Co-coding with AI in creative programming education

Anonymized
Anonymized
Anonymized, Anonymized, Anonymized

Anonymized
Anonymized
Anonymized, Anonymized, Anonymized

ABSTRACT

This paper presents a study of a group of university students using generative machine learning to translate from natural language to computer code. The study explores how the use of the AI tool can be understood in terms of co-creation, focusing on the one hand on how the tool may serve as a resource for understanding and learning, and on the other hand how the tool affects the creative processes. Findings show how the participants search for a 'correct' syntax in their instructions to the machine learning tool, and how the inconsistent and erroneous behavior can work as a way to generate clues and inspiration for generating creative expressions. The notion of friction is used to describe how systems like this can serve to both lower thresholds for programming, and also interfere with the creative processes, encouraging reflection and exploration of alternative solutions.

CCS CONCEPTS

• **Human-centered computing** → *Interaction design process and methods*; **Natural language interfaces**.

KEYWORDS

GPT-3, generative machine learning, co-creation, programming, post-human design

ACM Reference Format:
Anonymized and Anonymized. 2022. Co-coding with AI in creative programming education. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 5 pages. https://doi.org/xx

1 INTRODUCTION

In digitally mediated creative practices, such as interaction design and various forms of digital fabrication, smart or intelligent tools are becoming increasingly common to support the design process. This development aligns with conceptual developments in design research around notions such as post-anthropocentric [9] and post-human design [26], co-performance [13], more-than-human design [10] and machine agency [19] which all suggest a reconsideration of the view that humans should be viewed as the only source of creative agency, to instead see various forms of interactive and digital technologies as having the agency to spur ideas or form a creative expression. With recent developments in AI and generative machine learning [8] these forms of co-creativity become even more topical as several of these technologies are explicitly designed to act as or resemble intelligent forms of co-actors, for instance in producing various forms of fictional text, presentation material, or programming code. Recent development in AI research targeted at generative machine learning, or Generative Adversarial Networks (GAN) [8] has led to enormous improvements in the ability to generate original computer generated content. GAN-technologies have been used to generate images, audio and text in numerous contexts, for example realistic images of people that do not exist, and images based on textual descriptions. For this work we are particularly interested in the ability of these systems to interpret and translate natural language into working programming code. This means that you can ask the system using natural language to produce programming code that performs particular functions, such as simple cases of drawing a circle, as well as more advanced programming expressions.

We present a study of how a system for AI-generated programming code can co-exist with, and be put to use in a higher education programming course for interaction designers. The GPT-3 based Codex tool [6] for generating programming code based on natural language descriptions was introduced to a group of bachelor level students attending a course on creative programming in interaction design. The students were asked to use the tool to work with assignments similar to those that they had worked with previously during the course. They used the tool to produce code that formed creative graphic expressions and animations, both with instructions aiming for specific results (draw a red diamond in the middle of the screen), as well as open ended tasks aiming only for creating interesting expressions. The activity thus represents a form of co-creative design activity between participants and the AI-based tool. The study investigates questions concerned with co-creation in the particular context of learning programming as a tool for creative expression and design. To frame the challenges and sometimes contradicting ways that the AI system affected the processes of creation and sense-making among the participants, we propose the notion of *friction* [7, 14].

1.1 Co-creation in design and programming

The idea of computers generating content in collaboration with human designers has existed for a long time. Albaugh et al. [1] discuss CAD-systems as an example of a set of tools that can be categorized as "time saving systems" aiming at shortening the time or expertise required for each design iteration by delegating repetitive or time-consuming work to the machine, giving the users more space to focus on the creative aspects of design. Co-creation in relation to programming and programming education is also a fairly well-explored area, even if not often talked about in these terms. There are numerous tools designed to support and simplify various aspects of programming and programming practice. You could also consider the design of the programming languages themselves as

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/xx

creating support for co-creation between human and computer. A fundamental principle in computer programming is the use of abstractions and high-level representations. In high-level programming languages, hardware-specific commands are replaced with textual commands and a syntax that more or less builds on a formalized natural language syntax. There have been several attempts to create support for programming in natural language, for example by Miller already in 1981 [18], and later by Price et al. [21] These early examples however require that that the user adheres to a pre-defined syntax. This differs from the GPT-3 based system used in this paper, which is not restricted to a particular syntax or set of commands.

1.2 Post-human perspectives on co-creation

The conceptual underpinnings of this paper rely on theories of the interplay between human and machine with a broad foundation in the research field of Human-Computer interaction, but with a particular focus on post-human theories, and the recent articulation of post-human design [9, 27]. This connects to the broader conceptual conversation in HCI on material and machine agency [3, 10, 19, 22, 24] as being shared or distributed among humans, machines, and artefacts, and the reconsiderations of processes of design and interactivity that these imply. At its core, post-human perspectives on design rejects the traditional dichotomies between humans and machines in favor of a perspective on design in which humans and machines are considered to be co-creators in processes of ideation, design and making. These theories put to the fore how human agency must be understood as entangled with agencies that stem from non-human entities, e.g. as extensivly elaborated in notions such as agential realism [3] and machine agency [19]. Of particular relevance for the present work are studies that question traditional categorizations of programming and creativity as relying on step-by-step models as going from design ideas - to digital representations - to machine execution, and instead view humans, materials, and machines as working in conjunct in co-creative and co-performative ways. We argue that artificial intelligence systems are particularly interesting to explore as a form of non-human co-performer as they display properties that resemble those that would be ascribed to - or expected by - human actors, such as problem-solving and natural language interpretation. An important strand in post-human and post-anthropocentric interaction design has explored notions that challenge common expectations of AI systems to work as rational predictable actors, such as uncertainty [17], imperfection [11, 23, 25], and under-determination [1], and the consequences these have for the way we conceptualize and design tools and methods for these practices.

2 INSIGHTS AND OUTCOMES

From the point of view established practices of learning to program, there are conflicting views among the participants on how to understand and use a system such as this, as well as on the different ways it may aid the practice of computer programming in a creative context. Traditionally, computer programming has been framed as form of problem solving practice that resembles scientific thinking, but more recently creative [4], embodied [12], and craft-oriented [5] perspectives of programming has contributed to a shift in this view. These brings up questions of how the practice of programming should be understood in an interaction design context and how a system that "co-creates" with the users should be used and framed . Clearly, if the responses of the Codex-system is considered from the point of view of 'providing the code that the participants asked for' there are a considerable number of flaws and inconsistencies in how it works. However, as reflected in several accounts of the participants and as shown in previous work post-human design, uncertainty, inconsistencies or 'randomness' e.g., ([17, 23]) in the responses of the Codex system contributed to opening up for creativity and novel design ideas. Thus, viewing this interaction as a co-creative or co-performative activity, the outcome cannot be determined based on a literal interpretation of how the participants initially expressed their idea, but as a shared expression that unfolds in interaction between user and system. In the following sections, we will discuss the findings in relation to three broader themes relating to co-creation with artificial intelligence: 1) learning to 'speak' the language to creatively use the system, 2) co-creation as a process of reducing or inducing friction, and 3) managing user expectations by considering how the system is framed within the creative process.

2.1 Cracking the code

Our findings point in several directions with respect to how participants perceive the usefulness and the value of this system. In many of the accounts from the participants, they claim that the success of getting their intended response from the system depends on their ability to formulate sentences that the AI understands, thereby suggesting that there is some form of syntax that need to be figured out as one becomes more experienced in using the system. These accounts reflect research on voice-based conversational agents such as Alexa and Siri, showing how users tend to adapt and simplify their language in a way that is assumed that the system understands more easily. Such studies stand in contrast to work a graphical user interface, showing how natural language interfaces do not communicate its capabilities in the same way as graphical user interfaces. Instead, users bring expectations of using natural language from other settings, most commonly interaction with other natural language speakers, and adapt their interpretation of the system's response based on those [15, 20]. As noted above, while many participants became frustrated that the system did not generate code that worked in the fashion that they had foreseen, they also expressed value in the open-ended character of the interaction with the system. The open-endedness and unpredictability were thus interpreted positively as well as negatively. However, even if there is a value in that a system is not fully predictable - it might still be problematic when users expect that they have to figure out particular syntax when interacting with the system, when, in reality, there is none. As one participant said, *"it's like a mysterious system where you have to 'crack the code' in order to unlock the full potential of the system"*.

This particular mindset and understanding of the system can be attributed to that machine learning systems like the GPT-3 system used in this study can be described as 'black-box' systems, where it is inherently challenging to understand why the system behaves the way it does. Contemporary guidelines for the design of AI systems

[2] generally strive for increasing the transparency of the system by providing explaining what the system can do, how well it does it, and why it did what it did. In the context of co-creation and creative design it is however not obvious that it would be more beneficial with a completely transparent system where capabilities and means of communication are clearly articulated. The unpredictive and inconsistent behaviour instead points to that the behaviour of the system could be understood in terms of *animism*, as articulated by Marenko and van Allen [17]. In this context, animism should not be confused with anthropomorphism, as it is not a question of suggesting human-like capabilities and modes of communication. Instead, animism refers to a form of interaction that fosters the unexpected instead of prediction and linearity, and where conversations with things, rather than about or to things, take place. So learning to ’talk to the AI’ from an animistic perspective is neither about conforming to a particular syntax, nor expecting a communication that resembles that of another human. To ’crack the code’, becomes a matter of learning to treat the interactions with the system as contributions to a joint project of co-creation between human and machine, appreciating what can be experienced as unpredictability and inconsistency, rather than about unlocking what the machine is exactly capable of.

2.2 Reducing or inducing friction

As noted by Albaugh et al [1], systems aiming for co-creation with computers in creative contexts can be roughly divided into "time-saving systems", and "time-deepening systems", where the latter aims at "disrupting" or dehabituating an otherwise familiar practice, and supporting creative reflection. One way of rephrasing this is that the system may either reduce or introduce *friction* into the activity." Friction has previously been used in interaction design research for example by Laschke and Hassenzahl [14], where the notion of friction is described as a design quality that can be used to nudge users towards meaning making and reflective use. Similarly Cox et al. [7], talk about friction in terms of microboundaries that create obstacles, fostering a more mindful use of technology. Our findings show how the Codex-system to some extent served both to reduce friction, by e.g. providing programming code with accurate syntax that they could use as a starting point to explore an idea, but also to induce friction, by providing code that performed differently than the users expected it to, thereby challenging them to expand, refine or rethink their idea.

Our findings show numerous accounts of how the Codex-system may fill an important role in reducing friction in creative programming practices. Many participants pointed to that it was useful to get a ’starting point’ through a automatically generated piece of code that executes and is syntactically correct. Several participants point out that this could be particularly useful for inexperienced programmers. To reduce friction, it is important that the system responds in a consistent and predictable way, and that it actually produces code in line with the users’ expectations. From this point of view, the partially unpredictable behaviour of the system does not serve to lower friction, but was sometimes experienced to include flaws that would need to be mitigated by clarifying the system’s behaviour.

However, our findings also show how the creative processes might also benefit from introducing friction into the process. This becomes particularly relevant in cases where the designer has a less well-defined or even vague idea of what the end result should be. In such case, the system would instead play the role of co-exploring the design space and to provide novel directions and resources in a creative process. Our findings provide several examples where the participants initially aimed for a particular expression, but the system did not render exactly what they had hoped for. Instead the system generated code with unexpected elements, a form of friction, that required users into reflection and reconsideration . In many cases, this steered the design exploration in new and often more creative directions. In the context of creative design, this reflect work by e.g. [16], and [1], which have identified how aspects such as inconsistency, uncertainty and randomness etc, work as resources in a creative process.

These ways of experiencing the Codex-system represent conflicting views on how creative work can be supported, either by reducing or inducing friction. Understanding friction from these views requires that designers find ways of how to balance aspects of automation with aspects of interference in the creative process. In some cases, it is preferable for the system to more closely align with user’s literal articulations, whereas in other cases it might be better if the system interferes or diverges from them in order to expand or open up a design space. We argue that the concept of friction can capture both these dimensions of using and designing these kinds of systems and thus become a potentially useful concept in the context of co-creation with artificial intelligence.

2.3 Framing the system - tool or co-creator

In line with post-human design perspectives, introducing these kind of systems in creative practices such as interaction design and programming involves a repositioning of the relations between humans and machines in the creative process. In particular, as opposed to viewing design and creativity with such tools as uni-directional processes governed solely by the actions and intention of the human, this requires that we consider the use of these kind of systems as engaging humans and machines in co-creative dialogues. As our findings show, in a co-creative practice we cannot assume that a systems such as the one used here generates well-defined ideas or solutions, but rather such these systems should be understood as creatively contributing to the unfolding of the design process. By considering creative practice from this point of view, we argue that we contribute by challenging the idea of the designer as the sole source of the ideas and intents that form a particular creative expression and thereby, expanding the space for the creative potential of this kind of tools. By *framing* the activities with these kind of tools in different ways we can engage users in attributing various degrees of agency to the AI tool [13]. From one point of view, the system resembles a passive tool that just acts on the command from the user. If the user asks it to draw a circle, it should draw a circle. Sometimes the system will understand the commands correctly and produce a correct response, and sometimes it will fail to do so. From another point of view, the system can viewed as a co-performer or co-agent, which aligns with how a GPT-3 systems actually work. Understanding the interaction with these can be described through

Kuijer's and Giaccardi's notion of co-performance [13] to conceptualise the extent to which artefacts are capable of performing and exerting agency *together* with people. In contrast to notions of good or bad performance of a technological artifact, this shifts focus to the contextually situated 'appropriateness' of the joint human and artificial performances. In the context of this study, a focus on viewing the interaction as a joint performance, focus is shifted from how well the system output aligns with the user intention, and instead emphasizes resulting code interactive animated expressions as a joint achievement between user and system.

2.4 Conclusions

In this paper we have showed how the sometimes inconsistent and imperfect behaviour of an AI-based system for generating programming code provided various challenges for the participants engaged in a programming exercise. However, it also worked as a creative resource and as means to encourage reflection on the solution that they were developing. To frame the challenges that a system such as this might introduce, we propose the notion of *friction*, as a way to address the need to balance the qualities of automation with that of interfering in the creative process. We believe that finding ways of dealing with these two aspects are inherent to successful use of generative AI for co-creation in creative settings. We stress the importance of *framing* of the activities to the users when working with generative AI for co-creation. By designing settings of use that involve a post-human design perspective, the users' understanding of what can be expected from it can be addressed, thereby putting generativity and unpredictability at the core, rather than a well-defined query-response system.

REFERENCES

[1] Lea Albaugh, Scott E. Hudson, Lining Yao, and Laura Devendorf. 2020. Investigating Underdetermination Through Interactive Computational Handweaving. In *Proceedings of the 2020 ACM Designing Interactive Systems Conference (DIS '20)*. Association for Computing Machinery, New York, NY, USA, 1033–1046. https://doi.org/10.1145/3357236.3395538

[2] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N. Bennett, Kori Inkpen, Jaime Teevan, Ruth Kikin-Gil, and Eric Horvitz. 2019. Guidelines for Human-AI Interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3290605.3300233

[3] Karen Barad. 2007. *Meeting the Universe Halfway: Quantum Physics and the Entanglement of Matter and Meaning*. Duke University Press. Google-Books-ID: H41WUfTU2CMC.

[4] Ilias Bergstrom and R. Beau Lotto. 2015. Code Bending: A New Creative Coding Practice. *Leonardo* 48, 1 (Feb. 2015), 25–31. https://doi.org/10.1162/LEON_a_00934

[5] Leah Buechley, Mike Eisenberg, Jaime Catchen, and Ali Crockett. 2008. The LilyPad Arduino: using computational textiles to investigate engagement, aesthetics, and diversity in computer science education. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*. Association for Computing Machinery, New York, NY, USA, 423–432. https://doi.org/10.1145/1357054.1357123

[6] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating Large Language Models Trained on Code. *arXiv:2107.03374 [cs]* (July 2021). http://arxiv.org/abs/2107.03374 arXiv: 2107.03374.

[7] Anna L. Cox, Sandy J.J. Gould, Marta E. Cecchinato, Ioanna Iacovides, and Ian Renfree. 2016. Design Frictions for Mindful Interactions: The Case for Microboundaries. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '16)*. Association for Computing Machinery, New York, NY, USA, 1389–1397. https://doi.org/10.1145/2851581.2892410

[8] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A. Bharath. 2018. Generative Adversarial Networks: An Overview. *IEEE Signal Processing Magazine* 35, 1 (Jan. 2018), 53–65. https://doi.org/10.1109/MSP.2017.2765202 Conference Name: IEEE Signal Processing Magazine.

[9] Laura Devendorf, Abigail De Kosnik, Kate Mattingly, and Kimiko Ryokai. 2016. Probing the Potential of Post-Anthropocentric 3D Printing. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems (DIS '16)*. Association for Computing Machinery, New York, NY, USA, 170–181. https://doi.org/10.1145/2901790.2901879

[10] Elisa Giaccardi and Johan Redström. 2020. Technology and More-Than-Human Design. *Design Issues* 36, 4 (Sept. 2020), 33–44. https://doi.org/10.1162/desi_a_00612

[11] Miwa Ikemiya and Daniela K. Rosner. 2014. Broken probes: toward the design of worn media. *Pers Ubiquit Comput* 18, 3 (March 2014), 671–683. https://doi.org/10.1007/s00779-013-0690-y

[12] Martin Jonsson, Jakob Tholander, and Ylva Fernaeus. 2009. Setting the stage – Embodied and spatial dimensions in emerging programming practices. *Interacting with Computers* 21, 1-2 (Jan. 2009), 117–124. https://doi.org/10.1016/j.intcom.2008.10.004

[13] Lenneke Kuijer and Elisa Giaccardi. 2018. Co-performance: Conceptualizing the Role of Artificial Agency in the Design of Everyday Life. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3173574.3173699

[14] Matthias Laschke, Sarah Diefenbach, and Marc Hassenzahl. 2015. "Annoying, but in a Nice Way":. 9, 2 (2015), 12.

[15] Ewa Luger and Abigail Sellen. 2016. "Like Having a Really Bad PA": The Gulf between User Expectation and Experience of Conversational Agents. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. Association for Computing Machinery, New York, NY, USA, 5286–5297. https://doi.org/10.1145/2858036.2858288

[16] Betti Marenko. 2015. When making becomes divination: Uncertainty and contingency in computational glitch-events. *Design Studies* 41 (Nov. 2015), 110–125. https://doi.org/10.1016/j.destud.2015.08.004

[17] Betti Marenko and Philip van Allen. 2016. Animistic design: how to reimagine digital interaction between the human and the nonhuman. *Digital Creativity* 27, 1 (Jan. 2016), 52–70. https://doi.org/10.1080/14626268.2016.1145127 Publisher: Routledge _eprint: https://doi.org/10.1080/14626268.2016.1145127.

[18] L. A. Miller. 1981. Natural language programming: Styles, strategies, and contrasts. *IBM Systems Journal* 20, 2 (1981), 184–215. https://doi.org/10.1147/sj.202.0184 Conference Name: IBM Systems Journal.

[19] J. Brian Pickering, Vegard Engen, and Paul Walland. 2017. The Interplay Between Human and Machine Agency. In *Human-Computer Interaction. User Interface Design, Development and Multimodality (Lecture Notes in Computer Science)*, Masaaki Kurosu (Ed.). Springer International Publishing, Cham, 47–59. https://doi.org/10.1007/978-3-319-58071-5_4

[20] Martin Porcheron, Joel E. Fischer, Moira McGregor, Barry Brown, Ewa Luger, Heloisa Candello, and Kenton O'Hara. 2017. Talking with Conversational Agents in Collaborative Action. In *Companion of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*. ACM, Portland Oregon USA, 431–436. https://doi.org/10.1145/3022198.3022666

[21] David Price, Ellen Riloff, Joseph Zachary, and Brandon Harvey. 2000. NaturalJava: a natural language interface for programming in Java. In *Proceedings of the 5th international conference on Intelligent user interfaces (IUI '00)*. Association for Computing Machinery, New York, NY, USA, 207–211. https://doi.org/10.1145/325737.325845

[22] Lucy Suchman. 2017. Agencies in Technology Design: Feminist Reconfigurations*. In *Machine Ethics and Robot Ethics*. Routledge. Num Pages: 15.

[23] Jakob Tholander and Maria Normark. 2020. Crafting Personal Information - Resistance, Imperfection, and Self-Creation in Bullet Journaling. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376410

[24] Jakob Tholander, Maria Normark, and Chiara Rossitto. 2012. Understanding agency in interaction design materials. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. Association for Computing Machinery, New York, NY, USA, 2499–2508. https://doi.org/10.1145/2207676.2208417

[25] Vasiliki Tsaknaki and Ylva Fernaeus. 2016. Expanding on Wabi-Sabi as a Design Resource in HCI. In *Proceedings of the 2016 CHI Conference on Human Factors in*

Computing Systems (CHI '16). Association for Computing Machinery, New York, NY, USA, 5970–5983. https://doi.org/10.1145/2858036.2858459

[26] Ron Wakkary. 2020. A Posthuman Theory for Knowing Design. 14, 3 (2020), 12.

[27] Ron Wakkary. 2021. *Things We Could Design: For More Than Human-Centered Worlds.* MIT Press. Google-Books-ID: UiY6EAAAQBAJ.