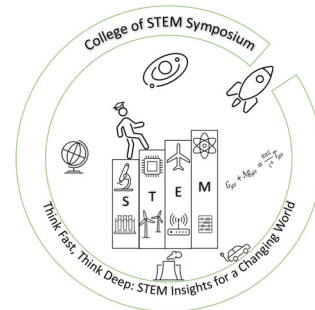




Using AI to develop simulations for the biology classroom

Stephen Burnett^{1*}

¹ College of STEM, School of Sciences, Clayton State University, Morrow, GA 30260 USA

Abstract. Many biological phenomena cannot be observed in the classroom for a variety of reasons (limited time, lack of appropriate facilities, etc.). When these phenomena are needed to understand the course concepts, faculty need methods to be able to explain them successfully. Computer simulations have been used in the classroom for decades, but finding ones that are suited to the instructor's need without incurring significant costs is often a major limitation. I will discuss the use of AI tools to develop simulations quickly. These tools allow the instructor to specify the parameters precisely, producing a simulation that is more useful with limited investment of time and money. I will also address pitfalls and issues with using these tools and ways that science faculty can get the greatest benefit from them.

1 Introduction

Studies show that active learning techniques can improve student success in STEM disciplines [1,2]. Unfortunately, faculty are often overworked [3,4] or lack appropriate background knowledge [5-7] and the time to develop active learning techniques can be prohibitive, leading to the use of strategies that are less effective [4]. In addition, concepts in biology are often difficult to address in a classroom setting due to practical constraints (e.g., insufficient time, equipment, etc.). As a result, faculty would benefit from techniques that overcome these issues. One that has shown success in the STEM classroom has been the use of computer simulations. Such simulations have been shown to improve learning and student success [8-19] although with some caveats [e.g., 11, 20-24]. This paper addresses the use of LLM's to develop simulations that can be deployed in the classroom or laboratory setting, allowing faculty with minimal training to produce useful software with a limited investment of time.

2 Choosing a simulation

There are a few issues that need to be considered when trying to use simulations in a classroom. These include the ability to access the simulations, the need to find simulations that address the instructional need [e.g., 25], and the potential cost. All of these issues present difficulties both individually and in combination.

Accessing simulations will require access to the required computer hardware and software for the simulation(s) being used. Many of the available simulations discussed below are web-based, which lowers the bar for accessing them, but some require installation on a computer, which requires all students have the appropriate hardware to run the simulation.

There are a variety of commercially available simulation platforms (e.g., Labster [www.labster.com/] and SimBio [simbio.com/]). With these systems, there is no guarantee that the particular system provides simulations that match the need and the software comes with a cost [26], which must be borne by either the students or the institution. Additionally, there is often no straightforward way to assess costs and their impacts [e.g., 27]. To lower costs, there are free simulations available (e.g., Biology Simulations [www.biologysimulations.com/simulations]; PhET [phet.colorado.edu/], and Virtual Biology Lab [virtualbiologylab.org/]). These have the same limitations as commercial simulations in terms of what simulations are available,

As a result, there is a strong argument for the ability of faculty to develop specific simulations to meet their instructional goals without adding prohibitive costs. However, this puts an additional burden on the faculty members, which will often prohibit this. Another roadblock exists because many biology faculty lack the necessary technical skills to develop new

* Corresponding author: sburnett@clayton.edu

simulations [28]. This is an area where LLMs could be leveraged to produce simulations with minimal time and effort even by faculty who may lack the technical skills to produce their own.

Before starting it's best to ask a number of questions so that you can use the most suitable simulation, including the LLM that will be used to develop the simulation, what the simulation needs to do, and how your students will access the simulation.

2.1 Which LLM will you use?

There are multiple LLM tools available, both general and specialized. The examples addressed here were generated using ChatGPT from OpenAI or Gemini from Google. Several other general LLMs exist, and new models are released frequently, so the best tool can change over time. It's best to gain some familiarity with best practices for a specific LLM to get the best results.

2.2 What does the simulation need to do?

While this is an obvious question to ask, the issue deserves careful thought. The more precise the information you provide in your prompt to the LLM, the more likely you are to get a result that meets your needs. If there are details missing from the prompt, the LLM will try to fill in those details, often producing results that do not meet your expectations.

2.3 How will you “deliver” the simulation?

When you develop a simulation for the classroom, you need to consider who will run it and what hardware they will have access to. If students are expected to run the simulations themselves, it's best to generate ones that will operate on the widest possible types of hardware. Simulations written in HTML can run on virtually any computer hardware currently available and may even be able to run on mobile devices. These will usually require minimal technical skills for students to use. Simulations based on general programming languages like Python can run on most computers, but may require students to install additional software, which can present difficulties. Simulations written in more technical software (Matlab, e.g.) often require higher-end hardware and limits on which operating systems can run them. It's also best to take advantage of any technical skills you already possess. If you have some familiarity with a particular programming language, it can help you produce useful simulations more quickly than one with which you are unfamiliar.

3 Type of simulation

Once you have decided how to deliver it, you need to assess your starting point. This will depend on what resources you have available to you.

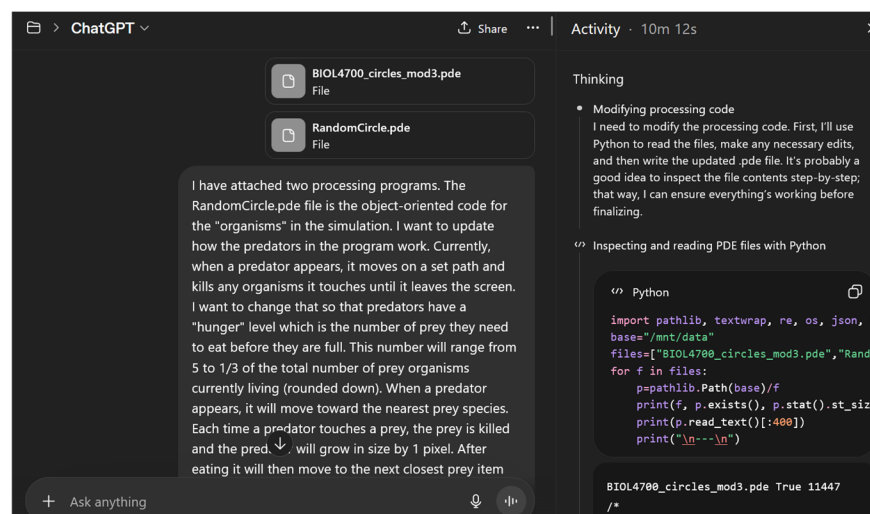


Fig 1: Example of using ChatGPT to import and update software written by the author. Note that it took about 10 minutes to import the program to where it could make changes I requested.

3.1 Modifying existing software

If you have software that already exists (written by you or someone else) then you can have the LLM import the software and make updates/improvements to it (Figure 1). This process can take some time for the LLM to process the code and to develop and add new capabilities.

LLMs can also be used to detect and correct bugs that might exist in software [29], if you provide suitable descriptions of the problems (Figure 2). As discussed later, this does sometimes present issues, but it can work quite well.

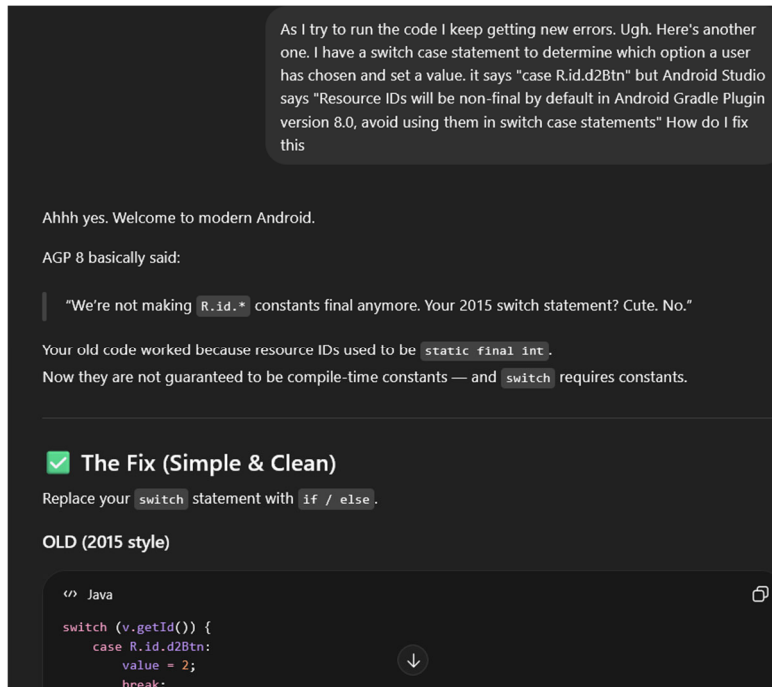


Fig. 2: Using ChatGPT to identify a bug in a program caused by changes in the programming language

3.2 Generating new software

Often, you will not have access to software that meets your requirements. If you describe your simulation, it is possible for an LLM to provide fully written code quickly. In this case, it's important to provide the most detailed prompt that you can. This will guide the LLM and minimize the risk that it is filling in details that don't match your needs. One of the major benefits is that the LLM will often produce the code you request in a matter of seconds (Figures 3 and 4).

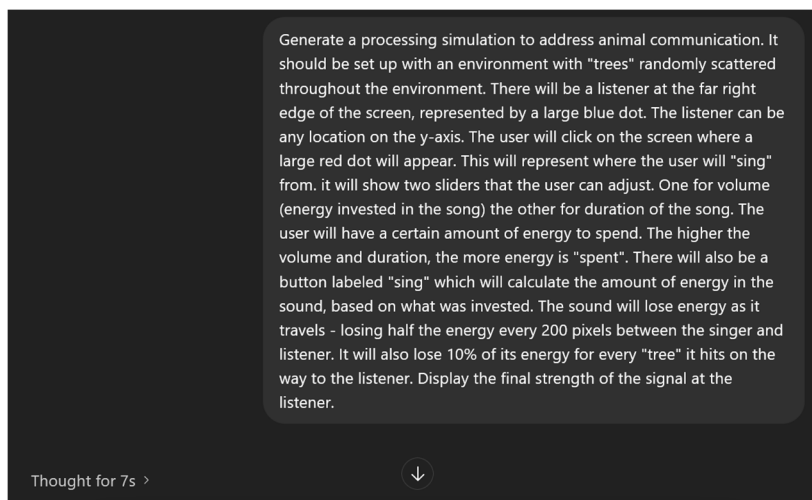


Fig. 3: Detailed caption for a new simulation that was produced in about 7 seconds.

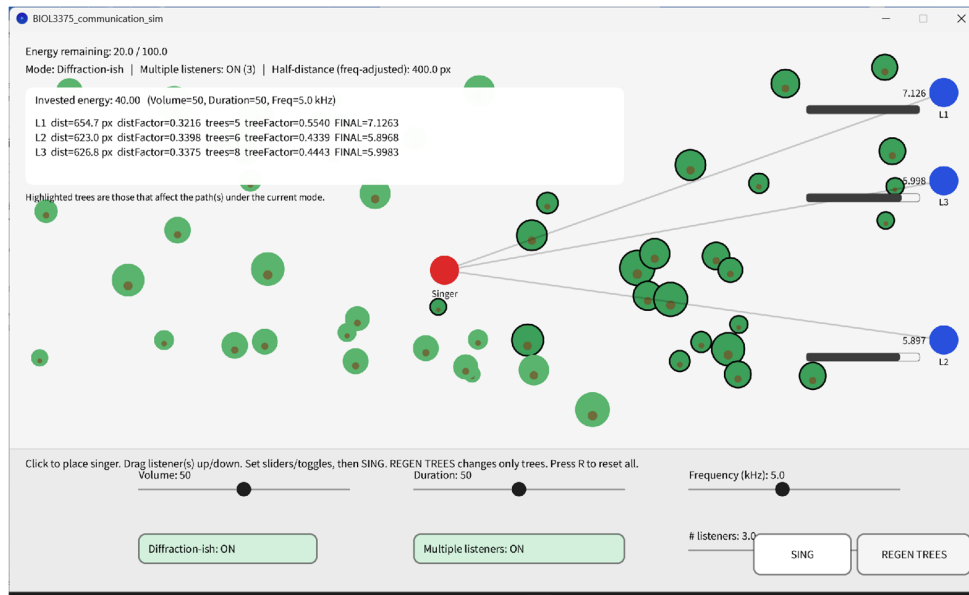


Fig 2: Simulation produced from the prompt in the previous figure. While it required further prompting to produce a finalized version, this one had almost everything I needed to use it in the classroom.

If you provide a similar, but less-detailed prompt (Figure 5) you may get a workable simulation, but it might be very different from what you were trying to get (Figure 6).

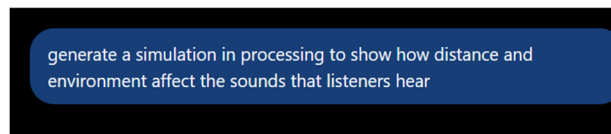


Fig 5: Shorter prompt to produce a similar simulation

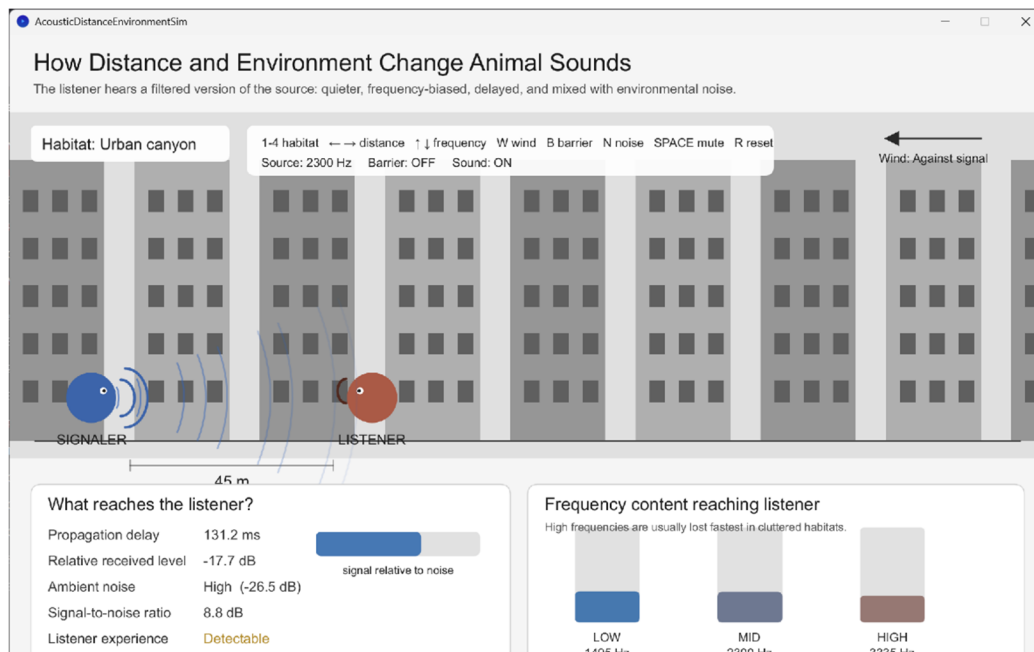


Fig 6: Simulation produced by the shorter prompt. While the simulation works, it is very different from the parameters the detailed prompt provided.

4 Issues with AI coding

A common misconception with LLMs is the perception that they understand what they are being asked to do [30-32]. This is encouraged by the output the LLMs provide [31] as well as how they are presented in popular media [33]. This can be most obvious when you don't provide details and the LLM produces a simulation that doesn't correctly match your prompt (Figure 6).

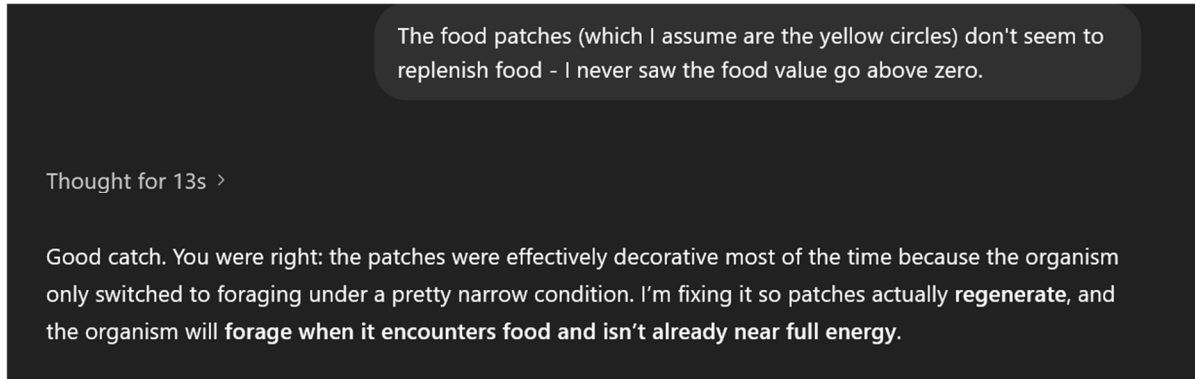


Fig. 3: ChatGPT produced code that didn't do what was requested - variables were not being updated. "Good catch" is a common response when such errors are pointed out to ChatGPT.

There can also be problems where the LLM is trained on code from older versions of the software or when it has been trained on inaccurate data. When generating simulations in Processing, the code would often contain a syntax error because ChatGPT was using syntax for earlier versions of the language than the version I was using (Figure 7).

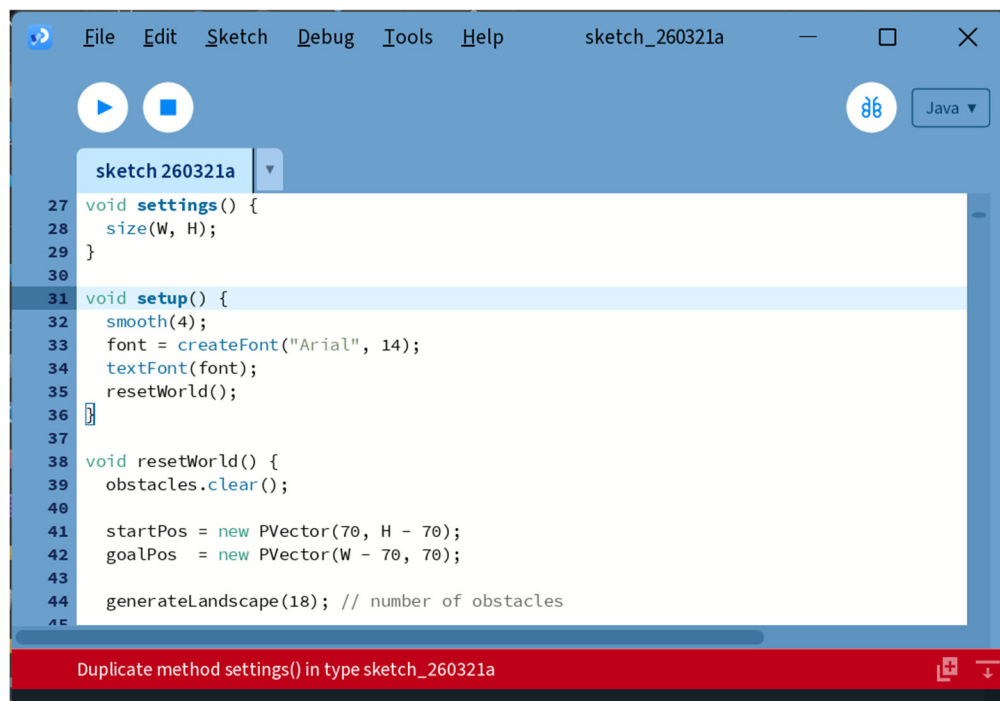


Fig. 7: Syntax error because newer versions of Processing do not allow the settings function.

There will also be times when the LLM is confidently incorrect. When it produces code based on your prompt, but it doesn't work at all (Figure 8).

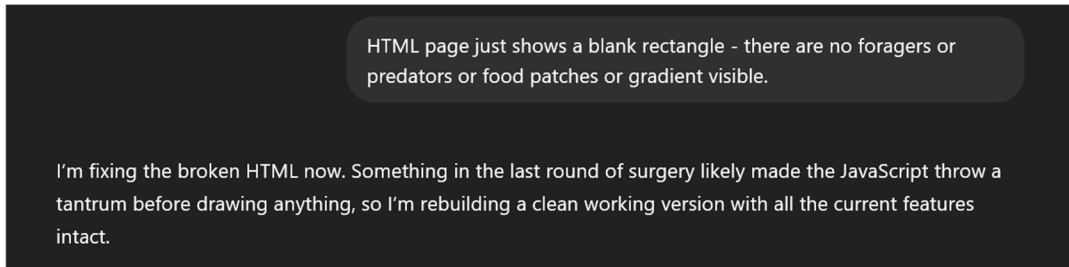


Fig. 4: ChatGPT produced a simulation with a bug that prevented it from running

While you can ask the LLM to identify and suggest fixes for code (either yours or some produced by the LLM) it may not provide feedback in a way that is useful (Figure 9). Fortunately, it is often possible to prompt the LLM for a more user-friendly result (e.g., asking for it change the code and provide you with a full copy).

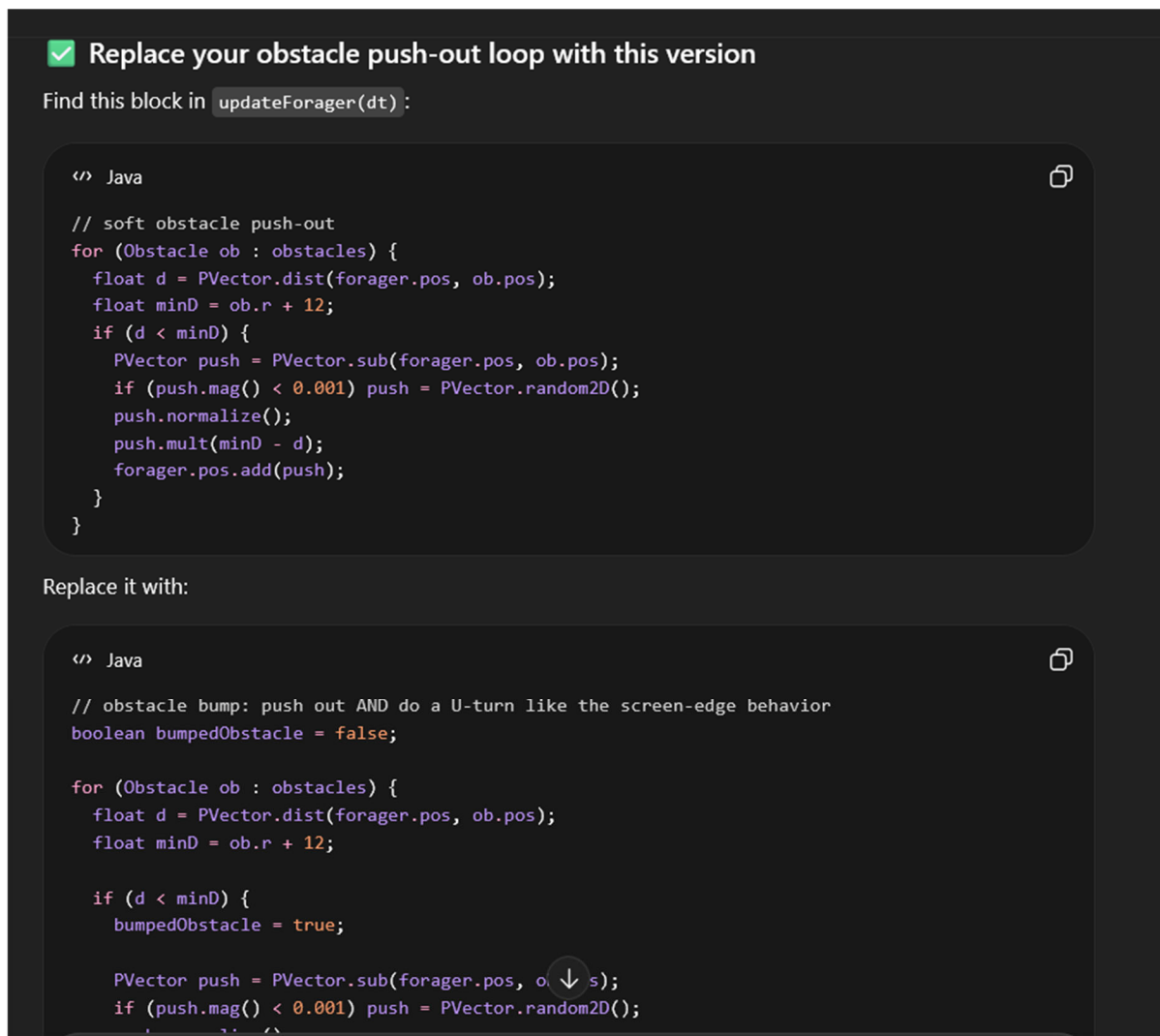


Fig. 9: Suggestions from ChatGPT that require the user to find and replace text in a program that contains over 1000 lines of code.

5 Addressing concerns with AI use

There are many issues raised with the use of AI. In terms of the techniques discussed here, potential issues include copyright infringement [34-36], AI's impact on the environment [37-38], deskilling [39-42], hallucinations [43-44], and software security [45-48].

5.1 Copyright

Regarding copyright, the AI models are trained on a lot of publicly available information such as might be posted in online forums like StackOverflow and Reddit [49-51]. This seems to reduce potential copyright issues since it's public information already. However, there is often strong pushback from people who submit answers to those sites [49,51].

5.2 Environmental impacts

Resource use is a definite concern and there are limited ways to mitigate this with the models that are currently available, although environmentally friendly energy sources could be used [37-38].

5.3 Deskilling

Deskilling is difficult to address as there is no clear data on how many biologists know how to code, especially the types of simulations as described here. Opinions tend to vary, although it's common to see claims that some form of programming skill is likely to be helpful (e.g., this thread on Reddit[†]). Most of the discussion in that thread relates to data analysis, which is a distinct process from developing new simulations, which suggests coding simulations may be considered less important (and therefore less common among faculty). Even so, building simulations with LLMs could still have a negative impact in the sense that it would prevent biology faculty from developing new skills or weaken current skills [42]. Given faculty workloads, it's likely that many faculty don't have or were not using these skills at high rates, however.

There are also concerns about using simulations in classes as they may not transfer to real-world situations [52] or give an illusion of learning without providing actual improvements in understanding [53]. It seems that this can be mitigated by carefully examining the type of simulation, how they are used, and testing for improvement (or lack thereof) in student understanding, success, etc.

5.4 Hallucinations

Hallucinations are an unfortunate side effect of the way LLMs work [31, 43-44], which necessitates careful checking of their output to ensure that the results are reasonable. In some cases, this might be done qualitatively (e.g., do the organisms in a simulation show behavior that fits what might be seen in nature) or quantitatively (e.g., does the simulation produce results that match experimental or theoretical evidence).

5.5 Software security

Security of the software is also a potential concern, which would be most relevant to software that is going to be distributed to others. This applies even to HTML simulations [54] and such security problems occur frequently with all major LLMs [47]. Careful examination of the code would be necessary to fully address security [55], which would be easier if the instructor already had some familiarity with the programming language. Some automated techniques show promise at code analysis [56] but these involve technical skills and resources that are likely not widely available. If the instructor is developing simulations that will only run as demonstrations in class, it is possible to minimize the use of libraries that contain malicious code (again, requiring careful analysis of the code). However, if the simulation is distributed, or if it will be used in research (as opposed to teaching), then much more care would be warranted to avoid the spread of malicious code or compromising research data [31].

6 Conclusions

The final decision to use simulations and/or AI in any context needs to be considered carefully, based on many different factors. Based on these initial results, the benefits of these tools seem to justify their use for developing simulations for the classroom, as they would provide a much larger set of capabilities to all faculty, even those with more limited technical expertise.

[†] https://www.reddit.com/r/biology/comments/1hbpg7b/how_necessary_is_it_for_a_biologist_to_know_coding/

References

1. S. Freeman, S.L. Eddy, M. McDonough, M.K., Smith, N. Okoroafor, H. Jordt, and M.P. Wenderoth. *PNAS*. **111**, 8410-8415. (2014)
2. E.J. Theobald, M.J. Hill, E. Tran, S. Agrawal, E.N. Arroyo, S. Behling, N. Chambwe, D.L. Cintrón, J.D. Cooper, et al. *PNAS*. **117**, 6476-6483. (2020)
3. H. Cadena-Povea, M. Hernández-Martínez, G. Bastidas-Amador, and H. Torres-Andrade. *Int. J. Environ. Res. Public Health*. **22**, 1214. (2025)
4. G.D. Pillar. *The ACAD Leader*. **21 Nov**, (2025)
5. J. Drew, W. Morgan, S. Galindo, A.J. Kleinschmit, M. McWilliams, M. Pauley, E.W. Triplett, J. Williams, B. Murdoch, and A. Rosenwald,. *Front. Educ.* **8**, 1317-181. (2023)
6. C.D. Perumal, A. Nofal, B.J. Kolber, and R.M. Neilan. *Spora*. **11**, 64-75. (2025)
7. J.J. Williams, et al. *PLOS One*. **14**, e02242-88. (2019)
8. M.T. Bonde, G. Makransky, J. Wandall, M.V. Larsen, M. Morsing, H. Jarmer, and M.O. Sommer,. *Nat. Biotechnol.* **32**, 694-697. (2014)
9. O. Chernikova, N. Heitzmann, M. Stadler, D. Holzberger, T. Seidel, and F. Fischer. *Rev. Educ. Res.* **90**, 499-541. (2020)
10. T. de Jong, M.C., Linn, and Z.C. Zacharia. *Science*. **340**, 305-308. (2013)
11. L.E. de Vries, and M. May. *Biochem. Mol. Biol. Educ.* **47**, 257-262. (2019)
12. N. Dyrberg, A.H. Treusch, and C. Wiegand. *J. Biol. Educ.* **51**, 358-374. (2016)
13. N.J. Gibbons, C. Evans, I.A. Payne, K. Shah, and D.K. Griffin. *Cell Biol. Educ.* **3**, 263-269. (2004)
14. G. Makransky, M.W. Thisgaard, and H. Gadegaard. *PLOS One*. **11**, e0155-895. (2016)
15. V.A. Miller, T. Marks, and D.K. Thompson. *J. Microbiol. Biol. Educ.* **26**, 1-13. (2025)
16. M. Miyamoto, D.M. Milkowski, C.D. Young, and L.A. Lebowicz. *J. Biocommun.* **43**, 23-31. (2019)
17. M. Thisgaard, and G. Makransky. *Front. Psychol.* **8**, 1-13. (2017)
18. M.M. Waldrop. *Nature*. **499**, 268-270. (2013)
19. J. Wang, Y. Jiang, J. Yang, Q. Zhou, H. Hu, Y. Chen, and J. Jiang. *BMC Med. Educ.* **25**, 1-14. (2025)
20. E. Bauermeister, E. Janssen, and J.-O. Engler. *Educ. Inf. Technol.* **30**, 10979-10995. (2025)
21. O. Chernikova, D. Holzberger, N. Heitzmann, M. Stadler, T. Seidel, and F. Fischer,. *Z. für Pädagog. Psychol.* **38**, 15-25. (2024)
22. J.S. Carroll, H. Najafi, and M. Steiner. *Biochem. Mol. Biol. Educ.* **53**, 422-432. (2025)
23. C.S. Griffin, S. Loughran, B. Kelly, E. Healy, G. Lambe, A. van Rossum, B. Murphy, E. Moore, C. Burke, A. Morrin, C. Breslin, F. Heaney, et al. *Adv. Physiol. Educ.* **49**, 314-330. (2025)
24. G. Makransky, T.S. Terkildsen, and R.E. Mayer. *Learn. Instr.* **60**, 225-236. (2019)
25. M. Jonsson, S. Ghirlanda, J. Lind, V. Vinken, and M. Enquist. *J. Open Source Softw.* **6**, 2891. (2021)
26. Z. Vecsy. How to Implement an Online Simulation into My Course? Accessed on: 17 Aug 2026. <https://www.fligby.com/how-to-implement-an-online-simulation-into-my-course/>. (nd)
27. G. Morgan. How Much Does It Cost to Build an Online Course? Accessed on: 17 Aug 2026. <https://onedtech.philhillaa.com/p/how-much-does-it-cost-to-build-an-online-course>. (2025)
28. N.C. Emery, E. Crispo, S.R. Supp, K.J. Farrell, A.J. Kerkhoff, E.K. Bledsoe, K.L. O'Donnell, A.C., McCall, and M.E. Aiello-Lammens. *BioScience*. **71**, 1274-1287. (2021)
29. T. Weber, M. Brandmaier, A. Schmidt, and S. Mayer. *Proc. ACM Hum. Comput. Interact.* **8**, 1-29. (2024)
30. M. Mitchell and D.C. Krakauer. *PNAS*. **120**, e22159-07120. (2023)
31. B. Mittlestadt, S. Wachter, and C. Russell. *Nat. Hum. Behav.* **7**, 1830-1832. (2023)
32. M. Steyvers, T. Heliodoro, A. Kumar, C. Belem, S. Karny, X. Hu, L.W. Mayer, and P. Smyth,. *Nat. Mach. Intell.* **7**, 221-231. (2025)
33. Z. Lin. *PNAS Nexus*. **5**, 236. (2026)
34. N. Bondari. AI, Copyright, and the Law: The Ongoing Battle Over Intellectual Property Rights. Accessed on: 21 Aug 2026. <https://sites.usc.edu/iprls/2025/02/04/ai-copyright-and-the-law-the-ongoing-battle-over-intellectual-property-rights/>. (2025)
35. D.S. Chawla. As Elsevier sues Meta over AI, other scholarly publishers are keeping a watchful eye. Accessed on: 21 Aug 2026. <https://cen.acs.org/policy/publishing/elsevier-sues-meta-over-ai/104/web/2026/05>. (2026)
36. C.T. Zirpoli. Generative Artificial Intelligence and Copyright Law. Accessed on: 21 Aug 2026. https://www.congress.gov/crs-product/LSB10922?__cf_chl_tk=OOe5OIkciZ87yoW.U0ZMq_KBvT1yFLKrlm1brTxkfGg-1787339444-1.0.1.1. (2025)
37. J. Morrison, C. Na, J. Fernandez, T. Dettmers, E. Strubell, and J. Dodge. *arXiv:2503.05804*. (2025)
38. A. Singh, N.P. Patel, A. Ehtesham, S. Kuma, and T.T. Khoei. *arXiv:2412.04782v2*. (2025)
39. Z. Abrams. *Monit. Psychol.* **57**, 48. (2026)

40. A. Choudhury, and Z. Chaudhry. *J. Med. Internet Res.* **26**, e56764. (2024)
41. A. Ferdman. *AI Soc.* **41**, 3001-3013. (2026)
42. D. Stromberg, V. Lei, and Y. Wu. *SSRN*. (2026)
43. M. Chelli, J. Descamps, V. Lavoué, C. Trojani, M. Azar, M. Deckert, J. Raynier, G. Clowez, P. Boileau, and C. Ruetsch-Chelli. *J. Med. Internet Res.* **26**, e53164. (2024)
44. Z. Zhao, Y. Wang, T. Stuart, M. De Vaan, P. Ginsparg, and Y. Yin. *arXiv:2605.07723*. (2026)
45. J. Deng, Z. Fan, and R. Meng. *arXiv:2606.23130*. (2026)
46. C. Hale. Nearly half of all code generated by AI found to contain security flaws - even big LLMs affected. Accessed on: 8 Jan 2026. <https://www.techradar.com/pro/nearly-half-of-all-code-generated-by-ai-found-to-contain-security-flaws-even-big-llms-affe>. (2025)
47. J. Ji, J. Jun, M. Wu, and R. Gelles. *Cybersecurity Risks of AI-Generated Code*. (2024)
48. S. Kaminsky. The hidden dangers of AI coding. Accessed on: 21 Aug 2026. <https://www.kaspersky.com/blog/vibe-coding-2025-risks/54584/>. (2025)
49. S. Gee. Stack Overflow Strives To Survive In Era Of AI. Accessed on: 21 Aug 2026. <https://www.i-programmer.info/news/99-professional/18102-stack-overflow-strives-to-survive-in-era-of-ai.html>. (2025)
50. M. O'Brien. AI is learning from what you said on Reddit, Stack Overflow or Facebook. Are you OK with that? Accessed on: 21 Aug 2026. <https://apnews.com/article/genai-training-data-stack-overflow-reddit-9d71b75ec16c78c0d2c51cc46121c1a4>. (2024)
51. N. Patel. Stack Overflow users don't trust AI. They're using it anyway. Accessed on: 21 Aug 2026. <https://www.theverge.com/podcast/844073/stack-overflow-ceo-ai-coding-chatgpt-code-red-interview>. (2025)
52. C. Elendu, D.C. Amaechi, A.U. Okatta, E.C. Amaechi, T.C. Elendu, C.P. Ezech, and I.D. Elendu. *Medicine*. **103**, e38813. (2024)
53. A.N. Shetti. *J. Educ. Technol. Health Sci.* **11**, 36-41. (2024)
54. B. O'Neill. Why are HTML Files Dangerous? Accessed on: 21 Aug 2026. <https://cloudmersive.com/article/Why-are-HTML-Files-Dangerous>. (2023)
55. J. Osher. The Risks of Code Assistant LLMs: Harmful Content, Misuse and Deception. Accessed on: 21 Aug 2026. <https://unit42.paloaltonetworks.com/code-assistant-llms/>. (2025)
56. H.A. Al-Hashimi. *Sci. Rep.* **16**, 4239. (2026)