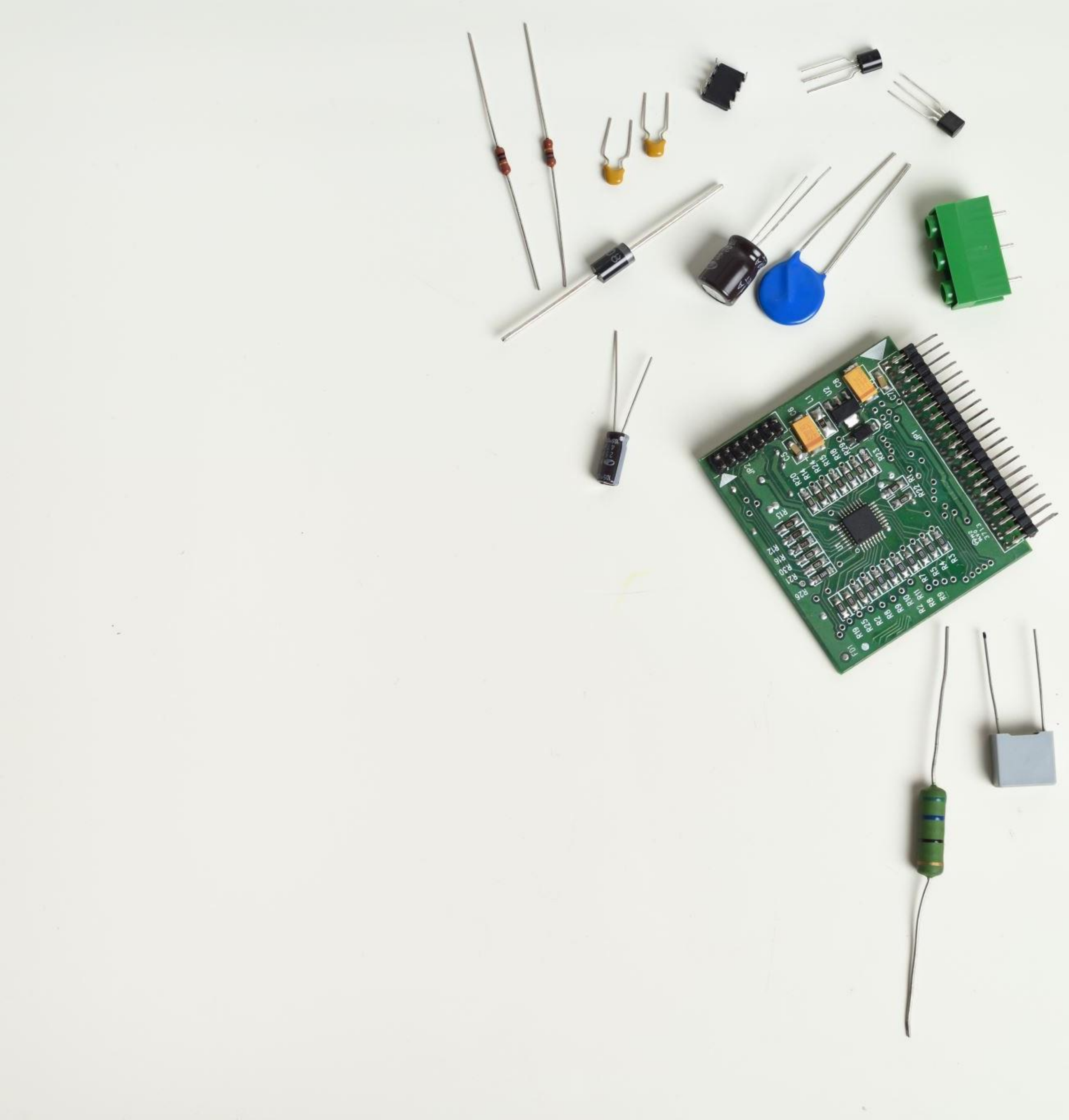
Enhancing Programming Learning with AI Generated Contextual Examples in Digital Creativity

Michael James Scott, Sokol Murturi, Joseph Walton-Rivers

Authors

- **Dr Michael James Scott**
  Head of Computing, Faculty of Screen, Technology, and Performance
  Associate Professor of Computer Science Education
- **Sokol Murturi**
  Lecturer in Computer Science
  Doctoral Candidate for Research into Mixed-Initiative Artificial Intelligence
- **Dr Joseph Walton-Rivers**
  Undergraduate Courses Leader for Computing, Games Academy



Introduction

- Stage 1 module in Digital Creativity
- 200-300 students
- Students create a virtual diorama using C#
- Multidisciplinary
  - Art, Design, Music, Programming, Writing, etc.

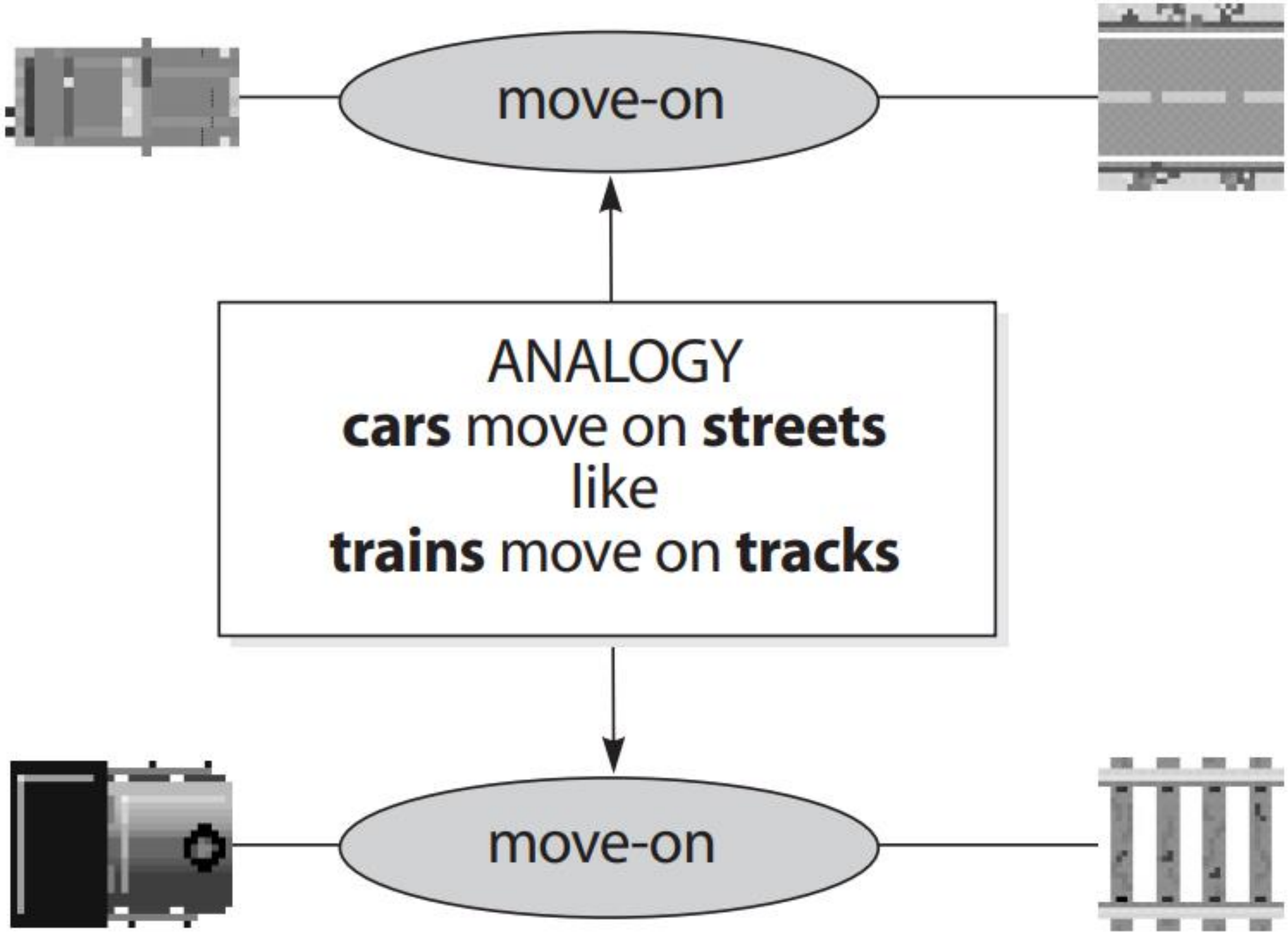
PUBLIC

100

Analogous Examples

> "Analogies are powerful cognitive mechanisms that people use to construct new knowledge from knowledge already acquired and understood…The combination of PBE with analogies is called Programming by analogous examples."

Repenning & Perrone (2000)



ANALOGY
cars move on **streets**
like
trains move on **tracks**

Contextual Examples

> "Context works if it’s related to the participant’s own experience: performance on the selection task is significantly facilitated when the presentation of the task allows the subject to recall past experience with the content of the problem, the relationship expressed, and a counter-example to the rule governing the relationship."

Griggs and Cox, 1982

Contextual Examples

A contextual example is a form of analogous example that is actively situated in a context that allows for recall of prior knowledge to enable the thematic facilitation of further learning

Challenges

- Matchmaking to situate examples in a relatable disciplinary context based on a student's background
- Tailoring material can be onerous and time-consuming to prepare; particularly in diverse modules that include learners from a wide range of backgrounds

Prior Work

- Brusilovsky (2001) proposes adaptive hypermedia systems
- Lubart (2005) outlines four ways computers can facilitate creativity: managing; supporting communication between collaborators; enhancing creativity; and cooperating in creative acts
- Mampadi *et al* (2011) show that adaptive hypermedia systems to tailor educational material to cognitive styles enhances learning
- Wikis increase in popularity, with potential for adaptive teaching of computing (Kim, Han & Han, 2006; González-Ortega et al, 2010; Lin, Wu & Chiu, 2018)
- Problem descriptions don't seem to help or hurt, beyond motivation, but studies focused on task instructions rather than contextualizing teaching material and have limited individualisation (Bouvier et al, 2016; Craig, Smith, and Petersen, 2017)

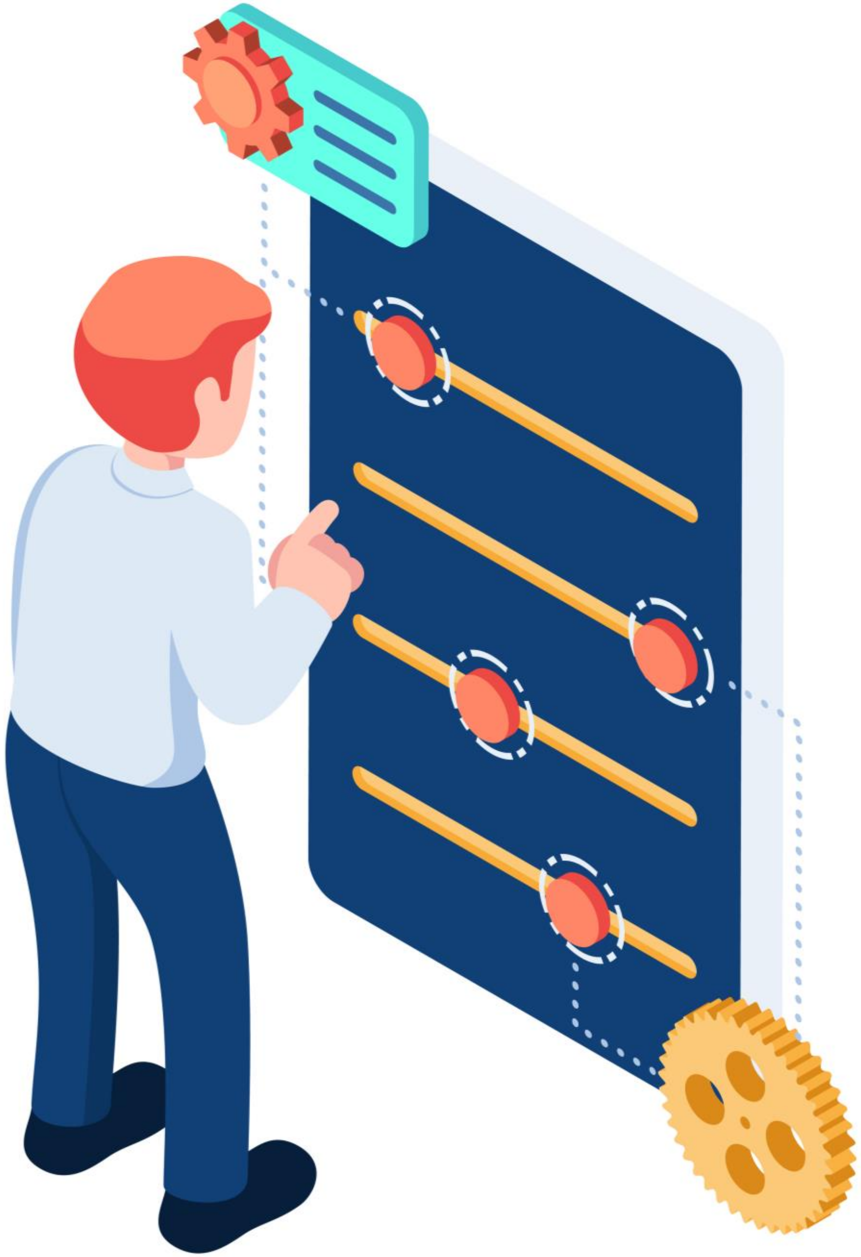
Prior Work

- Leinonen, Denny & Whalley (2021) find ontextualised problem solving in a programming contexts does help, but familiarity and underlying mathematics skills moderate the benefit
- Considerable research on the educational impact of ChatGPT on learning programming, showcasing its potential to automatically contextualise examples (Balse *et al*, 2023; Biwas, 2023; Ouh *et al*, 2023; Leinonen *et al*, 2023; Tian *et al*, 2023; etc.)
- Daun and Brings (2023) outline the threats and opportunities of ChatGPT for software engineering education, including the individualisation of education
- Chen *et al* (2023) present a ChatGPT-driven intelligent tutoring system for code explanation

PUBLIC

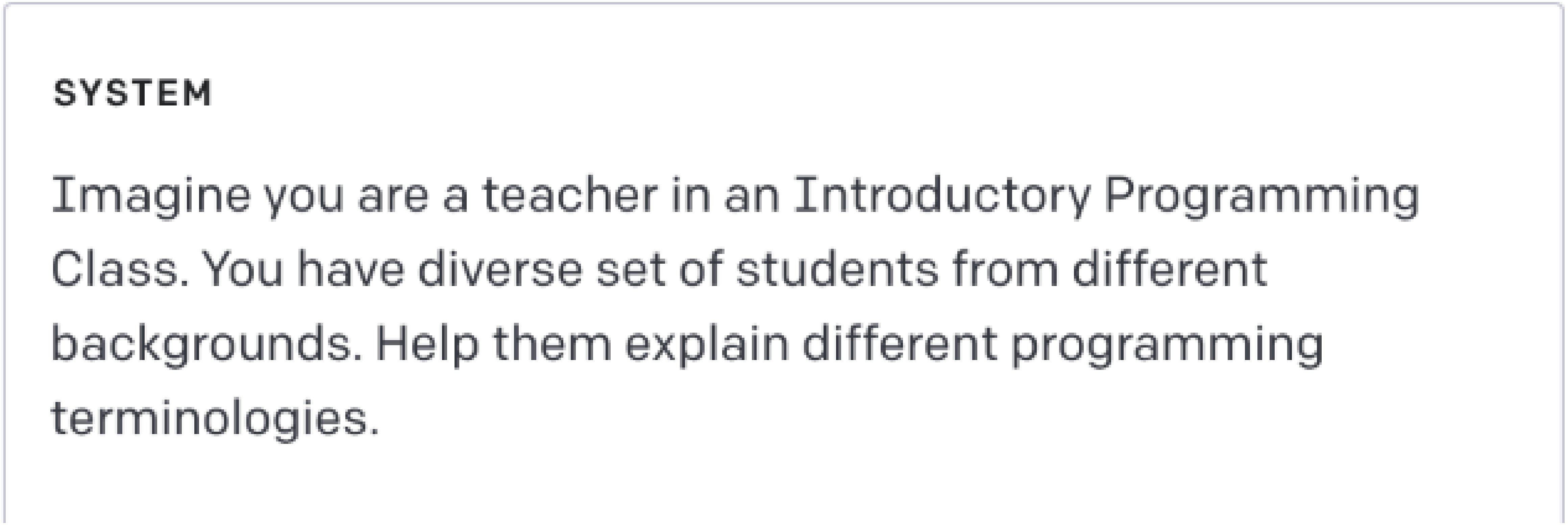
Research Question

To what extent does using ChatGPT to tailor contextual examples of programming concepts to disciplinary domains improve how students of those disciplines learn those programming concepts?



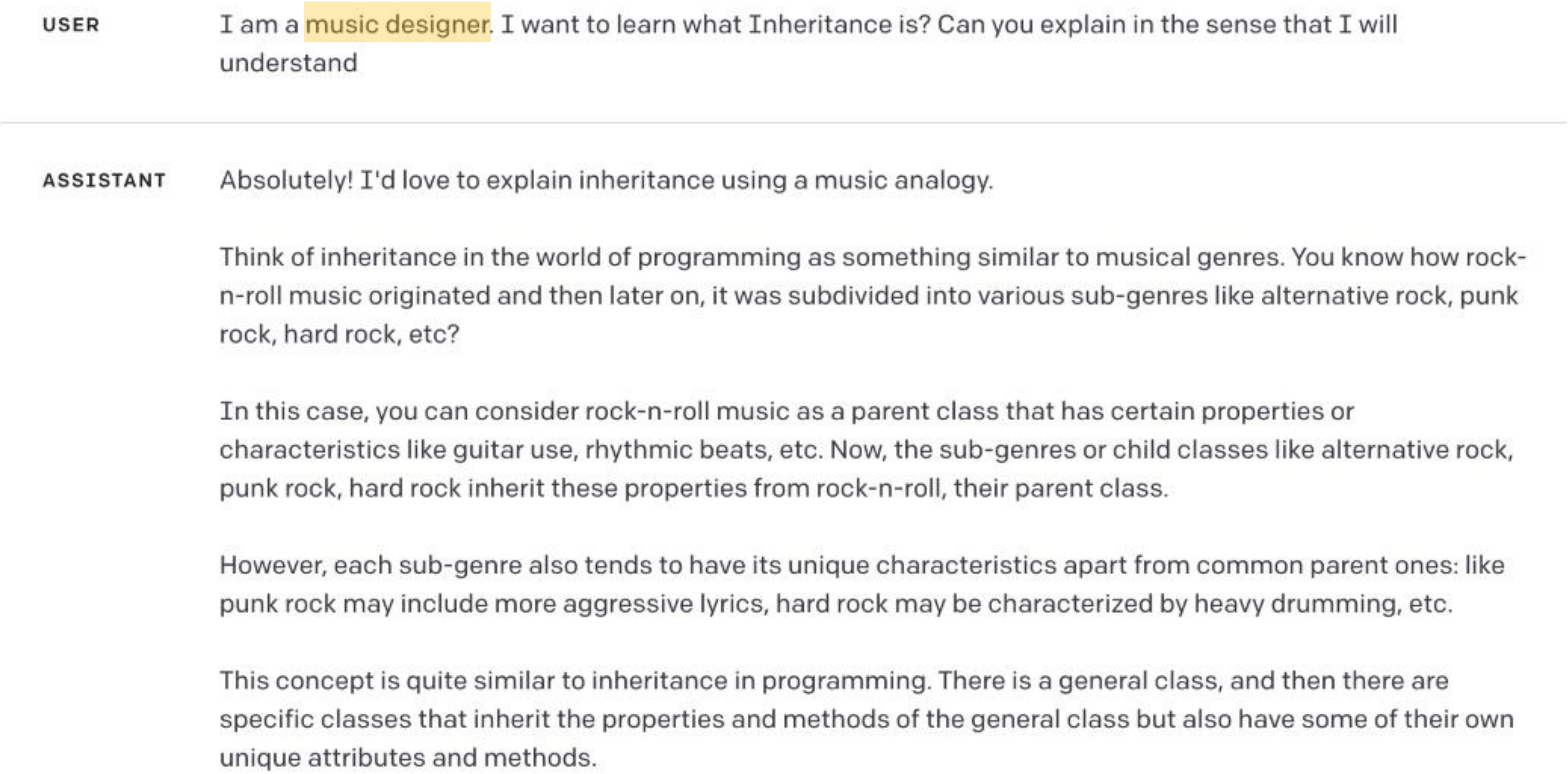
Tailoring Examples

- Wiki system integrated into our virtual learning environment, Moodle
- Currently populated with material about programming concepts typically encountered in CS1 and CS2 contexts
- Adding contextualised examples for 23/24
- Using ChatGPT (GPT-4) to produce analoguous examples for each disciplinary grouping: artist, designer, musician, programmer, writer, etc.
- Using a GET request (setup using a course variable in Moodle's URL resource) to display the correct version of the wiki page for students on each course



SYSTEM

Imagine you are a teacher in an Introductory Programming Class. You have diverse set of students from different backgrounds. Help them explain different programming terminologies.



USER I am a music designer. I want to learn what Inheritance is? Can you explain in the sense that I will understand

---

ASSISTANT Absolutely! I'd love to explain inheritance using a music analogy.

Think of inheritance in the world of programming as something similar to musical genres. You know how rock-n-roll music originated and then later on, it was subdivided into various sub-genres like alternative rock, punk rock, hard rock, etc?

In this case, you can consider rock-n-roll music as a parent class that has certain properties or characteristics like guitar use, rhythmic beats, etc. Now, the sub-genres or child classes like alternative rock, punk rock, hard rock inherit these properties from rock-n-roll, their parent class.

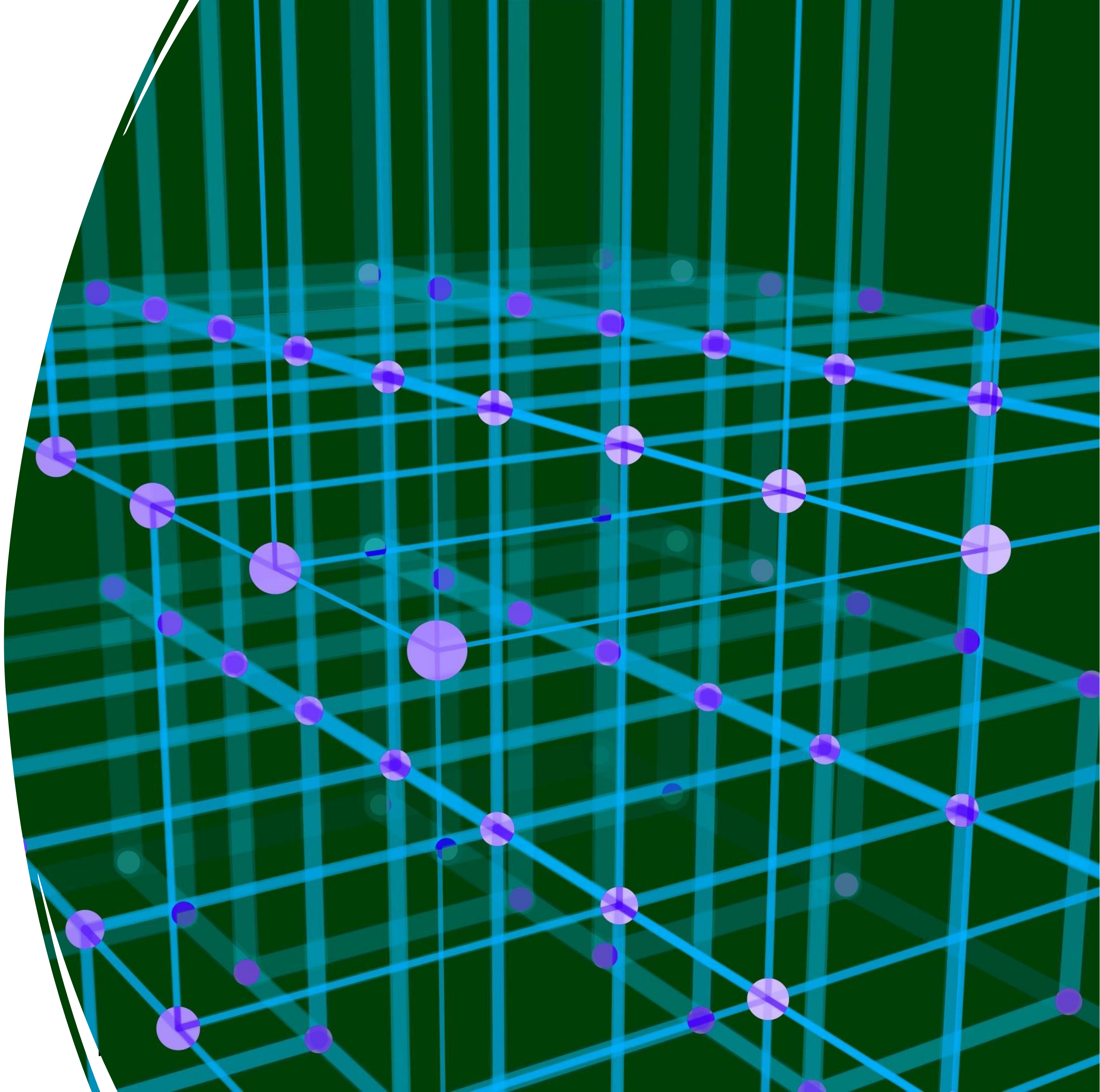
However, each sub-genre also tends to have its unique characteristics apart from common parent ones: like punk rock may include more aggressive lyrics, hard rock may be characterized by heavy drumming, etc.

This concept is quite similar to inheritance in programming. There is a general class, and then there are specific classes that inherit the properties and methods of the general class but also have some of their own unique attributes and methods.

Method

Pilot

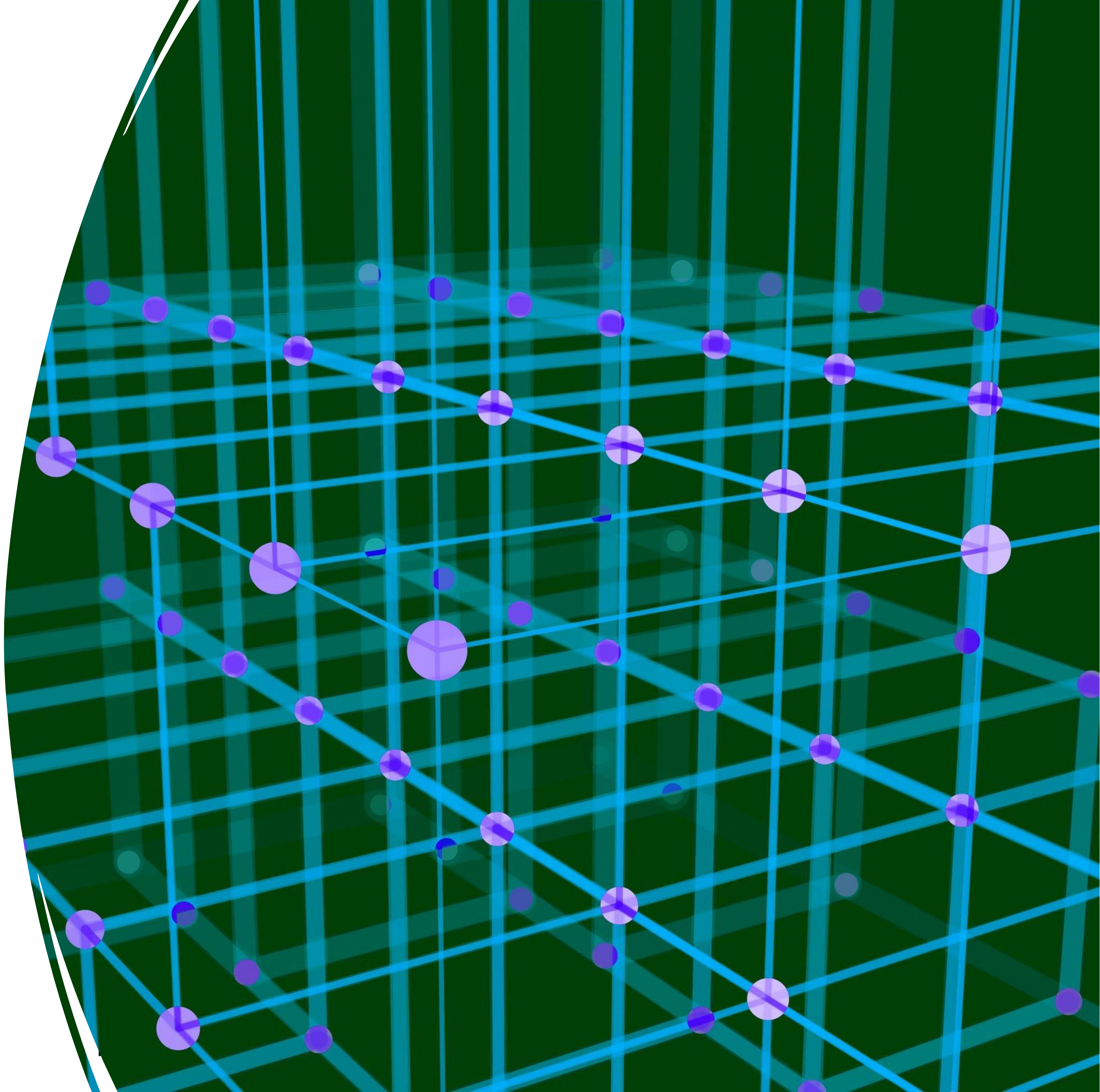
- Explanations were generated for object-orientated concepts (inheritance, polymorphism, abstraction, encapsulation) and circulated for comment
- Rated for "face validity" on a 3-point scale (bad, okay, good)
- Convenience sample of 10 colleagues in art, design, writing, etc. compared the existing example to the generated contextualised example
- Reviewed difference



Method

Future Study

- Experiment
- Three groupings:
  - Contextualised (Matched)
  - Contextualised (Mismatched)
  - Non-contextualised
- Three workshops:
  - Flow control structures
  - Inheritance and polymorphism
  - Common data structures
- Random allocation, within-participant



Method

	Flow Control	Inheritance and Polymorphism	Common Data Structures
Contextualized (Matched)	Group A	Group C	Grouo A
Contextualized (Mismatched)	Group B	Group A	Group C
Non-Contextualized	Group C	Group B	Group A

Each grouping is randomly allocated and will include students from across all the courses. The virtual learning environment will be set up to direct each individual student to the relevant wiki page based on their course enrollment and experimental grouping.

Pilot Findings

- Some of the generated examples are quite compelling
- Others need considerable tweaking in both the prompt and returned explanation
    - "Character artist" gave examples that were more closely tied to designers than artists, focusing on a fantasy role-playing game
- Initial evidence for face validity in comparing generated to existing

Considerations

- Would colleagues feel similarly positive in a blind review of examples?
- Will the wider student body receive the intervention as enthused as colleagues in the pilot?
- Will students reject expansions of our teaching material facilitated using artificial intelligence?
- What is the proportion of good tailoring to bad tailoring?
- Who owns the intellectual property for the wiki content?
- Who maintains the wiki content?
- Will bad content in the wiki undermine confidence?
- What level of regeneration/editing is still required?
- Will students spot and fix issues in the wiki?

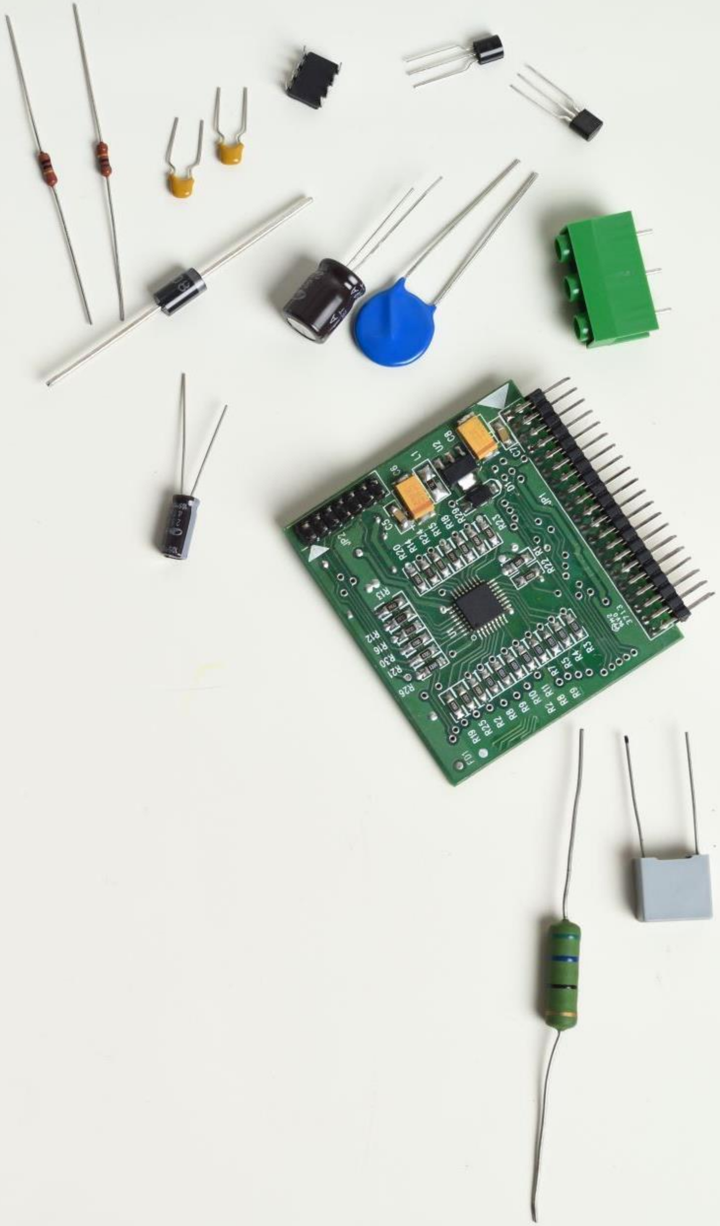
Further Work

Larger-scale rollout and trial in the Digital Creativity module running in 2023/24 at Falmouth University

Explore variables including:

- Continuation
- Attainment
- Attitude

If there is interest, a potential multi-national multi-institutional study facilitated using the Runestone ebook platform in 2024/25



Acknowledgements

We would like to thank Rishabh Balse and Jayakrishnan Warriem of the India Insititute of Technology Madras who encouraged us to pursue the project following conversations at the ACM ITiCSE conference.

References

- Lubart, T., 2005. How can computers be partners in the creative process: classification and commentary on the special issue. International journal of human-computer studies, 63(4-5), pp.365-369.
- Balse, R., Valaboju, B., Singhal, S., Warriem, J.M. and Prasad, P., 2023, June. Investigating the Potential of GPT-3 in Providing Feedback for Programming Assessments. In Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1 (pp. 292-298).
- Eng Lieh Ouh, Benjamin Kok Siew Gan, Kyong Jin Shim, and Swavek Wlodkowski. 2023. ChatGPT, Can You Generate Solutions for my Coding Exercises? An Evaluation on its Effectiveness in an undergraduate Java Programming Course. In Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1 (ITiCSE 2023). Association for Computing Machinery, New York, NY, USA, 54–60. https://doi.org/10.1145/3587102.3588794
- Daun, M. and Brings, J., 2023, June. How ChatGPT Will Change Software Engineering Education. In Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1 (pp. 110-116).

References

- Leinonen, J., Denny, P., MacNeil, S., Sarsa, S., Bernstein, S., Kim, J., Tran, A. and Hellas, A., 2023. Comparing code explanations created by students and large language models. *arXiv preprint arXiv:2304.03938*.
- Biswas, S., 2023. Role of ChatGPT in Computer Programming.: ChatGPT in Computer Programming. *Mesopotamian Journal of Computer Science*, *2023*, pp.8-16.
- Tian, H., Lu, W., Li, T.O., Tang, X., Cheung, S.C., Klein, J. and Bissyandé, T.F., 2023. Is ChatGPT the Ultimate Programming Assistant--How far is it?. *arXiv preprint arXiv:2304.11938*.
- Chen, E., Huang, R., Chen, H.S., Tseng, Y.H. and Li, L.Y., 2023. GPTutor: a ChatGPT-powered programming tool for code explanation. *arXiv preprint arXiv:2305.01863*.
- Malinka, K., Peresíni, M., Firc, A., Hujnák, O. and Janus, F., 2023, June. On the educational impact of ChatGPT: Is Artificial Intelligence ready to obtain a university degree?. In *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1* (pp. 47-53).

References

- González-Ortega, D., Díaz-Pernas, F.J., Martínez-Zarzuela, M., Antón-Rodríguez, M., Díez-Higuera, J.F., Boto-Giralda, D. and de La Torre-Díez, I., 2010. Using Wikis to Learn Computer Programming. In Knowledge Management, Information Systems, E-Learning, and Sustainability Research: Third World Summit on the Knowledge Society, WSKS 2010, Corfu, Greece, September 22-24, 2010. Proceedings, Part I 3 (pp. 357-362). Springer Berlin Heidelberg.
- Kim, S.H., Han, H.S. and Han, S., 2006. The study on effective programming learning using wiki community systems. In Innovative Approaches for Learning and Knowledge Sharing: First European Conference on Technology Enhanced Learning, EC-TEL 2006 Crete, Greece, October 1-4, 2006 Proceedings 1 (pp. 646-651). Springer Berlin Heidelberg.
- Lin, Y.T., Wu, C.C. and Chiu, C.F., 2018. The use of wiki in teaching programming: Effects upon achievement, attitudes, and collaborative programming behaviors. International Journal of Distance Education Technologies (IJDET), 16(3), pp.18-45.

PUBLIC

References

- Craig, M., Smith, J. and Petersen, A., 2017, November. Familiar contexts and the difficulty of programming problems. In Proceedings of the 17th Koli calling international conference on computing education research (pp. 123-127).
- Bouvier, D., Lovellette, E., Matta, J., Alshaigy, B., Becker, B.A., Craig, M., Jackova, J., McCartney, R., Sanders, K. and Zarb, M., 2016. Novice programmers and the problem description effect. In Proceedings of the 2016 ITiCSE Working Group Reports (pp. 103-118).
- Leinonen, J., Denny, P. and Whalley, J., 2021, February. Exploring the effects of Contextualized problem descriptions on problem solving. In Proceedings of the 23rd Australasian Computing Education Conference (pp. 30-39).
- Berssanette, J.H. and de Francisco, A.C., 2021. Cognitive load theory in the context of teaching and learning computer programming: A systematic literature review. IEEE Transactions on Education, 65(3), pp.440-449.

References

- Brusilovsky, P., 2001. Adaptive hypermedia. User modeling and user-adapted interaction, 11, pp.87-110.
- Mampadi, F., Chen, S.Y., Ghinea, G. and Chen, M.P., 2011. Design of adaptive hypermedia learning systems: A cognitive style approach. Computers & education, 56(4), pp.1003-1011.
- Anthony, P., Joseph, N.E. and Ligadu, C., 2013. Learning how to program in c using adaptive hypermedia system. International Journal of Information and Education Technology, 3(2), p.151.
- Repenning, A. and Perrone, C., 2000. Programming by example: programming by analogous examples. Communications of the ACM, 43(3), pp.90-97.
- R. A. Griggs and J. R. Cox. The elusive thematic-materials effect in Wason's selection task. British Journal of Psychology, 73(3):407--420, 1982.