

Need-Based Game Design: Developing a CT Game Based on Educational Needs

Niklas Humble, Peter Mozelius and Lisa Sällvin

University of Gävle, Sweden

Mid Sweden University, Östersund, Sweden

niklas.humble@hig.se

peter.mozelius@miun.se

lisa.sallvin@miun.se

Abstract: According to reports and previous research, present and future society show a high need for citizens with technical skills, such as computer programming. The need for technical competent citizens can however be challenging for higher education institutions to accommodate due to low numbers of students in higher education computer science. But also due to the high drop-out rates and low pass rates in introductory programming courses. With that said, it is important to both recruit new students to computer science and to support those who already study the subject, to meet the demands of the present and the future. An approach that has shown great promise in motivating students while still keeping focus on the learning aspects in education, is game-based learning (GBL). To facilitate support in higher education computer science, more specifically courses in fundamental programming, this study had the aim of identifying important design concepts in the development of a need-based educational game on computational thinking (CT). Skills related to programming have commonly been organized as part of the CT concept. To identify the educational needs, e-mail interviews were conducted with teachers that teach higher education courses on computer programming in Swedish universities. Based in an on-going design science project to create a learning game on computational thinking for higher education, the study discusses and compare the identified needs with design theories, such as affordances, emotional design, and sustainable design. The conclusion is a conceptual design of the educational game with a set of recommendations for a need-based game design in educational context.

Keywords: Educational game design, Design science, Design theory, Need-based design, Computational thinking

1. Introduction

Reports and previous research indicate that technical skills and computer programming will be important in both present and future societies (Smit et al., 2020; Wolz, Bergande & Brune, 2022). As digitalization continues to advance and new technologies such as AI have a greater impact, the demand for such skills will escalate (Gulliksen et al., 2020; Johnson et al., 2021). However, higher education institutions face challenges in meeting the demand for technically competent citizens. Factors such as low enrolment numbers in computer science (CS) programs and high dropout rates contribute to this problem (Hawlitcshek et al., 2020). Previous research also highlights the potential problem for learning programming if there is a lack of understanding during the initial contact with programming (Shuhidan, Hamilton & D'souza, 2009; Cheah, 2020).

Many CS programs globally start with an introductory programming course. The outcomes of such courses can be disappointing, as students tend to struggle with grasping fundamental programming concepts, procedures, and processes (Bennedsen & Caspersen, 2007; Cheah, 2020). This challenge has been addressed in prior research, such as Gorson and O'Rourke (2020). According to Rahman, Anuar and Said (2018) and Settle, Vihavainen and Sorva (2014), a possible factor for students dropping out of programming courses in later stages of their studies is a lack of motivation during the early stages. Given these challenges, it is crucial to not only attract new students to the field of computer science but also provide support and motivation to those who are already studying the subject. This is essential in order to meet the demands of present and future workforce.

Game-based learning (GBL) has emerged as a promising approach that effectively motivates students while simultaneously maintaining focus on learning (Klopfer, Osterweil & Salen, 2009; Humble, Mozelius & Sällvin, 2021; Humble, N., & Mozelius, P., 2022). Previous research has also shown promising results regarding students' engagement and attitudes towards computer programming when using GBL (Papadakis & Kalogiannakis, 2019). This study is part of a project aimed at developing an educational game that teaches fundamental programming in higher education, with a focus on Computational Thinking (CT). The aim of the study was to identify important design concepts in the development of a need-based educational game on computational thinking (CT). The research question to answer was:

RQ1. What are important design concepts in development of need-based educational games on computational thinking?

2. Theoretical Background

It should be noted that there is no absolute consensus on what constitutes design theory, but rather there are some characteristics of design theory that are more shared than others (Baskerville & Pries-Heje 2010). For example, design theory is often considered prescriptive, explaining how design should be conducted to improve, make more efficient and so forth (Walls, Widmeyer & El Sawy, 1992; March & Smith, 1995). The practical nature of design theory is often highlighted, for example, that “Design theories are considered as theorized practical knowledge” (Goldkuhl, 2004). For the design of the artefact, this study focuses on three specific theories for design (affordances, emotional design, and sustainable design) which are further described in subheadings below.

2.1 Affordances

James Gibson coined and developed the theory of affordances (Bower & Sturman, 2015; Stendal, Thapa & Lanamäki, 2016), which has commonly been defined as the action possibilities that the environment provides (Gibson, 2015; Bower & Sturman, 2015). Donald Norman later introduced the theory to Human-Computer Interaction (HCI) and further developed it to apply in research and design of artefact (Stendal, Thapa & Lanamäki, 2016). Norman (1999) specified the concept of affordances to also include perceived affordances, constraints, and conceptual models. According to Norman (2008), affordances as intended by Gibson did not have to be perceived by a user. However, since this is a crucial aspect in the design, Norman (1999) introduced the concept to the theory. Norman also frequently mentions two other concepts related to affordances: constraints and conceptual models. Constraints inform the user of which actions are not possible to perform with an artefact, these are often based on physical, logical, or cultural constraints (Norman, 1999). Norman (1999) have stated that conceptual models are the most important aspects of design, since these underly all other parts of design and are what everything else should be consistent with. A conceptual model can be understood as an explanation of how an artefact works, a simplified and useful description (Norman, 2013).

2.2 Emotional Design

According to Ho and Siu (2012), emotions and design are mainly being combined from the 1990s, while design previously (from the 1950s) focused primarily on the functionality of artefacts. In the HCI field, Norman’s model for emotional design is often considered one of the more influential (Adelson, 2010; Preece, Sharp & Rogers, 2015; Pengnate & Sarathy, 2017). Norman (2005) divides design into three levels that are processed by humans when interacting with an artefact, which together form a holistic experience: visceral, behavioral, and reflective. The initial impact or appearance of an artefact is referred to in the visceral design (Norman, 2005). The visceral design could also be said to classify the artefact’s attractiveness (Norman, 2005). The look and feel of an artefact, or the experience in total, is referred to in the behavioral design (Norman, 2005). It could also be said that the behavioral design is about classifying the artefacts regarding to their functionality and usability (Norman, 2005). Lastly, the thoughts and feelings afterwards, or the image and message left on the user, is referred to in the reflective design (Norman, 2005). The reflective design could also be said to classify the degree of prestige associated with an artefact (Norman, 2005).

2.3 Sustainable Design

Sustainable development emerged as a concept with pressing concerns for scarcity of resources in the 1960s (Carson, 1962; Hardin, 1968; Baldassarre et al., 2020). Sustainability is often understood as a collection of different types of sustainability, such as environmental, social, and economic (Mota et al., 2015). Combining and balancing these different aspects of sustainability has proven to be a challenge in, for example, production, consumption, and supply chain management (Mota et al., 2015; Azapagic et al., 2016). In the field of HCI, sustainability is addressed in, for example, Sustainable Interaction Design (SID) (Blevis, 2007). Blevis (2007) presents SID as a general framework for describing sustainability in relation to design values, design methods, and design reasoning. According to Blevis (2007), SID provides a discourse for reflecting on sustainability as a design value and relating sustainability to other design values. As a method for design, SID incorporates the idea to be mindful of the effects that interaction design can have on environmental sustainability and on behaviors that in turn effect environmental sustainability (Blevis, 2007). Lastly, SID can also be used for reasoning on the accountability of interaction design for sustainability, in terms of, for example, disposal, recycling, sharing, use and re-use (Blevis, 2007).

3. Method

This study is part of an on-going design science project with the purpose of creating a learning game on computational thinking for higher education. The design science process outlined by Johannesson and Perjons (2014), containing five phases, have been used for this project. The five phases are:

1. Explicate problem
2. Define requirements
3. Design and develop artefact
4. Demonstrate artefact
5. Evaluate artefact

The first phase, Explicating the problem, was carried out and described in a previous study (Mozelius, Sällvin & Humble, 2023) and in the introduction section of this paper. The focus for this study is on phase 2 and phase 3 of the design science process. To define the requirements (phase 2) and outline the design and development of the artefact (phase 3), this study has collected data through structured e-mail interviews with higher education teachers (further described in 3.1 Data collection) and compared the results to design theories (further described in 2. Theoretical background) in the discussion section of this paper. The conclusion of this paper is a conceptual design-model for a need-based game design, based on the findings from this study. This conceptual design-model will guide the continuing development of the artefact, which will be demonstrated (phase 4) and evaluated (phase 5) in a future study.

3.1 Data Collection

Structured e-mail interviews were sent to 18 teachers that teach higher education courses on computer programming at three different Swedish universities. The 15 interviews that so far has been answered have been used to define the requirements for the educational game. This is a continuation and extension of the preliminary requirement definition that was carried out in a prior study (Mozelius, Sällvin & Humble, 2023), a study that built on an analysis of the first 6 answered interviews. All interviews have been conducted with the same questionnaire with open-ended questions, and with the principle of keeping all informants as anonymous as possible during the research process. Informants have been selected according to the idea of a purposive expert sampling (Etikan, Musa & Alkassim, 2016), where all the interviewed teachers have a long and rich experience of introductory programming courses in higher education.

3.2 Data Analysis

In the first analysis, one of the investigators carried out a separate analysis of six interviews according to the open coding method as outlined by Khandkar (2009). In the open coding phase, interview data were read, reread, and colour coded to identify data extracts, codes, and potential categories. The second analysis phase used the Grounded theory idea of axial coding that has been described as: "coding that treats a category as an axis around which the analyst delineates relationships and specifies the dimensions of the category" (Bryant & Charmaz, 2007, p. 603). Moreover, the data were reassembled into more abstract conceptual categories in the axial coding with potential relationships between the categories. The focus in axial coding is often put on one central category (the phenomenon), with its relations to the other categories. Independent on the studied phenomena, there are always relations and interactions between the central phenomena and other categories (Vollstedt & Rezat, 2019).

In the axial coding process in this paper, data were organized around two central categories: 'Imperative Programming' and 'Object-Oriented Programming'. These two categories were chosen because they were central in the collected data, mentioned by several of the participating higher education teachers, and highlights aspect of what could be considered important for the game design (relates to the study's aim and research question). Imperative programming is often the programming paradigm that students are introduced to and could be described as a way of writing code where the control flow of the program is defined (Corral et al., 2014). Object-oriented programming (OOP) on the other hand, is a programming paradigm that encourages the programmer to think about the code in terms of objects with properties and behaviours, much like in the real world (Corral et al., 2014). These

4. Results

Result from the analysis is presented in sub-headings below. As described in the data analysis-section, findings have been organized around two central categories: 'Imperative Programming' and 'Object-Oriented

Programming'. However, the categories addressed around each central category are not necessarily exclusive, but rather represent the context and relationship expressed by the study participants. Quotes from the participants have been translated to English and to some extent rewritten from readability, but the underlying meanings have not been changed.

4.1 Imperative Programming

Categories related to imperative programming were the most common after the analysis. A total number of 13 categories were identified. These were organized in 3 super-categories for easier overview in this paper: a new mindset, basics, and design (Figure 1).

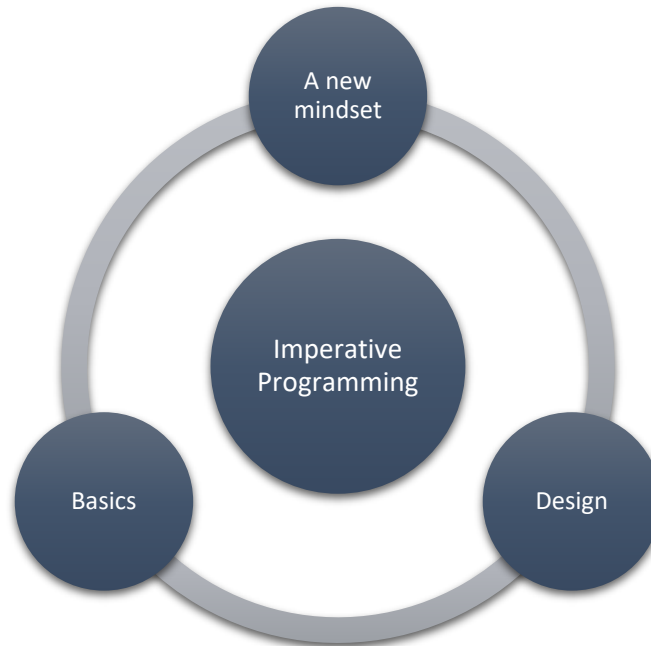


Figure 1: Super-categories organized around the central category 'Imperative Programming'

Study participants state that programming requires a new mindset that is often challenging for students. It requires, for example, a new way of thinking, understanding, abstracting, and looking at the whole. This new mindset is something that the game should address according to participants. Further, it is highlighted in the material that some students have difficulties to move from theoretical knowledge about programming to writing code that works. It is suggested that the game could address this, and other challenges that students face when learning programming, through debugging and problem-solving exercises.

In the collected material, participants highlight, for example, programming basics such as variables, selection, iterations, operators, lists, functions, and syntax. To some extent, it differs between the participants if they find some specific programming basics as challenging or easy for students in general. One participant state that "It's relatively easy for many to understand how to create the fundamental building blocks of programming.". While another participant state that "Many are in a hurry to advance and gets sloppy with the basics.". However, most agree that programming basics should have a central role in the game.

Lastly, study participants gave several suggestions for game design elements and games that they liked. Among these, Codewars, Scratch and RPG-style games were mentioned as examples of what the participants liked. Besides these, participants highlighted that the game should have a point system, encourage play, be dynamic to the player's skill level, and not be boring-looking. One participant state that "I think it's important that the game show progression, that you can see how far you have come and how far you have left". Participants also noted that the game should visualize programming for the students.

4.2 Object-Oriented Programming

Some of the categories identified through the analysis were more relevant in relation to object-oriented programming (OOP). These have been organized in the categories: classes and objects, OOP mindset, and methods (Figure 2).

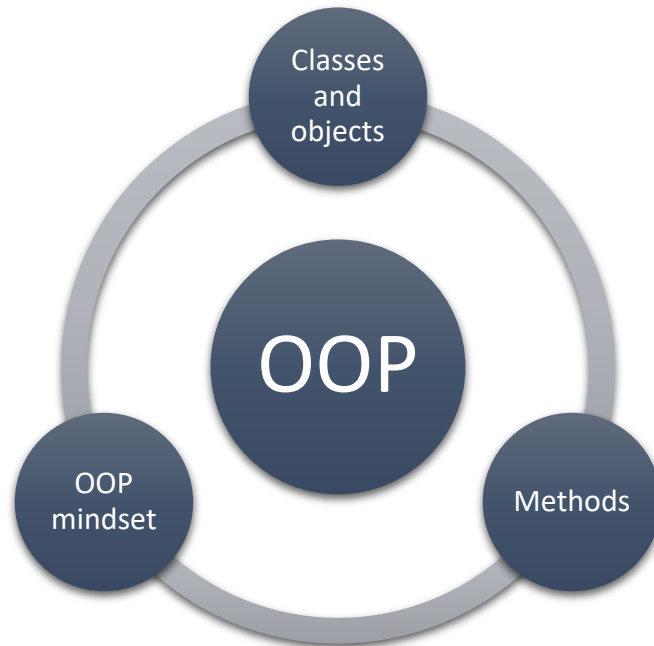


Figure 2: Categories organized around the central category 'Object-Oriented Programming'

Regarding classes and objects, study participants highlight different views if students find this challenging or easy. One participant state that simple classes and inheritance is often easy for students, while another participant state that “It’s the same with classes, there it varies from not being able to write a simple class to being able to write a class and its content but not understanding how to use it”.

In relation to classes and objects, study participants state that students have challenges with learning the concept of methods. This can be especially challenging if they just have grasped the concepts of variables, selection, and iterations. One study participant state the following about students’ challenges with methods: “Regarding specific programming techniques, my perception is that many have problems with the transition to methods, classes and objects”.

As in the previous section with programming as a new mindset, object-oriented programming is highlighted in the collected material as yet another level of difficulty. It is noted that object-oriented programming requires the students to look at data and logics as a whole, which can be challenging. One participant expresses the following about the challenges of learning object-oriented programming: “Even the simplest program requires more lines of code that you must learn by heart. Just like a musician needs to practice the difficult passages to get them in the muscle memory, a programmer needs to practice and memorize”.

5. Discussion

In this section, the results from the study are discussed and compared to the theoretical background to identify important design concepts for the development of a need-based educational game on computational thinking (CT). However, the discussion is mainly focused on the results concerning imperative programming. The reason for this being that imperative programming was highlighted to a larger extent in the collected data and is more commonly associated with fundamental programming courses. This makes imperative programming a better alternative for design and development of a need-based educational game on computational thinking, compared to object-oriented programming. The notions brought forward for OOP in this paper could be further investigated in a future study and be part of a second game development project.

5.1 Actions

What the player can and cannot do in the game is of course a central aspect of game design and should be carefully considered in the development. Further, the player’s interaction with the game can be related to the concepts of action possibilities (affordances) and constraints (Norman, 1999; Gibson, 2015; Bower & Sturman, 2015). Similar to Norman’s (1999; 2008) statement of perceivable action possibilities as a crucial part of design, participants in the study highlight actions they consider important for an educational game on CT. Players should for example be able to practice the basics of programming, such as variables, selection, iteration, and syntax.

This was considered important by many of the participants in the study, although some experienced that students already find this easy, while other experienced that students could be sloppy with the basics. Players should also be able to perform actions in the game that support them in developing a new mindset, which participants in the study highlighted as necessary when learning programming.

5.2 Engaging

Participants in the study have highlighted that it is important that the game, for example, encourages play, is dynamic to the player's skill level, and is not boring-looking. In short, the game should be engaging. Emotional design could be a path for achieving engagement in game design, especially the three levels of visceral, behavioural, and reflective design (Norman, 2005). Participants' notion of that the game should not be boring-looking could be a way of stating that it should be attractive (visceral). That the game should support players in moving from theoretical knowledge to practice, writing code, is about the usability of the game and could be said to relate to the behavioural level of emotional design. The reflective level of emotional design can be difficult since it refers to the image of the user, and this can be different between different users. For example, participants stated that they liked Scratch, Codewars, RPG-style games and so forth. It is therefore important to consider how the visual design of a game affect users with different preferences.

5.3 Sustainable

Sustainability was not directly addressed by any of the participants in the study, but it is still an important concept for game design to consider, since development have impact on what has been labelled as environmental, social, and economic sustainability (Mota et al., 2015; Azapagic et al., 2016). Similar to Blevis (2007) recommendations for sustainable interaction design (SID), game design should incorporate sustainability in the values, methods, and reasoning for game development. For example, game design should be considerate of the direct and indirect impact on environmental, social, and economic sustainability. This could include what values the game promotes, how the game is to be maintained and updated after launch, what methods and techniques are used for development, and how resource intensive these are.

5.4 Limitations and Future Research

Data collection for this study consists of 15 structured e-mail interviews with higher education teachers and with that said, the study has limitations that should be addressed. Because of the relatively small sample size, findings from this study should not be understood as a generalisable understanding of higher education teachers' perceptions or preferences on the topic for the study. Authors suggest that future research should investigate design concepts for development of need-based educational games further, preferable by quantitative methods, to complement the results from this study. Focus for this paper has mainly been on investigating and discussing design concepts for educational games related to design theories. The analysis and discussion could have focused more on comparing the study's results to previous research and theories on game design. This should be addressed in future research. Lastly, according to Grounded theory the conducted phases of open and axial coding should be followed by confirmatory coding. Future re-analysis of the collected dataset should be revised and refined with descriptions of the relationships between sub- and super-categories.

6. Conclusion

This paper has investigated higher education programming teachers' perceptions about what students find easy and challenging when learning programming, in order to identify important design concepts for the development of a need-based educational game on computational thinking (CT). The paper has mainly focused on CT-concepts relating to the fundamentals of programming courses and the imperative programming paradigm. Through a discussion, where results from the study are compared to a theoretical background, the paper derives at three important design concepts for development of a need-based educational game: actions, engaging, and sustainable (Figure 3). The conclusion, and recommendations, of the paper is that a need-based educational game should consider what actions the player need to perform in order to learn, what design and visuals the player find engaging, and what impact the game has on environmental, social, and economic sustainability.

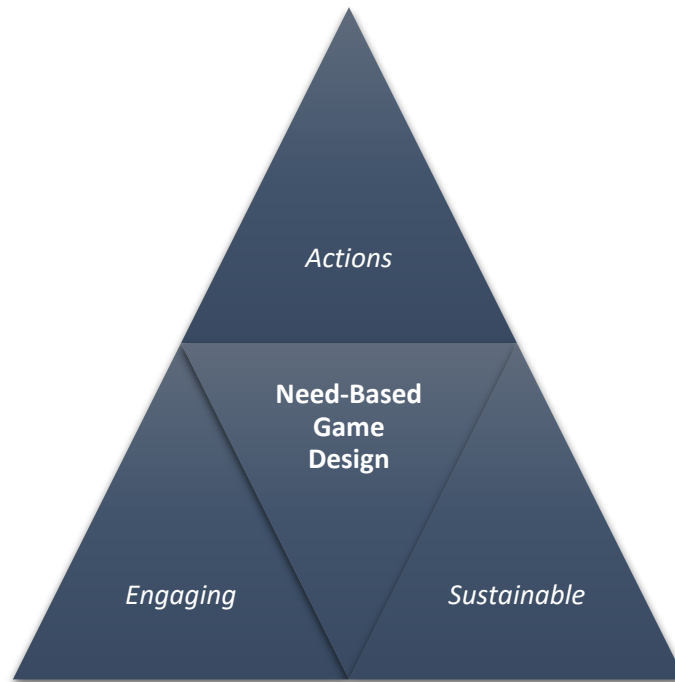


Figure 3: Conceptual design-model for a need-based game design

Acknowledgement

This study was made possible by support from the Higher Education and Digitalisation (HEaD) initiative. HEaD is a five-year strategic initiative at the Mid Sweden University, with the aim of improving the university capacity in the field of technology enhanced lifelong learning (Mozelius et al., 2022). HEaD do not only focus on the development of teachers' pedagogical digital competence, but also has the ambition to improve the general support with new tools and techniques. Authors of this paperwork in the SPEDAT project, which is one of many two-year educational development projects in the second round of the HEaD initiative.

References

- Adelson, B., 2010. The emotional design of everyday things: The 2006 Benjamin Franklin Medal in computer and cognitive science presented to Donald A. Norman. *Journal of the Franklin Institute*, 347(4), 681-687. <https://doi.org/10.1016/j.jfranklin.2008.04.004>
- Azapagic, A., Stamford, L., Youds, L., & Barteczko-Hibbert, C., 2016. Towards sustainable production and consumption: A novel decision-support framework integrating economic, environmental and social sustainability (DESIRES). *Computers & Chemical Engineering*, 91, 93-103. <https://doi.org/10.1016/j.compchemeng.2016.03.017>
- Baldassarre, B., Keskin, D., Diehl, J. C., Bocken, N., & Calabretta, G., 2020. Implementing sustainable design theory in business practice: A call to action. *Journal of cleaner production*, 273, 123113. <https://doi.org/10.1016/j.jclepro.2020.123113>
- Baskerville, R. & Pries-Heje, J., 2010. Explanatory design theory. *Business & Information Systems Engineering*, 2, pp.271-282. <https://doi.org/10.1007/s12599-010-0118-4>
- Bennedsen, J. & Caspersen, M. E., 2007. Failure rates in introductory programming. *ACM SIGCSE Bulletin*, 39(2), 32-36. <https://doi.org/10.1145/1272848.1272879>
- Blevis, E., 2007, April. Sustainable interaction design: invention & disposal, renewal & reuse. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 503-512). <https://doi.org/10.1145/1240624.1240705>
- Bower, M. & Sturman, D., 2015. What are the educational affordances of wearable technologies?. *Computers & Education*, 88, pp.343-353. <https://doi.org/10.1016/j.compedu.2015.07.013>
- Bryant, A. & Charmaz, K. eds., 2007. *The Sage handbook of grounded theory*. Sage.
- Carson, R., 1962. *Silent Spring*. Crest Book.
- Cheah, C. S., 2020. Factors contributing to the difficulties in teaching and learning of computer programming: A literature review. *Contemporary Educational Technology*, 12(2), ep272. <https://doi.org/10.30935/cedtech/8247>
- Corral, J. M. R., Balcells, A. C., Estévez, A. M., Moreno, G. J., & Ramos, M. J. F., 2014. A game-based approach to the teaching of object-oriented programming languages. *Computers & Education*, 73, 83-92. <https://doi.org/10.1016/j.compedu.2013.12.013>
- Etikan, I., Musa, S. A., & Alkassim, R. S., 2016. Comparison of convenience sampling and purposive sampling. *American journal of theoretical and applied statistics*, 5(1), 1-4. <https://doi.org/10.11648/j.ajtas.20160501.11>

- Gibson, J. J., 2015. *The Ecological Approach to Visual Perception*. Psychology Press.
- Goldkuhl, G., 2004. Design theories in information systems—a need for multi-grounding. *Journal of Information Technology Theory and Application (JITTA)*, 6(2), pp.59-72. Available at: <https://aisel.aisnet.org/jitta/vol6/iss2/7> (Accessed 5 April 2023).
- Gorson, J. & O'Rourke, E., 2020, August. Why do cs1 students think they're bad at programming?: Investigating self-efficacy and self-assessments at three universities. In *Proceedings of the 2020 ACM Conference on International Computing Education Research* (pp. 170-181). <https://doi.org/10.1145/3372782.3406273>
- Gulliksen, J., Cajander, Å., & Wiggberg, M., 2020. Digital spetskompetens – den nya renässansmänniskan: Genomlysning, definition, prognosverktyg och rekommendationer för framtida utveckling [Digital excellence – the new renaissance man: Transparency, definition, forecasting tool, and recommendations for future development]. Available at: https://tillvaxtverket.se/download/18.6855bfcf184896002ffb57/1668765822056/DigitalSpetskompetens_Definition_Gulliksenetal.pdf (Accessed 3 Maj 2023).
- Hardin, G., 1968. The tragedy of the commons: the population problem has no technical solution; it requires a fundamental extension in morality. *science*, 162(3859), 1243-1248. <https://doi.org/10.1126/science.162.3859.1243>
- Hawlitschek, A., Köppen, V., Dietrich, A., & Zug, S., 2020. Drop-out in programming courses—prediction and prevention. *Journal of Applied Research in Higher Education*, 12(1), 124-136. <https://doi.org/10.1108/JARHE-02-2019-0035>
- Ho, A. G. & Siu, K. W. M. G., 2012. Emotion design, emotional design, emotionalize design: A review on their relationships from a new perspective. *The Design Journal*, 15(1), 9-32. <https://doi.org/10.2752/175630612X13192035508462>
- Humble, N. & Mozelius, P., 2022, October. Refurbishing the educational escape room for programming: Lowering the threshold and raising the ceiling. In *Proceedings of the 16th European Conference on Games Based Learning (ECGBL)*. Academic Conferences International Limited. <https://doi.org/10.34190/ecgbl.16.1.603>
- Humble, N., Mozelius, P., & Sällvin, L., 2021, September. You can't escape learning, but maybe you can get out of the room!: Game-based learning for programming education. In *Proceedings of the 15th European Conference on Game Based Learning (ECGBL)*. (pp. 359-367). Academic Conferences International Limited. Available at: <https://urn.kb.se/resolve?urn=urn:nbn:se:miun:diva-43170> (Accessed 3 Maj 2023).
- Johnson, M., Jain, R., Brennan-Tonetta, P., Swartz, E., Silver, D., Paolini, J., Mamonov, S., & Hill, C., 2021. Impact of big data and artificial intelligence on industry: developing a workforce roadmap for a data driven economy. *Global Journal of Flexible Systems Management*, 22(3), 197-217. <https://doi.org/10.1007/s40171-021-00272-y>
- Johannesson, P. & Perjons, E., 2014. *An introduction to design science*. Springer.
- Khandkar, S.H., 2009. Open coding. University of Calgary. Available at: <http://pages.cpsc.ucalgary.ca/~saul/wiki/uploads/CPSC681/opencoding.pdf> (Accessed 3 Maj 2023).
- Klopper, E., Osterweil, S., & Salen, K., 2009. Moving learning games forward: Obstacles, opportunities, and openness. The Education Arcade, MIT. Available at: https://education.mit.edu/wp-content/uploads/2018/10/MovingLearningGamesForward_EdArcade.pdf (Accessed 3 Maj 2023).
- March, S. T. & Smith, G. F., 1995. Design and natural science research on information technology. *Decision support systems*, 15(4), pp.251-266. [https://doi.org/10.1016/0167-9236\(94\)00041-2](https://doi.org/10.1016/0167-9236(94)00041-2)
- Mota, B., Gomes, M. I., Carvalho, A., & Barbosa-Povoa, A. P., 2015. Towards supply chain sustainability: economic, environmental and social design and planning. *Journal of cleaner production*, 105, 14-27. <https://doi.org/10.1016/j.jclepro.2014.07.052>
- Mozelius, P., Sällvin, L., & Humble, N., 2023, June. An Educational Escape Room for Computational Thinking - Defining the Requirements. *International Conference on Education and New Developments (END 2023)*. The World Institute for Advanced Research and Science (WIARS).
- Mozelius, P., Bader, S., Jaldemark, J., Urbansson, P., & Engström, A., 2022. Educational Development-Challenges, Opportunities, Tools and Techniques. In *Proceedings of the 21st European Conference on e-Learning (ECEL)*. Academic Conferences International Limited. <https://doi.org/10.34190/ecel.21.1.626>
- Norman, D., 2013. *The design of everyday things: Revised and expanded edition*. The MIT Press.
- Norman, D. A., 2008. The way I see IT signifiers, not affordances. *interactions*, 15(6), 18-19. <http://doi.acm.org/10.1145/1409040.1409044>
- Norman, D. A., 2005. *Emotional design: Why we love (or hate) everyday things*. Basic Books.
- Norman, D. A., 1999. Affordance, conventions, and design. *interactions*, 6(3), 38-43. <https://doi.org/10.1145/301153.301168>
- Papadakis, S. & Kalogiannakis, M., 2019. Evaluating the effectiveness of a game-based learning approach in modifying students' behavioural outcomes and competence, in an introductory programming course. A case study in Greece. *International Journal of Teaching and Case Studies*, 10(3), pp.235-250. <https://doi.org/10.1504/IJTCS.2019.102760>
- Pengnate, S. F. & Sarathy, R., 2017. An experimental investigation of the influence of website emotional design features on trust in unfamiliar online vendors. *Computers in Human Behavior*, 67, 49-60. <https://doi.org/10.1016/j.chb.2016.10.018>
- Preece, J., Sharp, H., & Rogers, Y., 2015. *Interaction design: beyond human-computer interaction*. John Wiley & Sons.
- Rahman, T.F.B.A., Anuar, N., & Said, R.F.M., 2018, January. How the nature of programming and learning materials affects novice learner's motivation and programming ability. In *Proceedings of the 6th International Conference on Information and Education Technology* (pp. 124-128). <https://doi.org/10.1145/3178158.3178184>

- Shuhidan, S., Hamilton, M., & D'souza, D., 2009, January. A taxonomic study of novice programming summative assessment. In *Proceedings of the Eleventh Australasian Conference on Computing Education*, Volume 95 (pp. 147-156). <https://dl.acm.org/doi/10.5555/1862712.1862734>
- Settle, A., Vihavainen, A., & Sorva, J., 2014, June. Three views on motivation and programming. In *Proceedings of the 2014 conference on Innovation & technology in computer science education* (pp. 321-322). <https://doi.org/10.1145/2591708.2591709>
- Smit, S., Tacke, T., Lund, S., Manyika, J., & Thiel, L., 2020. The future of work in Europe: Automation, workforce transitions, and the shifting geography of employment. McKinsey Global Institute. Available at: <https://www.mckinsey.com/featured-insights/future-of-work/the-future-of-work-in-europe> (Accessed 3 Maj 2023).
- Stendal, K., Thapa, D., & Lanamäki, A., 2016, January. Analyzing the concept of affordances in information systems. In *2016 49th Hawaii international conference on system sciences (HICSS)*, (pp. 5270-5277). IEEE. <https://doi.org/10.1109/HICSS.2016.651>
- Vollstedt, M. & Rezat, S., 2019. An introduction to grounded theory with a special focus on axial coding and the coding paradigm. In: Kaiser, G. & Presmeg, N. (eds) *Compendium for early career researchers in mathematics education*. ICME-13 Monographs. Springer. https://doi.org/10.1007/978-3-030-15636-7_4
- Walls, J. G., Widmeyer, G. R., & El Sawy, O. A., 1992. Building an information system design theory for vigilant EIS. *Information systems research*, 3(1), pp.36-59. <https://doi.org/10.1287/isre.3.1.36>
- Wolz, S., Bergande, B., & Brune, P., 2022. Influence factors on students motivation in introductory programming lectures of computer science non-majors. *Cogent Education*, 9(1), 2054914. <https://doi.org/10.1080/2331186X.2022.2054914>