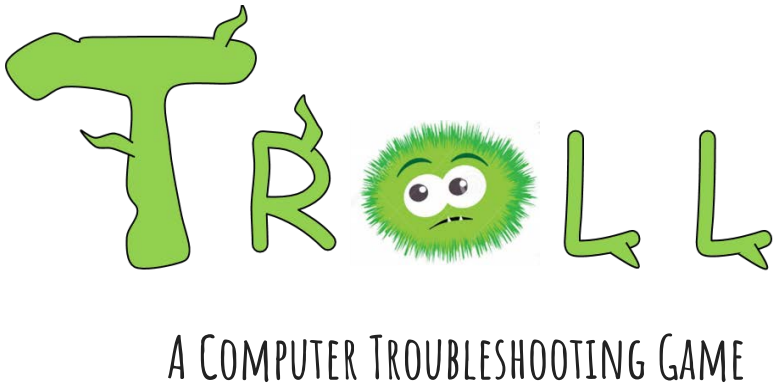
Troll: A Game-Based Program for Enhanced Computer Troubleshooting in the Classroom [Policy Brief]



Introduction to the Problem of Practice: How is technology currently affecting the K-12 classroom?

Students of the future need to know how to interface with their computer to solve common technical issues. As states continue to adopt and develop technology curriculum standards, as well as bring more technology into everyday classroom use, student interaction with technology is becoming a forefront issue in educational progress. With this comes the rise of technical challenges faced by teachers and students alike. In a live, high-stakes classroom environment, technical issues could slow down the pace of a lesson and provide an unnecessary barrier to technology integration. Often, schools' IT departments cannot solve every possible issue that arises, requiring a need for such troubleshooting skills to be in the hands of students and teachers. According to a 2016 national education survey, 19% of surveyed teachers reported that "software glitches" and 21% that "insufficient support from IT staff" were "significant" barriers to technology use in the classroom (Cortez, 2016). Here, we introduce a

computer-based simulation to allow students to practice common troubleshooting skills that could accelerate the process of learning and helping their peers with such issues, as well as reduce one barrier to technology in the classroom. Our game, Troll, simulates common issues that may arise on a students' computer, and with Troll, we hope to build the confidence of students to troubleshoot and thus smoothen the process of technology integration.

The Intended User and Context: How can students benefit from troubleshooting skills?

We have chosen to focus on students in high school first. One, because they should have more knowledge of the how a computer system works and have had some encounters with the computer terminal, and two, a desire and ability to learn more about how to interface with their computer's terminal. According to the Massachusetts states' technology standard handbook, troubleshooting is listed under one of their four guiding standards: Computing Systems (CS). In line with CS, students should be able to "use troubleshooting strategies to solve routine hardware and software problems," a standard we hope this game addresses, specifically the software component. An overarching goal of the standards is that "digital literacy and computer science ideas should be explored in ways that stimulate curiosity, create enjoyment, and develop depth of understanding" (Massachusetts Department, 2016). Through the implementation of a game that makes something that causes undue frustration more manageable, we hope to allow students to explore how to interact with their computers in a fun, meaningful, and useful way. We specifically intend to target students wanting to learn more about their computer through acquiring basic knowledge of the command terminal. We want to motivate students to independently troubleshoot and thus help their peers.

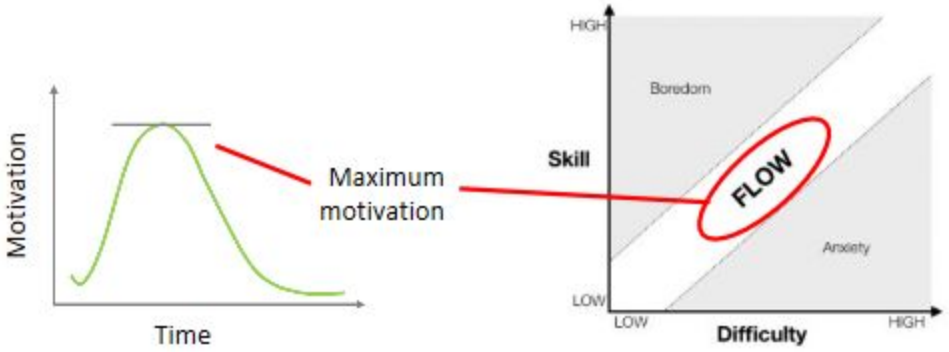
This leads to our learning objective:

We believe students of the future should be able to employ creative problem-solving skills to learn how to interface with their computer and gain the confidence to quickly troubleshoot common computer software issues.

We include the goal of confidence because we intend this game to instill in students the confidence to independently solve computer issues and then use their knowledge to teach other students without unnecessarily straining a teacher in charge of managing a full classroom.

Existing Solutions and Design Aspirations: What has been done about technological literacy?

Currently, there exist some solutions for helping computer-users troubleshoot. Many online tutorials exist for how to use the command line. Some tutorials are even interactive (for example, Learn Shell). Additionally, there are various tools online for the purpose of interrupting users to motivate them to stay on task (for example, parental controls, break reminders, and online task managers). We wanted to create a user experience that both felt interactive and was highly motivating. In order to learn command prompts, users often have to sift through long, jargony manuals (for example, the BASH manual), affording them little opportunities to put their reading skills to practical use. This is why we chose a game that simulates common computer issues, forcing a user to implement problem-solving skills on the spot to eliminate the issue. A popular concept in video game psychology is the principle of “flow.” Within a state of flow, a user is at their maximum motivation level and is intrinsically motivated to solve a problem. If either their ability or the difficulty of challenge are mismatched, the user is not at their maximum performance (Csikszentmihalyi, 1990) (See Figure below). Most often, video games achieve a state of flow because the level of the game is adjusted as the user progresses. Although Troll currently does not support multiple levels, that is something we hope to implement in future iterations.

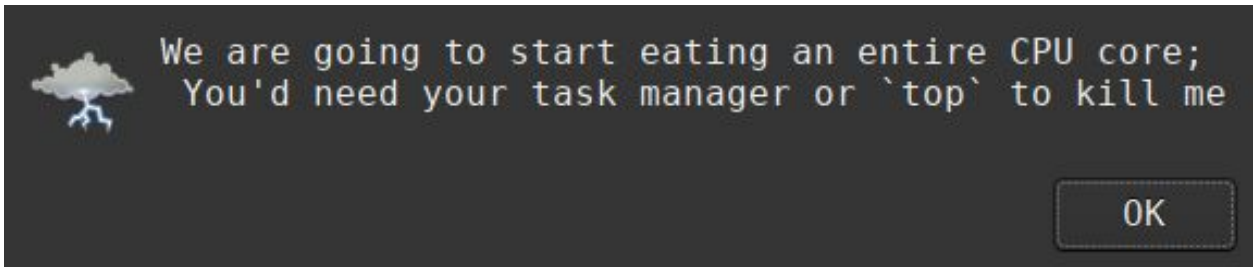


Innovative Solution and Technology Justification: How might we increase students' troubleshooting skills?

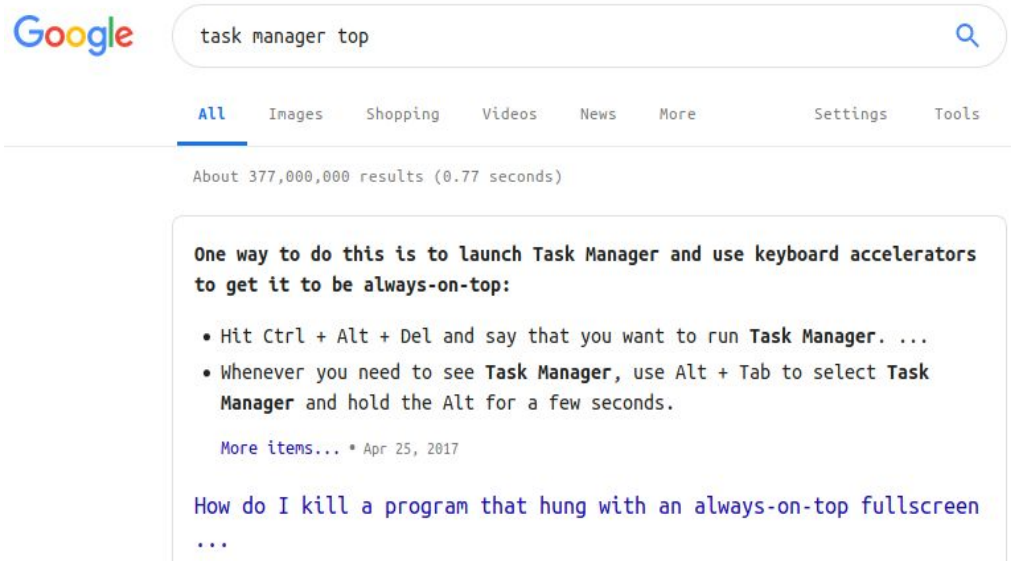
We introduce Troll: A Computer Troubleshooting Game, in which we hope to appropriately engage and challenge users to acquire troubleshooting skills that they can confidently apply when they face them in real life. Troll currently has three different challenges:

(1) One of the user's CPU cores is being taken up, significantly slowing down the computer. They must find the program that is doing this in the background and end it.
(2) A program is leaking memory in the RAM. Everything will be fine for a while, but the computer will significantly slow down after a while, and in some cases, completely freeze. They must find the program that is doing this in the background and end it.
(3) The user has forgotten where a certain important file is on the computer and has to locate it.

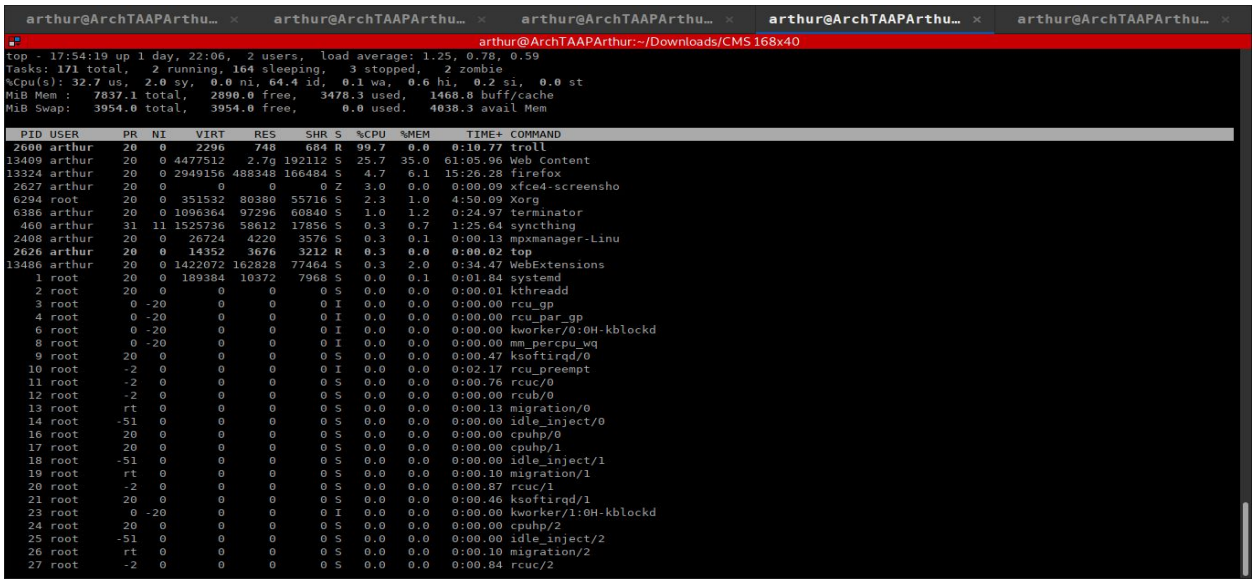
Here, we walk through the workflow for sample challenge (1):



Prompt to tell the user that something undesired is happening and gives a hint on how to fix the issue.



Searching for the command `top` as suggested in the hint.



The command `top` running. The programs are sorted by CPU-usage and 'troll' is the first in the list.

Success message when the user has successfully completed the task by ending the 'troll' program.

Playtest and Interviews: How did we apply user-centered design insights to Troll?

To gather insights from users, we interviewed two college students and one current 6th-grade teacher. Each of them had some basic understanding of troubleshooting, although powering off/on, Ctrl+C, or Googling were their most common techniques. However, using a search engine like Google to look up common commands is something we incorporated into our game. Based on user feedback, we included a hint for suggestions on what command to use or what to "google," since even that can be an overwhelming task. Our teacher interviewee, Ms. Johanna Spears, provided some additional insight into how technology impacts the classroom. Her school funds iPads for each student to use, leading to a host of technical issues, such as Wi-Fi connectivity and being locked out of applications. Although our game does not yet teach troubleshooting these issues, they are something to keep in mind if expanding the game's repertoire. Spears commented on the need for teachers to be able to quickly help students, saying that it "takes a lot of time and exploration" to solve computer issues. We decided to take this need and target the student users. Putting troubleshooting skills in the hands of students could help classroom management, since one teacher cannot individually assist a class of about 20-30 students.

Additionally, we performed two in-class playests with our peers in CMS.594. Both had some basic troubleshooting knowledge, but emphasized the use of hints as appropriate guidance for those learning these skills. They recommended that users be empowered to choose the

frequency of encountering the simulated issues (rather than random “trolls,” which was the initial inspiration for our game and logo). Users also commented on having a reward-based system in which one can achieve points or a mark of success. We currently indicate a user successfully completed a challenge, hopefully making troubleshooting a more fun experience. Users can also repeat a challenge until they feel they have learned that skill.

Future Work: What can we iterate upon moving forward?

There are many ideas we would want to add to a final version. First, we would want to add more features. This would include adding more challenges, difficulty levels, additional hints, and a point system to track progress. Next, we want to make the game cross-platform. Currently the User Interface (UI) only works on Linux/Mac, but it could be easily ported to the Windows Operating System (OS). Similarly, we want to make OS-specific challenges, such as dealing with the forced reboot of Windows. Lastly, we want to make the game a background process so it could spawn programs randomly or at a set time instead of needing to manually start it; users would then be able to choose the frequency of trolls (if choosing the random option) or set a specific time to undergo the simulation. With a revised prototype, we would plan to do further user testing with actual high school students. We are also considering expanding the intended user to teachers and population groups new to computing in order to increase computer literacy.

Acknowledgments

We would like to thank the Edtech teaching staff of CMS.594 for guiding and supporting us throughout the semester and providing a wonderful introduction to the field of educational technology!

Thank you to our classmates for providing good class discussions and playtesting our prototype. Thank you to all our interviewers as well for providing thoughtful insights. We appreciate everyone’s support during this project!

References

Cortez, M. B. (2016). Teachers Are Confident About Using Technology, Now More Than Ever. Retrieved from https://edtechmagazine.com/k12/article/2016/07/teachers-are-confident-about-using-technology-now-more-ever.

Csikszentmihalyi, M. (1990). *Flow: The Psychology of Optimal Experience*. New York: Harper and Row.

GNU Bash manual. (2019). *GNU Project - Free Software Foundation.* Retrieved from https://www.gnu.org/software/bash/manual/.

Kerrisk, M. (2019). User Commands. *Linux man-pages*. Retrieved from http://man7.org/linux/man-pages/man1/top.1.html.

Learn Shell. (n.d.). Retrieved from https://www.learnshell.org/.

Massachusetts Department of Elementary and Secondary Education. (2016). *DIGITAL LITERACY AND COMPUTER SCIENCE CURRICULUM FRAMEWORK*. Retrieved from http://www.doe.mass.edu/frameworks/dlcs.pdf.

Project Hamster. (2017). Retrieved from http://projecthamster.org/

Interviews:

Chang, H., personal communication. April 30, 2019.

Spears, J., personal communication. May 9, 2019.

Reddy, S, personal communication. April 30, 2019.

Playtests:

Graham, E., personal communication. May 8, 2019.

Vasudevan, S., personal communication. May 8, 2019.

CMS.594/CMS.894 Education Technology Studio
Spring 2019