

Leopold-Franzens Universität Innsbruck  
Fakultät für Mathematik, Informatik und Physik

Institut für Informatik

Semantic Technology Institute Innsbruck



## Paper

### **Improving Computing Education Research through Valuing Design**

from

Luca Kemenater  
(Matr.-Nr.: 12116603)

Submission Date: 07.01.2024

# Contents

<b>1</b>	<b>Abstract</b>	<b>1</b>
<b>2</b>	<b>Introduction</b>	<b>2</b>
<b>3</b>	<b>Body</b>	<b>4</b>
3.1	Value of Design . . . . .	4
3.2	Intersection . . . . .	5
3.3	Advantages . . . . .	5
3.4	Challenges for Learners . . . . .	6
<b>4</b>	<b>Conclusion</b>	<b>7</b>
	<b>References</b>	<b>8</b>

# 1 Abstract

This paper shows the important role of design in Computing Education Research (CER) and the possibility of reshaping educational methodologies. Focused of the many faces of design this paper shows the significance of these approaches. By examining the design driven educational tool Scratch, it shows the advantages and disadvantages for learners. But more it shows the potential of a collaborating between CER and design. Ultimately, the paper appeals to an open minded approach within the CER community to implement design driven paradigms to enrich the spectrum of diverse learners.

## 2 Introduction

In the expansive domain of computing research education (CER) [3], the focus centers mostly around refining pedagogical approaches, assessing learning outcomes and conducting empirical evaluations. The essential image of a student revolves around diligently programming within software environments and shapes the core of this research, focusing on enhancing foundational learning processes through the creation of diverse examples and instructional methods.

However a crucial fact remains noticeable absent, a fact that according to the *ACM Transactions on Computing Education (TOCE)* is integral for developing the learning paradigms: design. The intentional design of educational tools, methods, mostly overlooked within the research of CER, holds the revolutionizing potential to improve the very frameworks within which computing education works.

While TOCE is welcoming design focus submissions on how to improve these very frameworks, CER is still dedicating substantial efforts on perfecting the techniques and is therefore not welcoming design based publications in their venues to boost the discussion between designers and developers.

The clear overlook of design within CER shows the need for a focus shift or adjustment towards integrating design thinking and methodologies into educational research. Embracing these principles and ideas has not only the potential to augment efficiency but also to widen the learning environment into a more creative and inclusive landscape.

## *2 Introduction*

Hence an acknowledgment on the potency of design within CER needs a reevaluation, a reevaluation that comprehends between empirical evaluations and design centered considerations to improve the computing education.

## 3 Body

### 3.1 Value of Design

Within the realm of CER, the concept of design spans a broad spectrum, manifesting in different forms. It encompasses process innovation, transparent methodologies, and creative outcomes, all with the same goal: to critique and re-imagine existing systems and practices to expand the field of computer education for a a diverse and bigger audience. Looking to the field of *Human-Computer Interactions (HCI)* sheds light on the pivotal role of design. HCI does not only use design principles but also values them with scholarships. This encourages designer to invest time into this field which would be essential for CER to progress and develop as a hole. If you consider the work of Sten Menör, who is a pioneer contributor in blocked-based programming, in the International Journal of Man-Maschine Studies shows the importance and impact of design driven approaches within educational tools. The focus of his work was not only about empirical evaluations but the importance of innovative design itself. It is a basic for today's tools like Alice or Scratch, which millions of young people use to get their fist hand on programming. Interestingly, had Menör published his work within CER, it might have faced challenges in reaching the wider audience it has today.

## 3.2 Intersection

Implementing design driven does not merely mean aesthetic elements only, it means more adaptable learning structures that foster those who need a more diverse style of learning rather than traditional instructions. The essence in this intersection lies in creating a more dynamic rather than static environment. It drifts away from the "one size fits all" approach to a more flexible, inclusive and most importantly engaging educational experience to reason with learners across various learning spectrums.

## 3.3 Advantages

When delving into the advantages of design-driven educational tools, Scratch [1] stands out as a remarkable example, epitomizing widespread usability, functionality, and adaptivity within Computing Education Research.

Scratch's user-friendly interface serves as a pivotal factor contributing to its popularity. This accessibility acts as a gateway for young learners, offering an interactive platform to explore, exchange ideas, and comprehend intricate programming instructions and principles. Scratch exemplifies how design-driven projects can effectively promote creativity and collaboration among learners.

- **Visual Coding Environment:** Scratch's visual block-based coding interface empowers beginners to code without facing syntax barriers. This innovative approach allows users to create diverse projects such as stories, games, and animations with ease.
- **Community Learning:** The collaborative nature of Scratch's platform fosters a sense of community-driven learning. Users gain valuable insights into various problem-solving approaches by exploring and modifying others' projects. This aspect encourages peer learning and the exchange of diverse perspectives.
- **Iterative Learning Opportunities:** Scratch's design facilitates iterative

learning through its ease of project building. Users can seamlessly build upon existing projects or merge different creations together. This feature promotes experimentation, creativity, and continuous skill development within an inclusive and supportive environment.

## 3.4 Challenges for Learners

While design-driven educational tools like Scratch offer numerous benefits, learners might encounter certain challenges that can affect their overall experience and work life afterwards[2]:

- **Over reliance on Visual Coding:** When switching to text-based programming learners could face the problem of not understanding, because they rely too heavily on graphical interfaces.
- **Lack of Depth in Understanding:** When learning such difficult principles the "easy" way rather than the "right way" could result in a lack of understanding of the basic principles of programming and could hinder them from solving complex problems.



## 4 Conclusion

In Conclusion this paper highlights the important role of design in reshaping Computing Education Research (CER) and emphasizes its significance in transforming educational paradigms. By exploring this, it becomes evident that design driven methods don't intervene with the traditional instructions but rather offers a more inclusive and engaging educational experience.

Embracing design-driven approaches in CER has the potential to revolutionize learning outcomes, fostering adaptability and inclusivity within educational structures. This would lead to a wider spectrum of audience and learners.

In the example of Scratch it shows the possibilities of such design driven tools, it may not be optimal yet, because the challenges are quit high but with a collaboration of CER and Design these hurdles could be overcome.

Concluding we can say that the CER community holds an immense potential to revolutionize learning outcomes and should therefore be more open to the new generation of novel designs for learning.

# References

- [1] Maloney, J., Resnick, M., Rusk, N., Silverman, B. and Eastmond, E. [2010], ‘The scratch programming language and environment’, *ACM Transactions on Computing Education (TOCE)* **10**(4), 1–15.
- [2] Meerbaum-Salant, O., Armoni, M. and Ben-Ari, M. [2011], Habits of programming in scratch, *in* ‘Proceedings of the 16th annual joint conference on Innovation and technology in computer science education’, pp. 168–172.
- [3] Shapiro, R. B., DesPortes, K. and DiSalvo, B. [2023], ‘Improving computing education research through valuing design’, *Communications of the ACM* **66**(8), 24–26.