Streamers Teaching Programming, Art, and Gaming: Cognitive Apprenticeship, Serendipitous Teachable Moments, and Tacit Expert Knowledge

Ian Drosos
UC San Diego
La Jolla, CA, USA
idrosos@ucsd.edu

Philip J. Guo
UC San Diego
La Jolla, CA, USA
pg@ucsd.edu

***Abstract*—Livestreaming is now a popular way for programmers, artists, and gamers to teach their craft online. In this paper we propose the idea that streaming can enable cognitive apprenticeship, a form of teaching where an expert works on authentic tasks while thinking aloud to explain their creative process. To understand how streamers teach in this naturalistic way, we performed a content analysis of 20 stream videos across four popular categories: web development, data science, digital art, and gaming. We discovered four kinds of serendipitous teachable moments that are reminiscent of cognitive apprenticeship: 1) creators encountered unexpected errors that led to improvised problem solving, 2) they generated improvised examples on-the-fly, 3) they sometimes went on insightful tangents, 4) they paused to give high-level advice that was contextualized within the work they were currently performing. We also found missed opportunities for additional teachable moments due to creators not being able to express their tacit (unspoken) expert knowledge because of pattern irreducibility, context dependence, and routinization.**

***Index Terms*—livestreaming, teaching, cognitive apprenticeship**

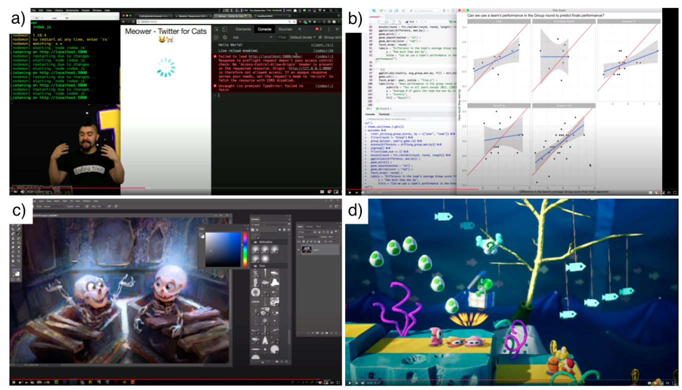

Fig. 1. We analyzed the content of 20 archived stream videos to characterize how streamers teach authentic technical workflows across four popular domains: a) web development, b) data science, c) digital art, d) gaming.

I. Introduction

Over the past decade there has been tremendous growth in livestreaming on platforms such as Twitch and YouTube [1]–[3]. While many streamers produce content for entertainment (e.g., playing video games), there is an increasing number who stream for educational purposes [4], [5]. For instance, programmer Suz Hinton does weekly livestreams showing herself working on open-source software and teaching programming concepts within the context of code that she is writing [6], [7]. These streams are often archived as videos on YouTube so that others can watch later (albeit without live interactions).

We refer to these kinds of videos as *instructional workflow streams* because they teach concepts within the context of a practitioner's naturalistic workflow, such as a programmer or data scientist writing code for their work or a digital artist using Photoshop. In this paper we propose the idea that instructional workflow streams are compelling from an educational perspective because they enable viewers to engage in a virtual form of *cognitive apprenticeship* [8], [9]: watching an expert practitioner working on their craft while thinking aloud to verbalize their creative and technical process. Cognitive apprenticeship captures an expert working on realistic tasks while "making thinking visible" [8] so that "learners can see the processes of work" [8].

To understand how streamers teach in this naturalistic way, we performed a content analysis of 20 instructional stream videos across four popular categories [1]: web development (design+programming), data science (programming), digital art, and gaming (see Figure 1) to highlight key features of instructional workflow streams. We made two new discoveries that have not been seen in prior work [1], [4], [5], [10]–[12]:

First we discovered four kinds of *serendipitous teachable moments* that are reminiscent of cognitive apprenticeship: 1) creators encountered unexpected errors that led to improvised problem solving, 2) they generated improvised examples on-the-fly, 3) they sometimes went on insightful tangents, 4) they paused to give high-level advice that was contextualized within the work they were performing.

Second, we found missed opportunities for additional teachable moments due to creators not being able to express their tacit (unspoken) expert knowledge [13], [14]. These arose in three ways: pattern irreducibility (cannot verbalize fine-grained details), context dependence, and routinization (actions become so routine that they are unconscious).

978-1-6654-4592-4/21/$31.00 ©2021 IEEE

This paper's contributions are:

- The idea that *instructional workflow streams* can act as a limited virtual form of cognitive apprenticeship [8].
- Analysis of 20 instructional workflow streams from four domains, framed via the lens of cognitive apprenticeship.

II. Related Work

Over the past decade, livestreaming has emerged as a popular form of online media where content creators stream their naturalistic activities on platforms such as Twitch and YouTube. The most popular kinds of streamed activities include gaming, digital art, design, programming, physical crafts, music, calligraphy, and casual socializing [1], [3], [4], [11], [15], [16]. HCI researchers have developed tools to facilitate live viewer interactions with streamers via multimedia-enhanced chat [17]–[19] and curated chat summaries [5], [20].

Streamers are motivated by goals such as entertainment, socializing, community-building, cultural heritage, financial interests, and teaching [1], [5], [15], [16]. For our study, we focus on the subset of streams that are purposely created with educational intent in mind: We call these *instructional workflow streams* since they come from a streamer demonstrating their naturalistic workflow with the intent of teaching skills to others. (A *non*-instructional stream is someone playing a video game or making music without any instructional commentary.)

Prior research on educational streams used surveys and interviews to discover the motivations of and challenges faced by streamers and viewers [1], [4], [5], [10]–[12]. We extend this emerging line of research by being the first, to our knowledge, to perform an in-depth analysis of the pedagogical content within stream videos. The closest related studies are of programming streamers, which corroborate parts of our findings using complementary approaches (e.g., interviews and surveys). Chen et al. found that viewers enjoyed learning "over-the-shoulder" by hearing experts articulate their thought processes out loud and even seeing streamers make mistakes [11]. Unlike Chen et al., we focus on analyzing the elements found within the produced artifacts (archived streams), rather than eliciting knowledge from the streamers themselves. Alaboudi and LaToza [10] and Faas et al. [4] analyzed a sample of archived stream videos and reported instances of viewers providing debugging help in the live chat, along with general knowledge transfer of programming concepts. Haaranen reported similar kinds of Q&A in chats [21].

III. Methods

A. Video Selection

We chose four domains that span the gamut of technology-related streams that are popular on sites like Twitch and YouTube [1]: 1) *Web Development*, 2) *Data Science*, 3) *Digital Art*, and 4) *Gaming*. These cover a variety of technical skills including programming and debugging (Web Development and Data Science), open-ended creative processes (Data Science and Digital Art), manual hand dexterity (Digital Art and Gaming), and optimizing for speed (Gaming speedruns [22]).

TABLE I

The 20 instructional workflow streams we analyzed: WebDevelopment (W1–W5), DataScience (D1–D5), Art (A1–A5), Gaming (G1–G5). Each was from a different creator; 'subs' is number of subscribers in Apr 2021 (k=thousand, M=million).

ID	Length	Subs	Contents
W1	2:29:06	4.3k	Designing/coding a personal website [23]
W2	1:29:18	78k	Wireframing/coding a portfolio website [24]
W3	1:27:43	1.3M	Building a full-stack Twitter clone [25]
W4	2:40:29	1.7k	Building a real-time web app [26]
W5	25:57	59k	Deploying personal website to the cloud [27]
D1	1:15:48	11k	Analyze Women's World Cup data with R [28]
D2	1:18:24	25k	Data analysis for academic research [29]
D3	1:21:49	41k	Doing a Kaggle challenge [30]
D4	1:33:03	5.8k	Computational finance with Excel [31]
D5	1:29:52	155	Handling missing data during analysis [32]
A1	56:16	26k	Walkthrough of regimented art training [33]
A2	1:25:55	51k	Digital coloring with Procreate 4 for iPad [34]
A3	1:51:08	52k	Practice with drawing human eyes [35]
A4	36:56	409k	Halloween painting combining 3-D and 2-D [36]
A5	31:50	92k	Two artists on creating interior scenes [37]
G1	17:01	163k	Building a character in Path Of Exile [38]
G2	35:03	1.5k	Speedrunning the Celeste game [39]
G3	10:38	176k	Creating macros in Final Fantasy 14 [40]
G4	15:26	18k	Guide for playing FF14's Scholar character [41]
G5	47:44	4.7M	Gameplay walkthrough ('playthrough') [42]

B. Media Content Analysis using Cognitive Apprenticeship

We performed a media content analysis on these videos, which is a standard qualitative research technique in fields such as communications and media studies [43]. This method involves treating the media itself as the primary subject of study and interrogating its contents via a theoretical lens. We analyzed the content within these 20 videos through the lens of *cognitive apprenticeship* [8], [9], which is a process by which novices learn from an expert practitioner by observing them at work in a naturalistic environment. Thus, we watched these videos to look for pedagogically-relevant content by adopting the role of apprentices watching an expert at work.

Two members of the research team individually watched each video, collected as pairs in each domain at a time, and read the archived chats (if available). We looked for how creators 1) articulated domain knowledge in the context of their workflow, 2) demonstrated their problem-solving heuristics, and 3) reflected on metacognitive strategies, which are the three common facets of cognitive apprenticeship [8]. We each performed open coding and met regularly to iteratively reconcile our codes using an inductive analysis approach [44].

C. Study Design Limitations

Although we strove to include videos from a variety of technical domains, any sample is limited and not representative of all instructional workflow streams; thus, our findings are about high-level patterns that we observed across multiple domains. We also did not interview creators and instead relied only on content analysis of their videos. Thus, we can only infer intent based on what we directly see in videos. Note that even if we had interviewed these creators, they might themselves be subject to cognitive biases (e.g., hindsight bias) when reflecting on videos they made years ago.

IV. Serendipitous Teachable Moments

In addition to high-level observations that corroborate prior work, like stream composition [10] and exchanging technical knowledge [4], [10], [21], our study made two new sets of discoveries, which we describe here and in Section V. First, since instructional workflow streams are recorded naturalistically while a streamer is working on real tasks, they contain lots of improvised content that may not appear in a lecture or tutorial video but still may actually have educational value since it emulates what happens during cognitive apprenticeship [45]. We call these *serendipitous teachable moments* since they are unplanned moments when the creator teaches something new.

A. Improvised Problem Solving Due to Unexpected Errors

Errors gave creators a chance to do improvised problem solving. This lets viewers see an expert genuinely struggle to solve problems that they were not prepared to encounter. This level of transparency simulates aspects of cognitive apprenticeship: seeing experts debug and troubleshoot in authentic work scenarios. This impromptu problem-solving was most visible in programming streams (data science and web development) where creators had to engage in information foraging and debugging to solve their coding problems. They typically did web searches, visited Stack Overflow and API documentation, copied code snippets into their IDE, and tested them out. The longform video format enabled them to take their time and show their entire process, even when it took unexpected turns. For instance, W5 encountered an error while deploying a live website to the cloud and said, *"Now the .htaccess file is screwed up, we might as well fix that too ... we just gotta do it. We got to get our local environment set up properly here."*

Seeing experts make and recover from mistakes can help learners get a sense of how creative work can be messy; one benefit of cognitive apprenticeship is that "learners can see the processes of work" [8]. Some fixed their mistakes and re-attempted, such as artist A1 acknowledging their mistake verbally then re-drawing a sketch layer: *"[laughs and deletes layer] let me do a better job with it [re-makes layer]."*

Errors can also make creators more relatable since viewers can see that experts also make mistakes as part of their normal workflow. At the end of their stream, W5 apologetically said, *"So sorry that was such a mess, you know in a perfect world I'd redo it and not make any mistakes and make this perfect workflow thing."* But the top-voted YouTube viewer comment below that video was: *"Watching you hilariously deal with typical everyday issues is the best part."*

Finally, the audience can participate in collaborative problem-solving with the streamer via the chat. For instance, when G3 is creating a custom gameplay macro, the audience notices some typos, which he fixes right away. the audience shares in the excitement of the working macro and further encourages the streamer. And in W2, W3, and W4 the audience helps to debug the web application code via text chat. The most engaged participation came in D2, where the data scientist asked the live audience to look at the data set together to try to spot anomalies that might affect the analysis. Several studies have documented how viewers can help programmers debug via chat [4], [10], but our findings show how collaborative problem-solving goes beyond debugging.

B. Generating Improvised Examples

Creators all work through a long-running main task in their videos, but they sometimes improvise to generate additional examples that are not part of their plan. These improvised examples can also serve as serendipitous teachable moments.

When explaining a concept as they are working, some creators make up mini-examples to demonstrate it more concretely. For instance, artist A1 spontaneously sketched a few *bad* pencil drawings when talking about common pitfalls that novices face. And when demonstrating a certain detail about drawing irises in eyes, A3 decides to zoom the camera in on his eyes and moves his eyes to provide physical examples.

Mistakes can also lead to improvised examples. For instance, D1 was confused that a data set did not contain some data that it should, so he wrote web scraping code to grab that data online. After he finished, he realized that the data he wanted was there all along in another table in his data set, so his improvised web scraping code was unnecessary. But he still left that code in his R script to serve as a self-contained example to demonstrate web scraping. When he realized his mistake, he said in a surprised tone, *"Oh wow! Whoops that was fun but I think I totally missed [this other data table], but I'm gonna leave this code in as a web scraping example."*

Viewer chat questions can also inspire creators to improvise new examples. This technique exemplifies "show, don't tell" since creators can craft a small example to respond to questions rather than just directly answering. For instance, a viewer notices data scientist D2 running a `pushd` command in the macOS terminal and asks a chat question about that command; D2 explains by improvising an example of using a pair of `pushd` and `popd` commands on the terminal to navigate through a Unix-like filesystem hierarchy on macOS.

C. Insightful Tangents

Creators sometimes go off on tangents that are unrelated to the main focus of the workflow they are demonstrating. For instance, while editing code in the Vim editor, W4 reflexively does a quick keyboard shortcut to delete an entire block of code within parentheses. He realizes that this may be interesting to some, so he goes on a brief tangent to explain how that keyboard shortcut works.

Some tangents have a more direct pedagogical purpose. For instance, D1 noticed that one of the Women's World Cup soccer game scores was 13-0, which seemed unusual, so he searched for the news on Google News to make sure that the score was accurate; then he spent a bit of time browsing news articles containing relevant sports statistics. While somewhat tangential, this action demonstrates a good expert habit of checking for data quality when outliers are present.

Many tangents also arose from viewer chat questions. For instance, in programming streams the audience may ask about what certain functions or API calls in the code do. When the creator tries to answer, they sometimes go off on a tangent to search the web for related resources to show the audience.

D. Contextualized High-Level Advice

We also noticed that creators periodically pause their work to give high-level advice about their craft.

For instance, after spending some time investigating missing data, D2 pauses to tell the livestream audience why it is important to spend the time upfront doing this tedious data science task: *"Like Elizabeth said, it's painful to chase N's [finding out why some data is missing] but really that's where I discover all the basic problems. I could have just gone into the data, just start reading it in, and you would have been like at the point of publishing a paper before you realized that maybe you were missing 20 million people [in your analysis]."*

Aside from general technical advice, creators also mentioned the importance of developing certain *expert mindsets*. For example, A2 advised learners that they *"should always be thinking in 3-D all the time, which takes a lot of practice"*. D1 mentioned the importance of domain-specific knowledge in becoming an expert data scientist – i.e., that simply knowing tools is not enough. W1 mentioned how experts often look for web design inspiration from other websites.

Another kind of high-level advice is experts commiserating with novices about common struggles. For instance, G3 reflects on aspects that he dislikes about gameplay macros that novices create but says that he also made those mistakes when he was new: *"I totally understand how new players think these things could help them when it actually hampers them."*

Finally, the live audience sometimes inspired the creator to engage in higher-level reflections in response to chat questions. For instance, a viewer asked A3 about manga-style drawings of faces (even though A3 was not drawing manga), which led to a high-level discussion about the philosophy of manga art.

V. Missed teachable moments: Tacit Knowledge

We also found missed opportunities for additional teachable moments. Some of these occurrences may be due to creators being unable to communicate their *tacit knowledge* [13], [14], [46] while streaming. As Polanyi states in his definition of tacit knowledge, *"We can know more than we can tell"* [14]. Experts are often not able to verbalize their expertise since the complex actions they take feel intuitive and unconscious to them. Elaborating on that definition, Horvath et al. [46] identified three reasons why knowledge remains tacit: 1) pattern irreducibility, 2) context dependence, and 3) routinization.

A. Pattern Irreducibility

Horvath et al. define *pattern irreducibility* as the phenomenon that "some knowledge concerns information patterns that cannot be reduced to rules or generalizations [...] such configurations may be easier to recognize than to define concisely." [46] For instance D1 coded up a few exploratory data visualizations to get a better "feel" for the shape of the Women's World Cup data set after looking at some of the raw data tables. He quickly went through a few visualizations without explaining his rationale for picking those ones out of the vast array of possible visualizations and parameter settings he could have chosen. It is easy for an expert data scientist to recognize certain data properties and intuit what visualizations may be the most fruitful in each scenario, but it is hard to verbalize those intuitions into hard-and-fast rules for novices.

B. Context Dependence

Some expert knowledge remains unspoken because it is highly context-dependent, which makes it hard to verbalize as generally-applicable advice. Thus, when asked whether to apply certain strategies in a given scenario, experts often say something like *"well, it depends."* This came up frequently in gaming videos since different players' gameplay contexts may vary greatly even if they are playing the same level. For instance, G4 mentioned that the strategy they show on-screen is very situationally-dependent and unlikely to be a universal rule that all players should follow. G2 even admits that it is hard to teach someone how to succeed in a particular boss fight in the game except for "winging it" based on the situation.

C. Routinization

A third type of tacit knowledge refers to actions that become so routine that experts perform them unconsciously. This came up most frequently in the art and gaming domains, which both involve large amounts of manual dexterity and muscle memory that can be hard to put into words. For instance, A3 tells viewers that while there are many techniques for drawing the human eye, what they suggest most strongly is to simply practice more since that is the only way to build muscle memory: *"If you want to get better at drawing eyes, draw 100 of them."* Gaming also requires mastering muscle memory due to the rapid pace and precision required to enter controller inputs to successfully navigate the game. When teaching speedrunning [22], G2 says *"the first maze in this section is really difficult to do quickly, so just use dashes PROPERLY and get through it as fast as you can."* G2 does not explain what it means to use dashes "properly" but repeats throughout the video that practicing is the only way to develop muscle memory for mastering this movement technique.

VI. Conclusion: Cognitive Apprenticeship@Scale

Over the past decade the volume of online learning resources has grown so vast to the point that a large array of domain-specific knowledge is freely available online. Novices can now pick up tons of facts and basic skills on-demand. However, what is much harder to learn are authentic and contextualized expert work practices that cannot be easily taught in textbooks or tutorial websites. This paper demonstrates that instructional workflow streams are a promising medium for transmitting such experiential knowledge via a limited virtual form of cognitive apprenticeship. We view this study as a first step toward informing the design of new tools to enable cognitive apprenticeship at scale.

VII. Acknowledgments

Thanks to Sean Kross and Sam Lau for their feedback on earlier drafts. This material is based upon work supported by the National Science Foundation under Grant No. NSF IIS-1845900.

References

[1] C. A. Fraser, J. O. Kim, A. Thornsberry, S. Klemmer, and M. Dontcheva, "Sharing the studio: How creative livestreaming can inspire, educate, and engage," in *Proceedings of the 2019 on Creativity and Cognition*, ser. C&C '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 144–155. [Online]. Available: https://doi.org/10.1145/3325480.3325485

[2] W. A. Hamilton, O. Garretson, and A. Kerne, *Streaming on Twitch: Fostering Participatory Communities of Play within Live Mixed Media*, ser. CHI '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 1315–1324. [Online]. Available: https://doi.org/10.1145/2556288.2557048

[3] M. Sjöblom and J. Hamari, "Why do people watch others play video games? an empirical study on the motivations of twitch users," *Computers in Human Behavior*, vol. 75, no. C, pp. 985–996, 2017. [Online]. Available: https://doi.org/10.1016/j.chb.2016.10.019

[4] T. Faas, L. Dombrowski, A. Young, and A. D. Miller, "Watch me code: Programming mentorship communities on twitch.tv," *Proc. ACM Hum.-Comput. Interact.*, vol. 2, no. CSCW, Nov. 2018. [Online]. Available: https://doi.org/10.1145/3274319

[5] Z. Lu, S. Heo, and D. J. Wigdor, "Streamwiki: Enabling viewers of knowledge sharing live streams to collaboratively generate archival documentation for effective in-stream and post hoc learning," *Proc. ACM Hum.-Comput. Interact.*, vol. 2, no. CSCW, Nov. 2018. [Online]. Available: https://doi.org/10.1145/3274381

[6] S. Hinton, "Lessons from my first year of live coding on twitch," https://www.freecodecamp.org/news/lessons-from-my-first-year-of- live-coding-on-twitch-41a32e2f41c1/, 2017, accessed: 2021-04-15.

[7] S. Hinton, A. Stacoviak, and J. Santo, "Live coding open source on twitch," Mar. 2018, accessed: 2021-04-15. [Online]. Available: https://changelog.com/podcast/288

[8] A. Collins, J. S. Brown, and A. Holum, "Cognitive apprenticeship: Making thinking visible," *American Educator*, vol. 15, no. 3, pp. 6–11, 1991.

[9] D. A. Schon, *The reflective practitioner: How professionals think in action*. Basic books, 1984, vol. 5126.

[10] A. Alaboudi and T. D. LaToza, "An exploratory study of live-streamed programming," in *2019 IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC 2019, Memphis, Tennessee, USA, October 14-18, 2019*, ser. VL/HCC '19. IEEE Computer Society, 2019, pp. 5–13. [Online]. Available: https://doi.org/10.1109/VLHCC.2019.8818832

[11] Y. Chen, W. S. Lasecki, and T. Dong, "Towards supporting programming education at scale via live streaming," *Proc. ACM Hum.-Comput. Interact.*, vol. 4, no. CSCW3, Jan. 2021. [Online]. Available: https://doi.org/10.1145/3434168

[12] X. Chen, S. Chen, X. Wang, and Y. Huang, ""i was afraid, but now i enjoy being a streamer!": Understanding the challenges and prospects of using live streaming for online education," *Proc. ACM Hum.-Comput. Interact.*, vol. 4, no. CSCW3, Jan. 2021. [Online]. Available: https://doi.org/10.1145/3432936

[13] A. T. Cianciolo and R. J. Sternberg, *Practical Intelligence and Tacit Knowledge: An Ecological View of Expertise*, 2nd ed., ser. Cambridge Handbooks in Psychology. Cambridge University Press, 2018, p. 770–792.

[14] M. Polanyi, *The Tacit Dimension*. University of Chicago Press, 1966.

[15] D. Wang, Y.-C. Lee, and W.-T. Fu, ""i love the feeling of being on stage, but i become greedy": Exploring the impact of monetary incentives on live streamers' social interactions and streaming content," *Proc. ACM Hum.-Comput. Interact.*, vol. 3, no. CSCW, Nov. 2019. [Online]. Available: https://doi.org/10.1145/3359194

[16] Z. Lu, M. Annett, M. Fan, and D. Wigdor, ""i feel it is my responsibility to stream": Streaming and engaging with intangible cultural heritage through livestreaming," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, ser. CHI '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 1–14. [Online]. Available: https://doi.org/10.1145/3290605.3300459

[17] D. L. Chen, D. Freeman, and R. Balakrishnan, "Integrating multimedia tools to enrich interactions in live streaming for language learning," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, ser. CHI '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 1–14. [Online]. Available: https://doi.org/10.1145/3290605.3300668

[18] P. Lessel, A. Vielhauer, and A. Krüger, "Expanding video game live-streams with enhanced communication channels: A case study," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, ser. CHI '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 1571–1576. [Online]. Available: https://doi.org/10.1145/3025453.3025708

[19] S. Yang, C. Lee, H. V. Shin, and J. Kim, "Snapstream: Snapshot-based interaction in live streaming for visual art," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, ser. CHI '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 1–12. [Online]. Available: https://doi.org/10.1145/3313831.3376390

[20] K. Kobs, A. Zehe, A. Bernstetter, J. Chibane, J. Pfister, J. Tritscher, and A. Hotho, "Emote-controlled: Obtaining implicit viewer feedback through emote-based sentiment analysis on comments of popular twitch.tv channels," *Trans. Soc. Comput.*, vol. 3, no. 2, Apr. 2020. [Online]. Available: https://doi.org/10.1145/3365523

[21] L. Haaranen, "Programming as a performance: Live-streaming and its implications for computer science education," in *Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education*, ser. ITiCSE '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 353–358. [Online]. Available: https://doi.org/10.1145/3059009.3059035

[22] S. T.-H. Sher and N. M. Su, "Speedrunning for charity: How donations gather around a live streamed couch," *Proc. ACM Hum.-Comput. Interact.*, vol. 3, no. CSCW, Nov. 2019. [Online]. Available: https://doi.org/10.1145/3359150

[23] D. Cortes, "How to design and code a website live from scratch! (twitch livestream)," https://www.youtube.com/watch?v=HRzVb9CO_gM, 2017, accessed: 2021-01-15.

[24] D. Israel, "Part 1: Live coding — build a developer portfolio site," https://www.youtube.com/watch?v=5IOrgu3R9sA, 2018, accessed: 2021-01-15.

[25] The Coding Train, "Build a full stack twitter clone with coding garden," https://www.youtube.com/watch?v=JnEH9tYLxLk, 2018, accessed: 2021-01-16.

[26] E. Zaneski, "Building real-time web apps with javascript," https://www.youtube.com/watch?v=IaiLu2vX9Ms, 2016, accessed: 2021-01-16.

[27] C. Coyier, "Screencast 109: Getting off ftp and onto git deployment with beanstalk," https://www.youtube.com/watch?v=cUeMF18zA4Y, 2012, accessed: 2021-02-02.

[28] D. Robinson, "Tidy tuesday screencast: analyzing women's world cup data," https://www.youtube.com/watch?v=ZOQSuapvHqA, 2019, accessed: 2021-01-15.

[29] R. Peng, "Data analysis livestream 2: Moving files around," https://www.youtube.com/watch?v=Rl1X0ngnbhY, 2019, accessed: 2021-01-15.

[30] D. Langer, "Introduction to data science with r - data analysis part 1," https://www.youtube.com/watch?v=32o0DnuRjfg, 2014, accessed: 2021-01-16.

[31] M. Shkreli, "Finance 7th lesson," https://www.youtube.com/watch?v=iDLU7TSqPYI, 2017, accessed: 2021-01-16.

[32] J. Knowles, "Data science livestream - row coding," https://www.youtube.com/watch?v=TkMRoRmbvRc, 2016, accessed: 2021-02-02.

[33] saejinoh, "Pain process - regimented art training and going from zero to pro," https://www.youtube.com/watch?v=wRNj1h5tVG4, 2018, accessed: 2021-01-15.

[34] Color with Kurt, "Procreate 4 coloring tutorial stream recording - coloring line art," https://www.youtube.com/watch?v=6l6PXhaEMk0, 2017, accessed: 2021-01-15.

[35] vonnart, "Stream: Drawing and understanding eyes tutorial," https://www.youtube.com/watch?v=AwuNRkwB49s, 2019, accessed: 2021-01-16.

[36] M. Bucci, "Digital art using 3d and 2d techniques plus q&a," https://www.youtube.com/watch?v=1PRiM1LAh2U, 2018, accessed: 2021-01-16.
[37] T. Edlin, "Workflow for creating sci-fi interior concept art bst:26," https://www.youtube.com/watch?v=ON6TppVv9ZU, 2016, accessed: 2021-02-02.
[38] Mathilification, "Arc totems, the uber elder slayers - gameplay guide," https://www.youtube.com/watch?v=TqGcQUUgJK8, 2018, accessed: 2021-01-15.
[39] swc19, "How to speedrun anything episode 1: Celeste (any%)," https://www.youtube.com/watch?v=ruVZGxnBMpM, 2019, accessed: 2021-01-15.
[40] Mrhappy1227, "Ffxiv: Do's and dont's of macros," https://www.youtube.com/watch?v=qshNKeDENlk, 2019, accessed: 2021-01-16.
[41] FFXIVMomo, "Ffxiv shadowbringers: How to scholar in dungeons," https://www.youtube.com/watch?v=wE1Ay8mmVB8, 2019, accessed: 2021-01-16.
[42] ZackScottGames, "Yoshi's crafted world - gameplay walkthrough part 1 - sunshine station! yarrctopus docks!" https://www.youtube.com/watch?v=VjSyAR1pAG8, 2019, accessed: 2021-02-02.
[43] J. R. Macnamara, "Media content analysis: Its uses, benefits and best practice methodology," *Asia Pacific Public Relations Journal*, vol. 6, no. 1, p. 1, 2005.
[44] J. M. Corbin and A. L. Strauss, *Basics of qualitative research: techniques and procedures for developing grounded theory*. Thousand Oaks, Calif: SAGE Publications, Inc., 2008.
[45] A. Collins, J. S. Brown, and S. E. Newman, "Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics," *Knowing, learning, and instruction: Essays in honor of Robert Glaser*, pp. 453–494, 1989.
[46] J. A. Horvath, J. Hedlund, S. Snook, G. B. Forsythe, and R. J. Sternberg, *Tacit Knowledge in Military Leadership: Some Research Products and Their Applications to Leadership Development*. US Army Research Institute for the Behavioral and Social Sciences, 1998, vol. 1081. [Online]. Available: https://www.doi.org/10.21236/ada344857