed. Prior work suggests that communication can be less disruptive when timing and communication method are selected appropriately [17].

4.5 Synchronize Progress (D5)

To increase the likelihood of beneficial cooperation, and to try and avoid the situation where one learner quickly finishes the tutorial and becomes bored, the system should encourage peers to work together and synchronize their progress through the tutorial steps.

Guided by the principles above and previous research in this area, we developed TwoTorials, a cooperative step-by-step tutorial system designed for pairs of learners, which we present next.

5 The TwoTorials System

TwoTorials offers a cooperative learning environment for two distributed users. The pair of users work cooperatively to learn 3D design software by each completing the same step-by-step tutorial in parallel (Figure 1). The system includes a set of features to support coordination and establish shared context within the tutorial. In the current work, we designed the system to work with Tinkercad, a popular web-based 3D solid modeling tool [4]. In this section, we start with a high-level overview description of TwoTorials, then we highlight the main features, noting in parentheses the relevant design principles each of these features is intended to address, and citing any prior work that influenced the designed features.

5.1 System Overview

Each remote peer gets an individual workspace, as well as access to a constantly-updating and editable view of their peer's workspace, enabling users to observe their peers and actively assist them if needed. Peers can communicate, help, and encourage each other using both verbal and non-verbal communication modalities. The system also provides implicit awareness cues to helps learners maintain awareness of their own progress in the tutorial and whether their peers are falling behind and may need help; and can enforce a level of interdependence between the learners as a means to encourage them to work together and help each other out. Finally, as a user completes each step, their peer is provided with a screen recording of their efforts on that step, providing further material for the peers to reference when helping one another. The sections that follow describe the above features in detail.

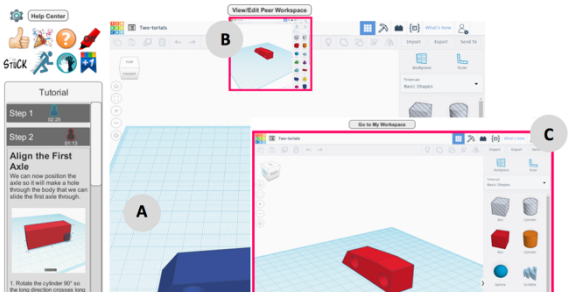

Figure 3: User and peer workspaces. (A) the user's workspace; (B) a constantly-updating view of the peer's screen; (C) an expanded view of the peer's screen (accessible by clicking the small view).

5.2 Seamless View, Transition, and Editing between Workspaces

In TwoTorials, each user gets a small, constantly-updating view of their peer's screen (Figure 3B) displayed above their own workspace (Figure 3A). Clicking the small view of their peer's screen expands it to full screen (Figure 3C) and allows the user to

directly edit their peer's workspace. Through these mechanisms, learners are able to constantly monitor their peer's progress, enabling over-the-shoulder learning [88], and helping to establish shared context (D1). The ability to make changes to their peer's workspace in the expanded view enables a user to directly provide assistance or demonstrate editing operations on the peer's 3D model (D2). Prior work has shown these kinds of seamless transition mechanisms from individual to shared spaces to be important for facilitating collaboration [29, 32, 53, 83].

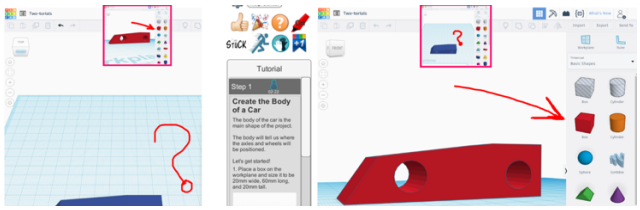

Figure 4: Drawing Annotations enable the user to draw over the 3D workspace, enabling cooperation and conversational grounding between peers [2, 44].

5.3 Verbal and Non-Verbal Communication Features

The system provides in-tutorial voice and text chat, allowing peers to verbally communicate or send text messages to each other (D4). To complement these communication methods, the user can also create free-hand drawing annotations on top of both workspaces in the form of free-hand lines and simple shapes (Figure 4) [2, 74]. These non-verbal communication help users to ground their conversations or direct their peer's attention to particular parts of the UI or 3D workspace (D1). Prior work has shown such mechanisms to be effective for keeping users engaged in collaborative activities in games [2, 89] and online courses [44].

Users can also send peer pings, a set of predefined visual messages that provide simple, non-disruptive communication between users. Clicking on one of these pings (Figure 5), sends a visual message to their peers that lasts for a couple of seconds, displayed on top of their peer's workspace. Each of these pings is designed to indicate a situation where cooperation is needed, such as having a question, being stuck, or expressing the need to move faster (D5). Peer pings are also supported to celebrate success or provide encouragement, such as sending fireworks, high-fives, and thumbs ups (D2). This type of lightweight communication mechanism has been shown to be effective in encouraging participation in gaming and live streaming contexts [43, 44, 69].

Figure 5: *Peer Pings* are predefined visual messages that provide lightweight communication between peers.

5.4 Implicit Awareness Cues

To enable users to maintain awareness of their peer's progress (D3), each user has a simple avatar that moves through the list of tutorial steps as they proceed though the tutorial content (Figure 6). A timer indicates the amount of time the user has spent at the current step, further fostering peer awareness. Finally, to provide spatial awareness [41, 42] of a peer's activities within a step, the system displays the peer's mouse cursor in the user's workspace.

The non-verbal awareness cues described above allow users to maintain awareness of each other's activities in a lightweight manner, without the need to constantly communicate their status explicitly (D5). Similar mechanisms have been shown to be important for enabling users to maintain shared awareness [40, 92], especially for distributed environments, which lack the sensory cues that ease collaboration in co-located settings [12].

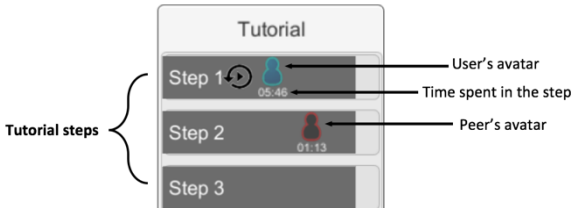

Figure 6: Tutorial-progress awareness cues, including the user avatar, and indicator of time spent on the current step.

5.5 Progress Control Mechanism

Before starting a tutorial, users can select one of three levels of step-synchronization, which affect how much the system enforces synchronization of activities between the peers (Figure 7). At the most extreme, the Strict setting prevents each user from moving on to the next step until they are both finished the current step (indicated by clicking a button). The Moderate setting enables a user to move one step ahead of their peer, and if they try to move any further they are prompted to wait. Finally, the Free setting puts no restrictions on movement through the steps. These mechanisms provide system-imposed synchronization of progress (D5), primarily motivated by our observational study results.

Figure 7: The progress control mechanism enables users to control the progress of each peer in the tutorial.

5.6 Workflow Replay

The system records a video of each user's screen as they work on a step and makes this recording available to their peer upon proceeding to the next step. By clicking a replay icon, the peer can view a video showing the exact steps the user took to complete that step (Figure 7). Past work has shown that this kind of short demonstration video can be particularly valuable for learning design software [37, 59], and this feature also frees a user from having to explain the exact process they followed – they can simply prompt their peer to check the recording video (D2).

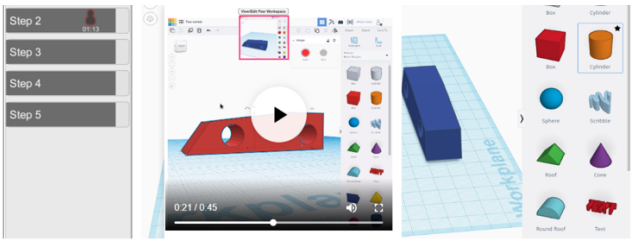

Figure 8: Video replay window.

5.7 Access to Online Help Resources

The system provides quick in-application access to online and community-based help resources (e.g., the Tinkercad help center). This enables users to access help without disengaging from the tutorial experience (D2).

5.8 System Implementation

TwoTorials was implemented in two parts. First, the step-by-step tutorial system was built as a Unity application. This enabled us to quickly build a multi-user system by taking advantage of Unity's networking capabilities to provide a reliable, low-latency connection between the peers for sending media streams, including voice and text chat, user progress data, shared annotations, and peer pings. Screen recording and playback was implemented using a Unity plugin that enables real-time video and audio capture and streaming. The Tinkercad application was embedded using a Unity web-browser component, which mirrored a locally-running version of Tinkercad. The second part of the system consisted of a modified version of the Tinkercad application to add required concurrent editing features to the application.

5.9 Tutorial Format, Authoring, and Progress Tracking

Each tutorial step consists of text and images (Figure 1, left). We adopted this format to match as close as possible the in-product tutorials available in Tinkercad, which we used for the baseline condition in our evaluation study, explained in the next section. In terms of tutorial authoring, text was manually entered, and figures were added to a folder that was read by the system. TwoTorials tracks progress solely based on navigation through the tutorial steps (users explicitly clicking "next step"). More sophisticated tracking of Tinkercad tool usage or the 3D content being created is an interesting avenue for future work.

6 Evaluation

We conducted a user study to understand users' reactions to the TwoTorials system and its cooperative features, and to gain further insights into the cooperative experience of step-by-step tutorials.

6.1 Study Procedure and Design

The study followed a within-subjects mixed-methods design, with each dyad of participants completing two step-by-step tutorials, one using TwoTorials, and the other using Tinkercad's built-in tutorial interface. These tutorials were the same as those used in the previous observational study, which had revealed them to be about the same level of difficulty. For the TwoTorials condition, the progress control setting was set to 'Free'. Although in-application voice chat was implemented in the system, the setup of the study resulted in us not needing to use it – participants were simply instructed to talk with each other over the divider (similar to our observational study and methods used in prior work [18]).

In the baseline condition, participants used the Tinkercad tutorial along with a live screencast of their workspace, shared with their peer through Google Hangouts. We provided this capability in the baseline condition because it seemed unrealistic for users to collaborate with no view of their peer's workspace whatsoever. Participants in this condition were also able to talk with each other over the divider.

To rule out ordering and learning effects, condition order and mapping of tutorials to conditions was fully counterbalanced.

At the start of the study, participants were provided informed consent, and asked to complete a questionnaire on demographics and prior 3D design software experience. Next, the experimenter introduced the study system, and the available cooperative features, before allowing the participants to work on the tutorial. The experimenter did not help participants with working through the tutorial instructions but did provide limited assistance in response to technical difficulties with the study system. After completing each condition, a set of Likert-style questions were administered on the overall experience, ease of following the tutorial, learning, and usefulness of the cooperative features. The NASA-TLX questionnaire was also administered, to assess workload [45, 46]. At the end of the study session, a post-study open-ended questionnaire was administered. The study took ~60 minutes total to complete.

6.2 Participants

Six dyads (12 participants total (10 male, 2 female), mean age 35.8 years (SD 7.9)) were recruited via an email to employees of a large software company. Each dyad was either friends or co-workers, with 1 dyad all female, and 5 all male. All participants were screened for prior experience with 3D modeling software; 1/12 participants had no experience, 4/12 participants had minimal experience, 5/12 participants had some experience, and 2/12 participants had extensive experience. Most common 3D modeling applications used previously by participants were Fusion 360, Maya, and SolidWorks. Only two participants had prior experience with TinkerCAD. Each participant received a $25 gift card as compensation for their participation.

6.3 Results

We begin by presenting the main quantitative findings, comparing TwoTorials to the baseline. We then present results from the post-condition questionnaire, and the usage and subjective ratings for TwoTorials features. Finally, we discuss our qualitative and semi-structured interview findings.

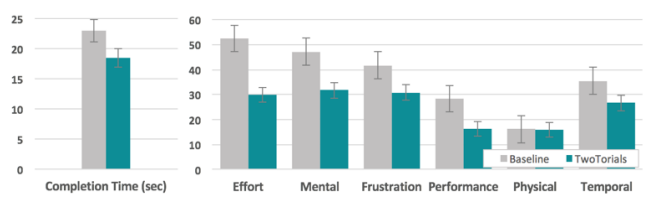

Figure 9: Completion time in seconds and NASA-TLX results (lower is better).

6.3.1 *Performance Results – TwoTorials vs. Baseline*

A Wilcoxon Signed-Rank Test showed that dyads spent significantly less time to complete the tutorial together using TwoTorials (M = 18.5) compared to the Baseline (M = 23) ($z = 2.831$, $p<.05$) (Figure 9). These findings provide evidence that the features of TwoTorials helped participants to complete the tutorial together more quickly.

6.3.2 Cognitive Load Results – TwoTorials vs. Baseline

For the cognitive load results (Figure 9), a Wilcoxon Signed-Rank Test showed significantly lower rating for effort ($z = 2.668$, $p<.01$), mental demand ($z = 2.201$, $p<.05$), and frustration ($z = 2.254$, $p<.05$) for the TwoTorials condition as compared to the baseline condition. These findings provide compelling evidence that the features of TwoTorials helped reduce the cognitive load on participants. For the rest of the TLX subscales, we found no significant difference.

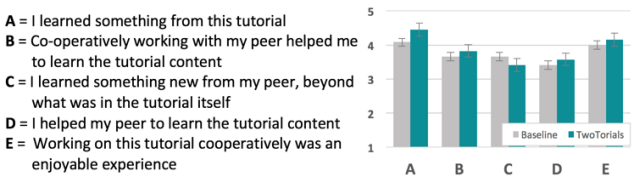

Figure 10: Rating on the learning experience questionnaire (higher is better). Error bars show standard error.

6.3.3 Questionnaire Results – TwoTorials vs. Baseline

When asked which of the two conditions they preferred overall, TwoTorials was rated higher by 5/12 participants compared to the in-application tutorial, with 6/12 participants expressing no preference and 1/12 preferring the baseline condition. While this suggests a preference for the TwoTorials system, a Wilcoxon signed-rank test did not show this difference to be statistically significant.

For each condition, we asked participants a set of questions on what they learned from the tutorial experience (Figure 10). For most of the questions we found no significant difference, but a Wilcoxon Signed-Rank Test showed a significant difference in medians for the statement "I learned something from this tutorial" favoring the TwoTorials condition over the baseline condition ($z = 2.000$, $p < .05$).

We also asked participants a set of questions on the various other aspects of the tutorial-following experience (Figure 11). A Wilcoxon Signed-Rank Test determined that there was a significantly higher median for the TwoTorials system for "maintaining awareness of your peer's activities" as compared to the baseline ($z = 2.197$, $p < .05$). We did not find a significant difference for the other questions in this group.

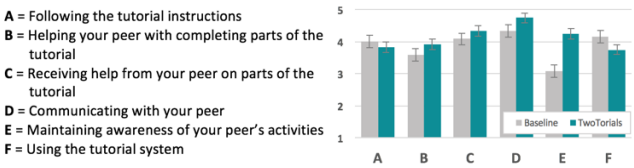

Figure 11: Ratings of the tutorial systems for various statements. Error bars show standard error.

6.3.4 TwoTorials Features

For the TwoTorials condition, we analyzed how many times each feature was used by participants, and asked participants to rate the usefulness of the individual features. In terms of usage, participants switched to their peer's workspace an average of 4.8 times (SD=1.94) and edited their peer's workspace directly 2.2 times (SD=0.75). Participants annotated each other's workspaces 2.3 times (SD=1.03) and sent 4.6 peer pings (SD=2.42). Considering that dyads in the TwoTorials condition took less than 25 minutes to complete the tutorial, these numbers suggest that the features of TwoTorials were used frequently by participants.

The ratings of usefulness for the individual features of TwoTorials are shown in Figure 12. Participants generally reported the features to be useful. There was strong support for the voice chat, the ability to view the peer's workspace, and the ability to directly edit the peer's workspace. The only feature to receive a strong negative rating for usefulness was the text chat, which is likely because the voice chat provided a much richer and more convenient communication medium.

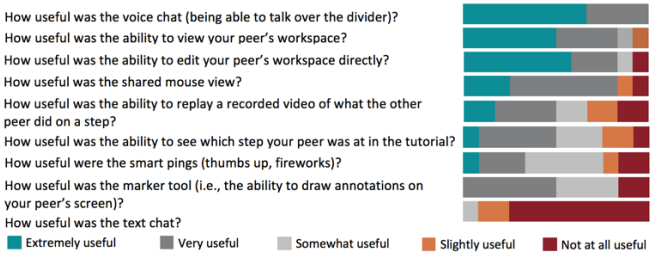

Figure 12: Rating of individual TwoTorials features.

6.4 Participant Feedback and Observations

At the end of the study session, we asked participants to contrast the experience of working with TwoTorials and the baseline tutorial system. Qualitative data were analyzed using methods drawn from grounded theory [33]. Specifically, open coding was used to label the data and emerging themes were identified by the first author and then shared and discussed with the broader research team.

6.4.1 Improved Communication, Awareness, and Coordination

Participants reported being able to coordinate with each other more effectively using the TwoTorials system, with smoother information flow between peers. Participants noted that having a constant view into their peer's workspace helped them solve problems more effectively without breaking the flow of working on the tutorial:

> Having the constant visual of my peer helped quite a bit to solve common problems on my workflow instead of having to stop the flow to find the assistance. (P11)

Participants also appreciated the ease with which they could switch from viewing their own workspace to that of their peer:

> The live view of your companion was a big plus. Easily being able to switch to their view and affect their workspace is a big plus as well. (P12)

Participants reported that TwoTorials helped them to maintain an ongoing awareness of the other user, and this helped to encourage dialog:

> The first system [TwoTorials] reminded me to think about discussing, because the view of the other screen was always present […] it helped slightly by encouraging dialog. (P9)

Participants also described using the shared awareness features to ground their discussions with their peer:

> It helped to see where the person was so we could say "look at my screen this is what you're supposed to have." (P6)

6.4.2 A Cooperative Learning Environment

A second common theme was that the TwoTorials features created an environment where cooperative learning was supported. Along these lines, one feature cited by participants was the ability to directly edit their peer's workspace. We observed several occasions where one peer would provide help by directly making changes in the workspace of their peer. Participants reported that this was an efficient way to help each other:

> The fact that I could work directly on my peer's workspace in [TwoTorials], let me help him more efficiently. (P7)

Participants also expressed appreciation for the annotation features, and highlighted how it created more of a "lesson experience" than a tutorial:

> In [TwoTorials], the fact that my peer could chime in and add his notes in real time made it more of a lesson experience than a tutorial - the chance to clarify and question each other as we followed the steps was a very useful addition. (P11)

This quote is particularly encouraging because it suggests the features of the TwoTorials system were able to change the experience to one where cooperation and helping each other was more natural. Along similar lines, P8 suggested that TwoTorials could be used in formal educational settings to enable teacher-student interactions:

> In [TwoTorials], getting help was much easier. I would imagine a TA or teacher helping students through that system. (P8)

Participants also commented that they took advantage of the expertise of their peer less in the baseline condition:

> If I got stuck, the person knew exactly where I was (they were there too or had just been there) and most likely had the same problems. I used the person less [in the baseline condition]. (P6)

Overall, this feedback provides validation that the TwoTorials features encouraged cooperation and helped to create an environment that supports cooperative learning.

6.4.3 Motivating and Enjoyable Experience

Finally, participants reported enjoying the cooperative tutorial experience (in both conditions), and found it to be engaging:

> Working cooperatively was fun and kept me engaged. Also, I learned some tips from the other person. (P9)

While participants reported enjoying the experience of cooperating in both tutorials, some participants noted that TwoTorials enhanced this aspect of the experience:

> In [TwoTorials], the second layer of interaction added a different [kind] of enjoyment, where we could interact and made the experience more fun. (P11)

A specific feature cited as creating an enjoyable experience was the peer pings. Four participants stated that they felt the peer pings were fun and helped encourage them to cooperate:

> "chat icons" were a nice touch to encourage each other. (P11)

> There was a sense of competition that reduced co-operative work in both tutorials. This was less so in [TwoTorials] because of the added features like thumbs up etc. (P8)

This final quote is particularly encouraging, it suggests that peer pings were able to reduce the sense of competition between the peers, which could stand in the way of the cooperative experience the system is designed to foster.

6.5 Challenges Encountered

While participants were generally supportive of the features of TwoTorials, some features elicited mixed feelings. Specifically, the ability to directly modify content in a peer's workspace was cited as undesirable by some participants:

> I do not want to interfere with my partner's screen. Annotation can be helpful though, and stickers [peer pings] make it more fun, but not direct interaction. (P1)

> I did not feel comfortable editing my partner's workspace. (P9)

As we discuss in the next section, we believe this indicates the need for better social mechanisms to be built around these features, to ensure that they can only be used to provide help or edit a peer's workspace when that help is welcome, as suggested by prior work on collaboration boundaries [83].

More broadly than any individual feature, one of the participants expressed that he would prefer to work on his own, because he did not like being observed while he worked:

> I personally like working on a tutorial alone and having others watching my work is kind of irritating. (P8)

This is important feedback, but in practice we believe that those who are interested in cooperative learning will choose to use TwoTorials or other systems like it, while those who are not can continue to use the many resources currently available to support individual learning.

7 Discussion and Future Work

Overall, our evaluation indicated that TwoTorials helped participants to engage in cooperative learning, improved their performance, reduced effort and mental demand, and helped participants to maintain awareness of each other's progress in the tutorial. Feedback from participants also suggests that the system's features helped to create a supportive environment for cooperative learning, helped keep learner motivation high, and helped foster a feeling of cooperation rather than competition between the learners. These are promising findings for applying the cooperative software learning approach to step-by-step 3D design software tutorials.

While our study results are generally encouraging, we found that some participants did not appreciate the ability to directly allow peers to edit one another's workspaces. This is important feedback, particularly because this study was conducted with peers who knew each other as friends or colleagues – it seems likely that learners will be more hesitant about this feature if they were working with peers with whom they don't have an existing relationship. To overcome this challenge, we believe that simple

permission mechanisms could be put in place. For example, a user could be prevented from editing their peer's workspace unless that peer explicitly asks for help and provides editing permission. Editing permission could also be limited to a short period of time, or to a selected subset of objects in the workspace. This approach would fit with prior research on groupware and MOOCS, which suggests that each user should have their own territory [44], with permission and roles mechanisms to enable users to control who can view and edit [77, 83]. Alternately, the system could enable a "forked demonstrations" paradigm, where a user could get a copy of their peer's current workspace that they could edit to demonstrate an operation to their peer, without making any lasting change to the peer's workspace itself.

7.1 Matching with Remote Peers

In this paper we focused on investigating features that could enable a cooperative learning experience for distributed pairs of users working on step-by-step software tutorials. Having established the benefits of this approach, a next important question is how to match pairs of remote users to work together on tutorials. There are several interesting possibilities here. The results of our observational study suggest that it may not be a good idea to match users with large differences in overall experience and expertise, which could result in the more experienced user becoming bored. Instead, the system could try to match users who are at similar levels of experience but have complementary skill sets. It would be particularly interesting if the system could consider both the skills of the learners and the required skills for the tutorial, to create an experience where peers would need to work together and help one another to reach the goal. These skill-based matchmaking mechanics could be designed in a similar way to those available in multiplayer games [1, 66].

7.2 Additional Peers

Another interesting area for future work would be to consider how the cooperative software tutorial approach could accommodate more than two learners. An advantage of the approach we have adopted, where each user is working in parallel on the tutorials, is that it could naturally support additional peers – in contrast, if more than two people were working on one shared workspace, it could quickly become unwieldy. The advantage of adding additional peers is more collective expertise, which could help get the group of peers unstuck when they face challenges. However, this could also create additional conflicts between users, or situations where certain users pair off, leaving others out. These challenges make this an interesting area for investigation, and we see the potential for a scaled up system to be used as a component of interactive 3D design MOOCS [38, 44].

7.3 Beyond 3D Design Software

Although we focused on step-by-step tutorials for 3D design software, we believe that the features of TwoTorials could be easily adapted to work in other software domains with a strong visual element, such as photo editing or the creation of games using game engines (e.g., Unity). From a technical standpoint, our system could be used with minimal modifications with any web-based software application.

7.4 Limitations

This work adds to a growing body of research on software learning (e.g., [13, 36, 51, 59, 78]) and provides insights into how step-by-step tutorial systems can be adapted to support remote cooperative learning. However, there are several limitations to this work which should be addressed in future research. First, our study was conducted with a small, specific sample (employees of a software company), which may limit the generalizability of the findings. A good next step would be to deploy TwoTorials in an online 3D design course, with remote students. Second, TwoTorials was compared against a baseline that offered minimal coordination features. This was intentional, in order to reveal which of TwoTorials' features were most useful to support collaboration, but future work should compare these features to those offered in state-of-the-art online collaborative learning solutions, such as free-form web curation tools [44, 61]. Third, prior research has shown that ethnocultural norms and backgrounds can influence the effectiveness of cooperative learning [49, 60, 86], so it is important to expand the evaluation of this type of system to a much larger and more diverse set of participants. Finally, we did not collect data on the long-term effects or value of our system in sustaining learner motivation or encouraging more extensive learning of a domain, which would be an interesting avenue for future work.

8 CONCLUSION

This work has demonstrated an approach and a set of features for creating cooperative remote software tutorial systems. Our findings indicate that participants enjoy the cooperative learning experience that this approach enables. Overall, we see this work as a first step toward a future where anyone, anywhere can gain the learning benefits of working alongside peers on interesting and engaging projects.

REFERENCES

1. Sharad Agarwal and Jacob R. Lorch. 2009. Matchmaking for Online Games and Other Latency-sensitive P2P Systems. In *Proceedings of the ACM SIGCOMM 2009 Conference on Data Communication* (SIGCOMM '09), 315–326. https://doi.org/10.1145/1592568.1592605
2. Sultan A. Alharthi, Ruth C. Torres, Ahmed S. Khalaf, Zachary O. Toups, Igor Dolgov, and Lennart E. Nacke. 2018. Investigating the Impact of Annotation Interfaces on Player Performance in Distributed Multiplayer Games. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (CHI '18), 314:1-314:13. https://doi.org/10.1145/3173574.3173888
3. Wahab Ali. 2020. Online and Remote Learning in Higher Education Institutes: A Necessity in light of COVID-19 Pandemic. *Higher Education Studies* 10, 3: 16. https://doi.org/10.5539/hes.v10n3p16
4. Autodesk, Inc. 2018. *Tinkercad*. Retrieved September 17, 2018 from https://www.tinkercad.com/
5. Davida Bloom. 2009. Collaborative Test Taking: Benefits for Learning and Retention. *College Teaching* 57, 4: 216–220. https://doi.org/10.1080/87567550903218646
6. Chantelle Bosch, Elsa Mentz, and Gerda Reitsma. 2020. Cooperative Learning as a Blended Learning Strategy: A Conceptual Overview. *Emerging Techniques and Applications for Blended Learning in K-20 Classrooms*, 65–87. https://doi.org/10.4018/978-1-7998-0242-6.ch004
7. Andrea Bunt, Patrick Dubois, Ben Lafreniere, Michael A. Terry, and David T. Cormack. 2014. TaggedComments: Promoting and Integrating User Comments in Online Application Tutorials. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '14), 4037–4046. https://doi.org/10.1145/2556288.2557118

8. John M. Carroll and Mary Beth Rosson. 1987. *Paradox of the Active User*.
9. Robyn Carston. 1999. Herbert H. Clark, Using language. Cambridge: Cambridge University Press, 1996. Pp. xi+432. *Journal of Linguistics* 35, 1: 167–222. Retrieved September 17, 2018 from https://www.cambridge.org/core/journals/journal-of-linguistics/article/herbert-h-clark-using-language-cambridge-cambridge-university-press-1996-pp-xi432/05AF0B5CBF76D03CBDB1DC4BFA648654
10. Centers for Disease Control and Prevention. Coronavirus Disease 2019. Retrieved September 5, 2020 from https://www.cdc.gov/coronavirus/2019-ncov/index.html
11. Yan Chen, Steve Oney, and Walter S. Lasecki. 2016. Towards Providing On-Demand Expert Support for Software Developers. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (CHI '16), 3192–3203. https://doi.org/10.1145/2858036.2858512
12. Victor Cheung, Y.-L. Betty Chang, and Stacey D. Scott. 2012. Communication Channels and Awareness Cues in Collocated Collaborative Time-critical Gaming. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work* (CSCW '12), 569–578. https://doi.org/10.1145/2145204.2145291
13. Pei-Yu Chi, Sally Ahn, Amanda Ren, Mira Dontcheva, Wilmot Li, and Björn Hartmann. 2012. MixT: Automatic Generation of Step-by-step Mixed Media Tutorials. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology* (UIST '12), 93–102. https://doi.org/10.1145/2380116.2380130
14. P. K. Chilana, N. Hudson, S. Bhaduri, P. Shashikumar, and S. Kane. 2018. Supporting Remote Real-Time Expert Help: Opportunities and Challenges for Novice 3D Modelers. In *2018 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 157–166. https://doi.org/10.1109/VLHCC.2018.8506568
15. Parmit K. Chilana, Andrew J. Ko, and Jacob O. Wobbrock. 2012. LemonAid: Selection-based Crowdsourced Contextual Help for Web Applications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '12), 1549–1558. https://doi.org/10.1145/2207676.2208620
16. Atef Chorfi, Djalal Hedjazi, Sofiane Aouag, and Djalleleddine Boubiche. 2020. Problem-based collaborative learning groupware to improve computer programming skills. *Behaviour & Information Technology* 0, 0: 1–20. https://doi.org/10.1080/0144929X.2020.1795263
17. Laura Dabbish and Robert E. Kraut. 2004. Controlling Interruptions: Awareness Displays and Social Motivation for Coordination. In *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work* (CSCW '04), 182–191. https://doi.org/10.1145/1031607.1031638
18. Sarah D'Angelo and Andrew Begel. 2017. Improving Communication Between Pair Programmers Using Shared Gaze Awareness. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (CHI '17), 6245–6290. https://doi.org/10.1145/3025453.3025573
19. Sarah D'Angelo and Darren Gergle. 2018. An Eye For Design: Gaze Visualizations for Remote Collaborative Work. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (CHI '18), 349:1-349:12. https://doi.org/10.1145/3173574.3173923
20. Sir John Daniel. Education and the COVID-19 pandemic. 6.
21. Sebastian Deterding, Dan Dixon, Rilla Khaled, and Lennart Nacke. 2011. From Game Design Elements to Gamefulness: Defining "Gamification." In *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments* (MindTrek '11), 9–15. https://doi.org/10.1145/2181037.2181040
22. Pierre Dillenbourg. 1999. What do you mean by collaborative learning? *P. Dillenbourg (Ed) Collaborative-learning: Cognitive and Computational Approaches*.: 1–19.
23. Tao Dong, Mira Dontcheva, Diana Joseph, Karrie Karahalios, Mark Newman, and Mark Ackerman. 2012. Discovery-based Games for Learning Software. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '12), 2083–2086. https://doi.org/10.1145/2207676.2208358
24. Jeff Dyck and Carl Gutwin. 2002. Groupspace: A 3D Workspace Supporting User Awareness. In *CHI '02 Extended Abstracts on Human Factors in Computing Systems* (CHI EA '02), 502–503. https://doi.org/10.1145/506443.506450
25. Volodymyr Dziubak, Ben Lafreniere, Tovi Grossman, Andrea Bunt, and George Fitzmaurice. 2018. Maestro: Designing a System for Real-Time Orchestration of 3D Modeling Workshops. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology* (UIST '18), 287–298. https://doi.org/10.1145/3242587.3242606
26. Clarence A. Ellis, Simon J. Gibbs, and Gail Rein. 1991. Groupware: Some Issues and Experiences. *Commun. ACM* 34, 1: 39–58. https://doi.org/10.1145/99977.99987
27. Mica R. Endsley. 1995. Toward a theory of situation awareness in dynamic systems. *Human Factors* 37, 1: 32–64. https://doi.org/10.1518/001872095779049543
28. Jennifer Fernquist, Tovi Grossman, and George Fitzmaurice. 2011. Sketch-sketch Revolution: An Engaging Tutorial System for Guided Sketching and Application Learning. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology* (UIST '11), 373–382. https://doi.org/10.1145/2047196.2047245
29. Susan R. Fussell, Robert E. Kraut, and Jane Siegel. 2000. Coordination of Communication: Effects of Shared Visual Context on Collaborative Work. In *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work* (CSCW '00), 21–30. https://doi.org/10.1145/358916.358947
30. William W. Gaver, Abigail Sellen, Christian Heath, and Paul Luff. 1993. One is Not Enough: Multiple Views in a Media Space. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems* (CHI '93), 335–341. https://doi.org/10.1145/169059.169268
31. Philipp M. Gemmel, McKenna K. Goetz, Nicole M. James, Kate A. Jesse, and Britni J. Ratliff. 2020. Collaborative Learning in Chemistry: Impact of COVID-19. *Journal of Chemical Education*. https://doi.org/10.1021/acs.jchemed.0c00713
32. Darren Gergle, Robert E. Kraut, and Susan R. Fussell. 2013. Using Visual Information for Grounding and Awareness in Collaborative Tasks. *Human–Computer Interaction* 28, 1: 1–39. https://doi.org/10.1080/07370024.2012.678246
33. Barney G. Glaser and Anselm L. Strauss. 2009. *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Transaction Publishers.

34. Saul Greenberg. 1991. Personalizable groupware: Accommodating individual roles and group differences. In *Proceedings of the Second European Conference on Computer-Supported Cooperative Work ECSCW '91*, Liam Bannon, Mike Robinson and Kjeld Schmidt (eds.). Springer Netherlands, Dordrecht, 17–31. https://doi.org/10.1007/978-94-011-3506-1_2
35. Tovi Grossman and George Fitzmaurice. 2010. ToolClips: An Investigation of Contextual Video Assistance for Functionality Understanding. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '10), 1515–1524. https://doi.org/10.1145/1753326.1753552
36. Tovi Grossman, George Fitzmaurice, and Ramtin Attar. 2009. A Survey of Software Learnability: Metrics, Methodologies and Guidelines. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '09), 649–658. https://doi.org/10.1145/1518701.1518803
37. Tovi Grossman, Justin Matejka, and George Fitzmaurice. 2010. Chronicle: Capture, Exploration, and Playback of Document Workflow Histories. In *Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology* (UIST '10), 143–152. https://doi.org/10.1145/1866029.1866054
38. Philip J. Guo, Juho Kim, and Rob Rubin. 2014. How Video Production Affects Student Engagement: An Empirical Study of MOOC Videos. In *Proceedings of the First ACM Conference on Learning @ Scale Conference* (L@S '14), 41–50. https://doi.org/10.1145/2556325.2566239
39. Carl Gutwin and Saul Greenberg. 1998. Design for Individuals, Design for Groups: Tradeoffs Between Power and Workspace Awareness. In *Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work* (CSCW '98), 207–216. https://doi.org/10.1145/289444.289495
40. Carl Gutwin, Saul Greenberg, and Mark Roseman. 1996. Workspace Awareness in Real-Time Distributed Groupware: Framework, Widgets, and Evaluation. In *People and Computers XI*, 281–298.
41. Carl Gutwin, Mark Roseman, and Saul Greenberg. 1996. A Usability Study of Awareness Widgets in a Shared Workspace Groupware System. In *Proceedings of the 1996 ACM Conference on Computer Supported Cooperative Work* (CSCW '96), 258–267. https://doi.org/10.1145/240080.240298
42. Carl Gutwin, Gwen Stark, and Saul Greenberg. 1995. Support for Workspace Awareness in Educational Groupware. In *The First International Conference on Computer Support for Collaborative Learning* (CSCL '95), 147–156. https://doi.org/10.3115/222020.222126
43. William A. Hamilton, Oliver Garretson, and Andruid Kerne. 2014. Streaming on Twitch: Fostering Participatory Communities of Play Within Live Mixed Media. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems* (CHI '14), 1315–1324. https://doi.org/10.1145/2556288.2557048
44. William A. Hamilton, Nic Lupfer, Nicolas Botello, Tyler Tesch, Alex Stacy, Jeremy Merrill, Blake Williford, Frank R. Bentley, and Andruid Kerne. 2018. Collaborative Live Media Curation: Shared Context for Participation in Online Learning. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (CHI '18), 555:1-555:14. https://doi.org/10.1145/3173574.3174129
45. Sandra G. Hart. 2006. Nasa-Task Load Index (NASA-TLX); 20 Years Later. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 50, 9: 904–908. https://doi.org/10.1177/154193120605000909
46. Sandra G. Hart and Lowell E. Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In *Advances in Psychology*, Peter A. Hancock and Najmedin Meshkati (eds.). North-Holland, 139–183. https://doi.org/10.1016/S0166-4115(08)62386-9
47. Justin B. Houseknecht and Lucy K. Bates. 2020. Transition to Remote Instruction Using Hybrid Just-in-Time Teaching, Collaborative Learning, and Specifications Grading for Organic Chemistry 2. *Journal of Chemical Education*. https://doi.org/10.1021/acs.jchemed.0c00749
48. Nathaniel Hudson, Benjamin Lafreniere, Parmit K. Chilana, and Tovi Grossman. 2018. Investigating How Online Help and Learning Resources Support Children's Use of 3D Design Software. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (CHI '18), 257:1-257:14. https://doi.org/10.1145/3173574.3173831
49. Eric A. Hurley, Brenda A. Allen, and A. Wade Boykin. 2009. Culture and the Interaction of Student Ethnicity with Reward Structure in Group Learning. *Cognition and Instruction* 27, 2: 121–146. https://doi.org/10.1080/07370000902797346
50. Nen-Chen Richard Hwang, Gladie Lui, and Marian Yew Jen Wu Tong. 2008. Cooperative Learning in a Passive Learning Environment: A Replication and Extension. *Issues in Accounting Education* 23, 1: 67–75. https://doi.org/10.2308/iace.2008.23.1.67
51. Nikhita Joshi, Justin Matejka, Fraser Anderson, Tovi Grossman, and George Fitzmaurice. 2020. MicroMentor: Peer-to-Peer Software Help Sessions in Three Minutes or Less. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (CHI '20), 1–13. https://doi.org/10.1145/3313831.3376230
52. Juho Kim, Philip J. Guo, Daniel T. Seaton, Piotr Mitros, Krzysztof Z. Gajos, and Robert C. Miller. 2014. Understanding In-video Dropouts and Interaction Peaks Inonline Lecture Videos. In *Proceedings of the First ACM Conference on Learning @ Scale Conference* (L@S '14), 31–40. https://doi.org/10.1145/2556325.2566237
53. Robert E. Kraut, Darren Gergle, and Susan R. Fussell. 2002. The Use of Visual Information in Shared Visual Spaces: Informing the Development of Virtual Co-presence. In *Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work* (CSCW '02), 31–40. https://doi.org/10.1145/587078.587084
54. Robert E. Kraut, Mark D. Miller, and Jane Siegel. 1996. Collaboration in Performance of Physical Tasks: Effects on Outcomes and Communication. In *Proceedings of the 1996 ACM Conference on Computer Supported Cooperative Work* (CSCW '96), 57–66. https://doi.org/10.1145/240080.240190
55. Hideaki Kuzuoka. 1992. Spatial Workspace Collaboration: A SharedView Video Support System for Remote Collaboration Capability. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '92), 533–540. https://doi.org/10.1145/142750.142980
56. Ben Lafreniere, Andrea Bunt, Matthew Lount, and Michael Terry. 2013. Understanding the Roles and Uses of Web Tutorials. In *Seventh International AAAI Conference on*

Weblogs and Social Media. Retrieved September 18, 2018 from https://www.aaai.org/ocs/index.php/ICWSM/ICWSM13/paper/view/6094

57. Ben Lafreniere and Tovi Grossman. 2018. Blocks-to-CAD: A Cross-Application Bridge from Minecraft to 3D Modeling. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology* (UIST '18), 637–648. https://doi.org/10.1145/3242587.3242602
58. Benjamin Lafreniere, Tovi Grossman, and George Fitzmaurice. 2013. Community Enhanced Tutorials: Improving Tutorials with Multiple Demonstrations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '13), 1779–1788. https://doi.org/10.1145/2470654.2466235
59. Wei Li, Tovi Grossman, and George Fitzmaurice. 2014. CADament: A Gamified Multiplayer Software Tutorial System. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '14), 3369–3378. https://doi.org/10.1145/2556288.2556954
60. G. William Lucker, David Rosenfield, Jev Sikes, and Elliot Aronson. 1976. Performance in the Interdependent Classroom: A Field Study. *American Educational Research Journal* 13, 2: 115–123. https://doi.org/10.3102/00028312013002115
61. Nic Lupfer, Andruid Kerne, Andrew M. Webb, and Rhema Linder. 2016. Patterns of Free-form Curation: Visual Thinking with Web Content. 12–21. https://doi.org/10.1145/2964284.2964303
62. Jean MacMillan, Elliot E. Entin, and Daniel Serfaty. 2004. Communication overhead: The hidden cost of team cognition. In *Team cognition: Understanding the factors that drive process and performance*. American Psychological Association, Washington, DC, US, 61–82. https://doi.org/10.1037/10690-004
63. Arwen M. Marker and Amanda E. Staiano. 2014. Better Together: Outcomes of Cooperation Versus Competition in Social Exergaming. *Games for Health Journal* 4, 1: 25–30. https://doi.org/10.1089/g4h.2014.0066
64. David McConnell. 2000. *Implementing Computing Supported Cooperative Learning*. London : Kogan Page. Retrieved September 4, 2018 from https://hal.archives-ouvertes.fr/hal-00702948
65. Joanne M. McInnerney and Tim S. Roberts. 2009. Collaborative and Cooperative Learning. *Encyclopedia of Distance Learning, Second Edition*: 319–326. https://doi.org/10.4018/978-1-60566-198-8.ch046
66. Marçal Mora-Cantallops and Miguel-Ángel Sicilia. 2018. MOBA games: A literature review. *Entertainment Computing* 26: 128–138. https://doi.org/10.1016/j.entcom.2018.02.005
67. Benedikt Morschheuser, Alexander Maedche, and Dominic Walter. 2017. Designing Cooperative Gamification: Conceptualization and Prototypical Implementation. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing* (CSCW '17), 2410–2421. https://doi.org/10.1145/2998181.2998272
68. Benedikt Morschheuser, Marc Riar, Juho Hamari, and Alexander Maedche. 2017. How games induce cooperation? A study on the relationship between game features and we-intentions in an augmented reality game. *Computers in Human Behavior* 77: 169–183. https://doi.org/10.1016/j.chb.2017.08.026
69. Ahmed E. Mostafa, Kori Inkpen, John C. Tang, Gina Venolia, and William A. Hamilton. 2016. SocialStreamViewer: Guiding the Viewer Experience of Multiple Streams of an Event. In *Proceedings of the 19th International Conference on Supporting Group Work* (GROUP '16), 287–291. https://doi.org/10.1145/2957276.2957286
70. Michael P. A. Murphy. 2020. COVID-19 and emergency eLearning: Consequences of the securitization of higher education for post-pandemic pedagogy. *Contemporary Security Policy* 41, 3: 492–505. https://doi.org/10.1080/13523260.2020.1761749
71. Martin A. Nowak and Karl Sigmund. 2000. Cooperation versus Competition. *Financial Analysts Journal* 56, 4: 13–22. Retrieved September 17, 2018 from https://www.jstor.org/stable/4480255
72. Jennifer K. Olsen, Daniel M. Belenky, Vincent Aleven, and Nikol Rummel. 2014. Using an Intelligent Tutoring System to Support Collaborative as well as Individual Learning. In *Intelligent Tutoring Systems* (Lecture Notes in Computer Science), 134–143. https://doi.org/10.1007/978-3-319-07221-0_16
73. Jennifer K Olsen, Nikol Rummel, and Vincent Aleven. Learning Alone or Together? A Combination Can Be Best! *The 12th International Conference on Computer Supported Collaborative Learning*: 8.
74. Jiazhi Ou, Susan R. Fussell, Xilin Chen, Leslie D. Setlock, and Jie Yang. 2003. Gestural Communication over Video Stream: Supporting Multimodal Interaction for Remote Collaborative Physical Tasks. In *Proceedings of the 5th International Conference on Multimodal Interfaces* (ICMI '03), 242–249. https://doi.org/10.1145/958432.958477
75. Theodore Panitz. 1999. The Motivational Benefits of Cooperative Learning. *New Directions for Teaching and Learning* 1999, 78: 59–67. https://doi.org/10.1002/tl.7806
76. Paul John Peña and Dickson Lim. 2020. *Learning With Friends: A Rational View of Remote Learning with Network Externalities in the Time of COVID-19*. Social Science Research Network, Rochester, NY. Retrieved September 5, 2020 from https://papers.ssrn.com/abstract=3621056
77. David Pinelle, Mutasem Barjawi, Miguel Nacenta, and Regan Mandryk. 2009. An Evaluation of Coordination Techniques for Protecting Objects and Territories in Tabletop Groupware. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '09), 2129–2138. https://doi.org/10.1145/1518701.1519025
78. Suporn Pongnumkul, Mira Dontcheva, Wilmot Li, Jue Wang, Lubomir Bourdev, Shai Avidan, and Michael F. Cohen. 2011. Pause-and-play: Automatically Linking Screencast Video Tutorials with Applications. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology* (UIST '11), 135–144. https://doi.org/10.1145/2047196.2047213
79. Jiten Rama and Judith Bishop. 2006. A Survey and Comparison of CSCW Groupware Applications. In *Proceedings of the 2006 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on IT Research in Developing Countries* (SAICSIT '06), 198–205. https://doi.org/10.1145/1216262.1216284

80. Vernellia R. Randall. 1999. Increasing Retention and Improving Performance: Practical Advice on Using Cooperative Learning in Law Schools. *Thomas M. Cooley Law Review* 16: 201. Retrieved from https://heinonline.org/HOL/Page?handle=hein.journals/tmclr16&id=235&div=&collection=
81. Justin Reich, Christopher J. Buttimer, Alison Fang, Garron Hillaire, Kelley Hirsch, Laura Larke, Joshua Littenberg-Tobias, Roya Madoff Moussapour, Alyssa Napier, Meredith Thompson, and Rachel Slama. 2020. *Remote Learning Guidance From State Education Agencies During the COVID-19 Pandemic: A First Look*. EdArXiv. https://doi.org/10.35542/osf.io/437e2
82. Eduardo Salas, Terry L. Dickinson, Sharolyn A. Converse, and Scott I. Tannenbaum. 1992. Toward an understanding of team performance and training. In *Teams: Their training and performance*. Ablex Publishing, Westport, CT, US, 3–29.
83. Stacey D. Scott, M. Sheelagh T. Carpendale, and Kori M. Inkpen. 2004. Territoriality in Collaborative Tabletop Workspaces. In *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work* (CSCW '04), 294–303. https://doi.org/10.1145/1031607.1031655
84. Magy Seif El-Nasr, Bardia Aghabeigi, David Milam, Mona Erfani, Beth Lameman, Hamid Maygoli, and Sang Mah. 2010. Understanding and Evaluating Cooperative Games. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '10), 253–262. https://doi.org/10.1145/1753326.1753363
85. Shlomo Sharan. 1980. Cooperative Learning in Small Groups: Recent Methods and Effects on Achievement, Attitudes, and Ethnic Relations. *Review of Educational Research* 50, 2: 241–271. https://doi.org/10.3102/00346543050002241
86. Robert E. Slavin. 1983. When does cooperative learning increase student achievement? *Psychological Bulletin* 94, 3: 429–445. https://doi.org/10.1037/0033-2909.94.3.429
87. Zachary O. Toups, Jessica Hammer, William A. Hamilton, Ahmad Jarrah, William Graves, and Oliver Garretson. 2014. A Framework for Cooperative Communication Game Mechanics from Grounded Theory. In *Proceedings of the First ACM SIGCHI Annual Symposium on Computer-human Interaction in Play* (CHI PLAY '14), 257–266. https://doi.org/10.1145/2658537.2658681
88. Michael B. Twidale. 2005. Over the Shoulder Learning: Supporting Brief Informal Learning. *Computer Supported Cooperative Work (CSCW)* 14, 6: 505–547. https://doi.org/10.1007/s10606-005-9007-7
89. Deepika Vaddi, Zachary Toups, Igor Dolgov, Rina Wehbe, and Lennart Nacke. 2016. Investigating the Impact of Cooperative Communication Mechanics on Player Performance in Portal 2. In *Proceedings of the 42Nd Graphics Interface Conference* (GI '16), 41–48. https://doi.org/10.20380/GI2016.06
90. S. Valin, A. Francu, H. Trefftz, and I. Marsic. 2001. Sharing viewpoints in collaborative virtual environments. In *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, 12 pp.-. https://doi.org/10.1109/HICSS.2001.926213
91. Katherine M. Van Heuvelen, G. William Daub, and Hal Van Ryswyk. 2020. Emergency Remote Instruction during the COVID-19 Pandemic Reshapes Collaborative Learning in General Chemistry. *Journal of Chemical Education*. https://doi.org/10.1021/acs.jchemed.0c00691
92. Jason Wuertz, Sultan A. Alharthi, William A. Hamilton, Scott Bateman, Carl Gutwin, Anthony Tang, Zachary Toups, and Jessica Hammer. 2018. A Design Framework for Awareness Cues in Distributed Multiplayer Games. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (CHI '18), 243:1-243:14. https://doi.org/10.1145/3173574.3173817