Helsinki Studies in Education, number 113

Mika Letonsaari

Nonlinear Storytelling Approach to Developing Computational Thinking Skills

Doctoral dissertation, to be presented for public discussion with the permission of the Faculty of Educational Sciences of the University of Helsinki via online stream on the 5th of June 2021 at 12 o'clock.

Helsinki 2021

Pre-examiners
Professor Markku Tukiainen, University of Eastern Finland
Dr. Jari Laru, University of Oulu

Custos
Professor Kristiina Kumpulainen, University of Helsinki

Supervisors
Professor Kristiina Kumpulainen, University of Helsinki
Adjunct Professor Sara Sintonen, University of Helsinki

Opponent
Professor Piet Kommers, University of Twente

Cover
Maisa Letonsaari

Yliopistopaino Unigrafia, Helsinki

ISBN 978-951-51-7294-5 (nid.)
ISBN 978-951-51-7295-2 (pdf)

University of Helsinki, Faculty of Educational Sciences
Helsinki Studies in Education, number 113

Mika Letonsaari

Nonlinear Storytelling Approach to Developing Computational Thinking Skills

Abstract

Computational thinking skills are extensively required in our technologically developing society. Current methods for developing computational thinking skills usually have a technical and programming-centric approach and not suitable for all people. In this research, the use of nonlinear storytelling as an educational method was examined. The specific interest was to analyze its relationship with the concept of computational thinking and to investigate if nonlinear storytelling can be used as a low-threshold method for teaching fundamental computational thinking skills. Fundamental computational thinking skills are mostly concerned with primary education but they are relevant also in upper levels of education and adult education especially for students with learning disabilities and people with notable gaps in basic information technology skills.

This research situates itself in computer science education which is a distinct field of pedagogy characterized by the use of concepts and ideas from computer science, engineering, and mathematics. It consists of four independent studies. Study I investigates how nonlinear storytelling can be integrated into an adult education course for developing basic information technology skills. Special attention is given to understanding the role of storytelling in the process. Students with low information technology skills (N = 14) participated in a computer game development project by writing nonlinear stories. The result of this study was a method that integrates nonlinear storytelling into educational game development.

Study II studied the relationship between nonlinear stories and computational thinking by examining how typical computer programs are implemented using nonlinear stories. The study shows that nonlinear stories are best suited for implementing finite state machine programs and programs that include interaction. The natural character of applicability indicates that nonlinear storytelling can improve students' readiness for learning programming skills.

In study III, experiences and observations made at the end of the aforementioned adult education course for developing basic information technology skills are reported. The technical quality of the stories collected (N = 14) was investigated and common challenges in the storytelling process such as understanding hyperlinking and its purpose in gamification were identified. In this study, a practical classification for storytelling software and metrics for analyzing stories were developed.

Finally, study IV focused on investigating whether the concept of computational thinking allows broader interpretations compared to how it is traditionally used. In the traditional programming-centric approach, learners practice stepwise algorithmic problem solving using simplified computer programming using tools such as visual programming or card games. Starting from simplified programing principles, more complex programming tasks are employed. This approach is not naturally applicable to the storytelling approach in which skills for managing complexity and abstract thinking are practiced by developing more sophisticated stories without direct computer programming-related learning goals. In this theoretical study, the broader context for computational thinking was explored using the Extended Mind thesis by Clark and Chalmers.

This research indicates that nonlinear storytelling is a viable method for training fundamental computational thinking skills. Its pedagogical content does not completely correspond to that of traditional programming-centric methods. Yet, as technology and its use get more common in near future, information technology-aided problem-solving will probably be transformed into more human-centric and diverse forms instead of the current computer programming paradigms. Therefore, it is not imperative to use the same assessment tools and interpretations that are traditionally used in evaluating computational thinking.

Keywords: nonlinearity, storytelling, computational thinking, interactive fiction

Helsingin yliopisto, Kasvatustieteellinen tiedekunta
Kasvatustieteellisiä tutkimuksia, numero 113

Mika Letonsaari

Epälineaarinen tarinankerronta menetelmä algoritmisen ajattelun taitojen kehittämiseen

Tiivistelmä

Teknistyvä yhteiskunta vaatii ihmisiltä uudenlaisia algoritmisen ajattelun taitoja. Nykyiset menetelmät algoritmisen ajattelun opetuksessa ovat usein teknisiä ja ohjelmointikeskeisiä eivätkä ne sovi kaikille kohderyhmille. Tässä työssä selvitettiin epälineaaristen tarinoiden käyttöä opetuksessa. Erityinen kiinnostuksen kohde oli se, mikä on epälineaaristen tarinoiden suhde algoritmiseen ajatteluun ja voiko epälineaarisia tarinoita käyttää matalan kynnyksen menetelmänä algoritmisen ajattelun harjoittamiseen. Opetuksessa algoritmisen ajattelun perustaitoja opetellaan erityisesti peruskoulussa, mutta niitä voidaan harjoitella myös ylemmillä koulutusasteilla sekä aikuiskoulutuksessa, erityisesti niiden opiskelijoiden kanssa, joilla on oppimisvaikeuksia tai huomattavia aukkoja tietotekniikkaosaamisessa.

Tämä tutkimus sijoittuu tietotekniikan opetuksen alalle. Tietotekniikan opetus on pedagogiikkaa, jossa sovelletaan tietotekniikan, tekniikan sekä matematiikan käsitteitä ja ideoita. Työ koostuu neljästä osatutkimuksesta. Osatutkimuksessa I tutkittiin, miten epälineaarinen tarinankerronta voidaan ottaa osaksi tietotekniikkavalmiuksia kehittävää aikuiskoulutusta. Koulutukseen integroitiin osaksi tietokonepelin kehitys, johon opiskelijat (N = 14) osallistuivat kirjoittamalla epälineaarisista tarinoista koostuvia episodeja. Tutkimuksen tuloksena syntyi menetelmä, joka yhdistää epälineaarisen tarinankerronnan pelinkehitykseen.

Osatutkimuksessa II tutkittiin epälineaaristen tarinoiden suhdetta algoritmiseen ajatteluun selvittämällä tyypillisten tietokoneohjelmien toteutuksia. Toteutuksista kävi ilmi, että epälineaariset tarinat sopivat erityisesti äärellisillä automaateilla esitettävissä olevien ohjelmien sekä interaktiivisten ohjelmien toteuttamiseen. Tarinallisten toteutusten luontevuus osoitti, että tarinoiden avulla voidaan harjoittaa opiskelijoiden ohjelmointivalmiuksia.

Osatutkimuksessa III raportoitiin edellä mainitun tietotekniikkavalmiuksia kehittävän aikuiskoulutuksen aikana tehtyjä havaintoja ja saatuja kokemuksia hankkeen loputtua. Erityisesti selvitettiin hankkeessa kerättyjen tarinoiden (N = 14) teknistä laatua sekä yleisimpiä ongelmia hyperlinkkien toiminnan sekä pelillisen merkityksen ymmärtäminen kanssa. Osatutkimuksen tuloksena kehitettiin käytettävissä olevien ohjelmistojen luokittelu sekä tarinoiden metriikkaa.

Osatutkimuksessa IV selvitettiin, miten algoritmisen ajattelun käsitettä voidaan tulkita sen perinteistä tulkintaa laajemmin. Perinteinen lähestymistapa algoritmisen ajattelun opettamiseen on yksinkertaistetun ohjelmoinnin opettelu. Siinä opiskelijat harjoittelevat ongelmien askelittaista, algoritmista ratkaisemista helposti lähestyttävillä menetelmillä, esimerkiksi graafisella ohjelmoinnilla tai pahvikortteja käyttäen Tämä yksinkertaisesta ohjelmoinnista kohti perinteistä ohjelmointia etenevä lähestymistapa ei luontaisesti sovellu epälineaaristen tarinoiden kanssa käytettäväksi. Tarinankerronnassa kompleksisuuden hallintataitoja ja abstraktin ajattelun keinoja lisätään tarinaa kehittämällä ilman suoria ohjelmointiosaamisen tavoitteita. Tässä teoreettisessa tutkimuksessa selvitettiin miten algoritmisen ajattelun käsite voidaan ymmärtää sen perinteistä käyttöä laajemmassa kontekstissa. Problematiikkaa lähestyttiin Clarkin ja Chalmersin laajennetun mielen hypoteesia käyttäen.

Tutkimuksen perusteella epälineaarista tarinankerrontaa voidaan soveltaa algoritmisen ajattelun harjoittamiseen. Tarinankerronta ei sisällöllisesti täysin vastaa perinteisiä ohjelmointikeskeisiä menetelmiä. Tietotekniikan kehittyessä ongelmanratkaisu tulee todennäköisesti kehittymään ihmiskeskeisemmäksi ja vähemmän tekniseksi. Siksi menetelmää arvioitaessa ei ole välttämätöntä sitoutua tiukasti siihen miten algoritmisia ajattelutaitoja on perinteisesti totuttu arvioimaan ja tulkitsemaan.

Avainsanat: epälineaarisuus, tarinankerronta, algoritminen ajattelu, interaktiivinen fiktio

Acknowledgements

With great gratitude, I thank my supervisors, Professor Kristiina Kumpulainen and Adjunct Professor Sara Sintonen for their continuous support and guidance during my doctoral research.

I express my sincere thanks to Professor Piet Kommers for accepting the invitation to be an official opponent. I thank the thesis pre-examiners Professor Markku Tukiainen and Dr. Jari Laru for their valuable comments.

Special thanks to my colleague Jukka Selin who evoked the idea to pursue the doctoral degree, and research director Noora Talsi who supported my research.

I thank all my colleagues and friends who have helped me with my research. I also thank all the researchers and authors whose work I have had a great pleasure to study. Research on computer science education is such a treasure trove of brilliant and beautiful ideas.

Mikkeli, 10.5.2021
Mika Letonsaari

Contents

List of figures

List of tables

List of original articles

This doctoral dissertation is based on the following four articles, which are referred to in the text by their Roman numerals (Studies I-IV):

Study I	Letonsaari, M., Selin, J. and Lampi, M., 2017, April. Co-creative serious games design process using nonlinear storyline editing. In International Conference on Computer Supported Education (Vol. 2, pp. 582-588). SciTePress.
Study II	Letonsaari, M. and Selin, J., 2017. Modeling computational algorithms using nonlinear storytelling methods of computer game design. Procedia computer science, 119, pp.131-138.
Study III	Letonsaari, M., Karjalainen, L. and Selin, J. 2019. Nonlinear Storytelling Method and Tools for Low-Threshold Game Development. Seminar.net (Vol. 15, No. 1, 2019)
Study IV	Letonsaari, M., 2018, June. Extended Cognition Hypothesis Applied to Computational Thinking in Computer Science Education. In International Conference on Computational Science (pp. 304-317). Springer, Cham.

1 Introduction

Information technology has changed our world greatly. Only fifteen years ago, we did not have modern smartphones. The original iPhone was released in 2007 and the first Android phone in 2008. Mobile Internet has enabled the wide adoption of social media such as Facebook (created in 2004), YouTube (2005), and Twitter (2006), as well as the many everyday mobile services we use today.

For people to be able to participate in this modern information society and be functional members of it, continuous learning is required. For educators, this poses a challenge in how to develop efficient pedagogies and provide relevant learning opportunities.

The research discussed in this thesis began during an information and communication technology (ICT) education project for adults[1]. Many adults still lack the basic ICT skills needed in working life as well as in many everyday tasks, such as mobile banking and utilizing state-provided online and mobile services for citizens. The project, coordinated by the Worker's Educational Association of Finland, identified students' learning difficulties and developed new methods for learning and updating their ICT skills. The aim was to promote active ICT use and provide students with tools for continuous learning.

As part of the project, an educational computer game was developed. The game was developed mainly by teachers and information technology students, but those students who were still learning basic ICT skills were also able to participate. A storytelling method requiring no special information technology skills was developed, in which computer game plots such as dialogs could be written. Students were asked to write stories with simple gamification elements using their own interests and knowledge. The stories were then semi-automatically integrated into the game.

In the process of developing the game, it was noticed that the storytelling techniques needed for gamified stories were fundamentally different from traditional storytelling. In gamified stories, the player can make decisions and choose how the game plot advances. There must be several storylines that implement all the scenarios, taking account of the actions of the player. The decision where the storyline is chosen is called branching. Together, branching storylines form a complex story structure. This type of story structure is defined as a nonlinear story. This is unlike traditional linear stories without branching, where the user or reader has no choice over the storyline.

Traditional linear storytelling is a well-established educational method. It is considered an essential part of the participatory culture of the 21st century (Jenkins 2006). Multimedia storytelling and digital storytelling have extended storytelling methods in many ways, for example learning advanced digital storytelling techniques and co-creation (Alismail 2015). But almost all storytelling used in education today is linear. Even most multimedia and digital storytelling still utilizes only a single plot without the possibility of branching.

[1] "Competent Employees for Digitalizing Work" project, more information available (in Finnish language) at https://www.eura2014.fi/rrtiepa/projekti.php?projektikoodi=S20547

Therefore, nonlinearity is a relatively poorly known feature in educational storytelling, and it is rarely introduced in the theory of storytelling. When it is used, it is sometimes called nonlinear storytelling, sometimes interactive storytelling, and often not recognized as a special feature at all. This raised a need to define this quality and study its properties in greater detail in our project.

Another realization in the project was that creating nonlinear stories involved challenges that do not exist with traditional linear stories. Most people are used to writing traditional linear stories, as it is taught at school. But nonlinear stories and storyline branching are mostly unfamiliar features, especially to those who have little or no experience in computer games or mobile gaming. Even those who have played games have rarely created games by themselves.

What then is the challenge in writing hypertext stories? Reading and comprehending hypertexts has been studied comprehensively. In the early days of hypertext, it was an important research question whether hypertext learning materials are beneficial or detrimental for the learning process and understanding. Hypertext can make learning a more active and meaningful activity, but it involves cognitively challenging navigation and decision-making (Salmerón et al. 2018; DeStefano and LeFevre 2007).[2]

Creating and authoring hypertexts is a much less heavily researched topic. It is most commonly considered a task for advanced students and professionals, and not for everyone. The most common tools for writing nonlinear content are Wiki systems. Research has found benefits in collaboration (Notari 2006) and knowledge management skills (Biasutti and Heba 2012) but also challenges in using Wikis efficiently (Cole 2009).

In our project, storytelling was part of game development. This led to the author's realization that there is considerable similarity between nonlinear storytelling and computer programming. Low-level programming languages are characterized by step-by-step execution and branching of the control flow, very similar to the nonlinear storyline and its branching. Writing nonlinear stories seems to require similar abstract thinking and skills to manage nonlinear complexity as is required in computer programming.

In information technology education, computational thinking skills are transferable skills that are required for using computers in practical problem-solving and automation tasks (Wing 2006, Grover and Pea 2013). These skills are traditionally practiced with easy-to-use programming languages such as block-based Scratch and Alice. Typically, simple problem-solving tasks such as robot routing are studied, while engaging and entertaining programming tasks, such as animations and games, are implemented.

The similarity between the structures of nonlinear stories and computer programs

[2] To the author's understanding, this question did not have a clear answer, and the pros and cons must be balanced case-by-case. Influential features are, for example, target audience (age, experience in using hypertexts), the general quality of learning material (authoring hypertext materials requires special knowledge and planning), structure of the hypertext (see for example, DeStefano and LeFevre 2007), and educational context (learning certain content matter or developing a deeper understanding on the subject).

raised the question of whether nonlinear stories could be used to develop computational thinking skills. In principle, storytelling could provide an alternative natural-language-based approach for learning computational thinking. Understanding stories and telling stories are very cognitively natural activities. Therefore, this approach might offer some advantages compared to more technical methods.

To the author's knowledge, there is no previous research on this subject on how the specific property of nonlinearity in storytelling is connected to developing computational thinking. This thesis, therefore, builds fundamental knowledge to support the use of nonlinear storytelling in developing computational thinking.

2 Background

In this chapter, some background for the research is presented. First, I introduce and define storytelling and its property of nonlinearity in section 2.1. A historic overview of nonlinearity in stories and storytelling is presented in section 2.2 to provide a broader context. The special interest in this research is in educational applications. Therefore, a short literature review is presented in section 2.3 to showcase how nonlinear storytelling has previously been used in education. The main subject of the research is the relationship between nonlinear stories and computational thinking skills. Computational thinking is introduced in section 2.4.

2.1 Nonlinear storytelling

The human understanding of the world is deeply connected to stories. Before written text, verbal stories were how people communicated with others and learned from them. Human cognition has evolutionarily adapted to the use of stories. Our brains have a special type of memory, called episodic memory, which is specialized in remembering sequences of events (Tulving 2002). Even modern humans conceptualize and understand the world through stories.

Stories and storytelling are also widely used in education. In educational storytelling, the learner creates or tells a story that is related to the subject matter. A learner can use their personal point of view, previous knowledge of the subject, and their interests to create their personal story. In this way, storytelling is a highly constructivist method of learning. Storytelling can be associated with many positive cognitive, social, and emotional benefits, including improvement in language skills, concentration, imagination, comprehension, and critical thinking (Boltman 2001, pp. 17-18).

Linear stories express a single story, and the story is always the same. For example, the book is read from the first page to the last page and a movie is seen from the first frame to the last frame in an ordered sequence. The story advances directly from the beginning of the story to the end, and it is the same story every time[3].

Nonlinear stories are stories where the plot of the story can be branched and linked. For example, Liu et al. (2010, p. 4787) describe linear stories

> *"Linear stories contain exactly one begin, one middle and one end. All children collaborate on a shared story in the form of relay and no branches can be developed."*

and nonlinear stories

[3] Ignoring the obvious interpretational differences raising from readers' subjective views and experiences.

> *"By contrast, nonlinear stories enable children to link and orchestrate different ideas. Children can thus integrate other's episodes to develop different branches of stories."*

The Fundamentals of Game Design (Adams 2009) describes linear stories (p. 642) as

> *"Stories whose plots do not change in response to player actions."*

and nonlinear stories (p. 644) as

> *"Stories whose plot can change in response to dramatic actions on the part of the player."*

Nonlinearity is also the term used by software such as Alice[4] or Twine[5] that allows the writing of game plots with a branching storyline.

An example of a simple nonlinear story is a story with two endings. The structure of such a story is given in Figure 1. The choice of ending can be game-like. For example, at a crucial moment in the story, the protagonist may need to make an important decision on how they will act. The reader can decide what the protagonist chooses to do, and thus select the corresponding ending.

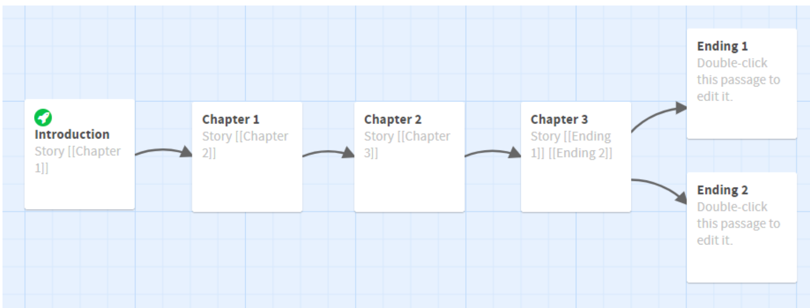

Figure 1. An example of a simple nonlinear story with two possible endings in Twine story-view. The story starts with a passage marked with a green sign. Arrows present possible paths in the story.

A well-known example of a story with two endings is the novel *Great Expectations* (1861) by Charles Dickens. The publisher did not like the original ending of the novel, and Dickens therefore wrote an alternative, happier ending for the story. Some editions of the book have both endings so that the reader can choose which one to read.

Nonlinear stories can have very complex plots. In Figure 2, the structures of two stories are presented. In the first story, the plot has many alternatives, but the story always returns to the main plot. In this way, the reader can explore the parts of the world that they find interesting and customize the reading experience while the author retains control over the overall story.

---

[4] The educational storytelling software Alice by Carnegie Mellon University, https://www.alice.org/resources/lessons/design-interactive-narrative/

[5] Open-source tool for telling interactive nonlinear stories by Chris Klimas, http://twinery.org/

The second story has very high branching. Right from the beginning, the story branches into three completely distinct paths. For example, the same story might be told from three separate perspectives, from which the reader can select one.

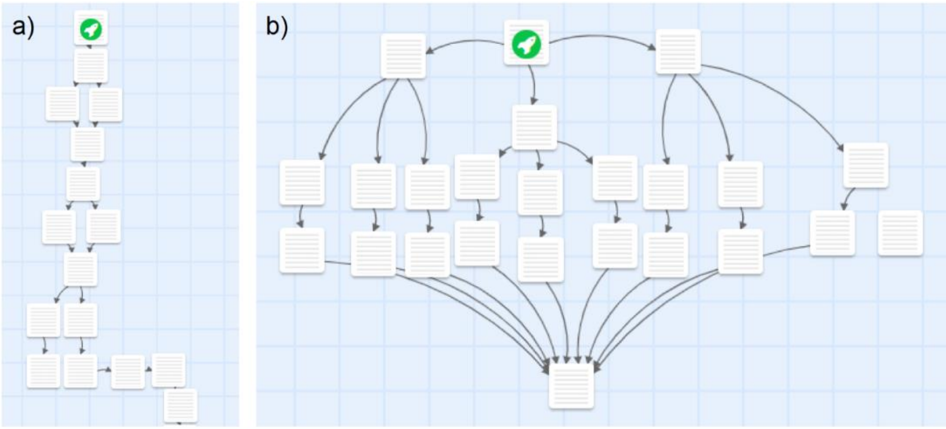

Figure 2. Nonlinear story with (a) a low-branching structure, and (b) a high-branching structure in Twine story-view. (Letonsaari and Selin 2018)

In addition to low-branching and high-branching story structures, all storylines can have a common endpoint, or there can be any number of different endings. The structure can include even loops where the story returns to its earlier state. The structure can be formally described by using graph theory and network topology. (Lin 2003; Bromme and Stahl 2005)

Nonlinearity is often associated with interactive stories and games, since the reader is usually given a chance to decide how the story advances. But nonlinearity can also be implemented without interaction, for example by chance or by informed choice unrelated to the story. Sometimes the terms nonlinear and interactive are used as synonyms, but sometimes their distinct function needs to be acknowledged. The concepts are summarized in Table 1.

Table 1. Linearity and interactivity in storytelling.

	Noninteractive	**Interactive**
Linear	traditional stories with a single narrative and the reader can make no decisions	the reader can, for example, customize the story by providing personal preferences such as their name, but these options do not change the narrative
Nonlinear	alternative narratives which are presented for example by random chance	the story has alternative narratives and the reader's decisions affect how they are presented

The research literature on interactive and nonlinear storytelling sometimes overlaps, and the terminology might depend on the researcher's background or approach to the subject.

Nonlinearity usually emphasizes the structure of the story and the branching function of the storyline. It is concerned with creating different scenarios and mapping possibilities. Interactive storytelling emphasizes the interaction provided to the user. It generally explores different techniques and technologies that can be used to create interaction. Interactive storytelling is interested in how to engage the user and how to provide immersion for the user by enabling them to play an active role.

Linearity and nonlinearity are general terms originating from mathematics (referring to similarities to the properties of the line as a mathematical object). These terms are used to describe the sequential nature or one-dimensionality of many different phenomena. It is therefore important to note that there is some common terminology that should not be confused with linearity and nonlinearity as defined here.

Nonlinear video editing is a term sometimes used in digital storytelling. It is used to describe video editing methods where the video source is not altered while editing (Media College 2017). It is a technical term and not related to storytelling.

The term nonlinear narrative in literature and film studies is used to describe events that are not presented in chronological order, and events that are not causally related (Morgan 2017). The nonlinear narrative still usually describes a single story without branches and linking, and is therefore not a nonlinear story as defined here.

2.2 History of nonlinear storytelling

While stories with the simple linear structure are overwhelmingly more common in popular culture, there are many examples of nonlinear stories in storytelling culture. Aarseth (1997, p. 2) has explored the history of cybertext and hypertext, and defines ergodic literature (from the Greek words *ergon* and *hodos*, "work" and "path") thus:

> *“In ergodic literature, nontrivial effort is required to allow the reader to traverse the text. If ergodic literature is to make sense as a concept, there must also be nonergodic literature, where the effort to traverse the text is trivial, with no extranoematic responsibilities placed on the reader except (for example) eye movement and the periodic or arbitrary turning of pages.”*

Aarseth mentions an early example of ancient Egyptian wall inscriptions in temples. These texts are connected two-dimensionally (on one wall) or three-dimensionally (extending from room to room). According to Aarseth, *“this layout allowed a nonlinear arrangement of the religious text in accordance with the symbolic architectural layout of the temple.”* (Aarseth 1997, p. 6; Gundlach 1985)

Similarly, the creation and the interpretation of visual arts can generally be considered fundamentally nonlinear processes (Zausner 2007). It is not predefined from what part of a painting or a sculpture one is expected to start examining the work. Viewers can choose what parts they want to look at in detail, and in which order. Each interpretation is a unique story.

As an early literary example, Aarseth mentions the *I Ching* from the Chinese Western Chou dynasty (1122-770 BCE). It was used as a divination manual and it was interpreted by tossing and counting stalks or coins (Ritsema and Karcher 1994). It therefore has a nonlinear structure, but it is also an example of non-interactive nonlinear work as random chance is used to choose the interpretation.

Theatre also has a long history of nonlinear storytelling. The Atellan Farce from the 4th century BCE is an early case of improvisational theatre (Petrides 2014). From similar initial conditions, improvisation can tell a different story depending on the actors, the response of the audience, and even the pure randomness of events.

The specific use of nonlinearity and interactivity in popular culture is still quite recent. According to Demian's Gamebook Web Page, which keeps an archive of interactive books, the first authors were Doris Webster and Mary Alden Hopkins with their title *“Consider the Consequences!”* in 1930 (Katz n.d.). The book consists of a group of romantic stories and the reader can choose the ending for each intertwined story.

Four years later Ayn Rand’s theater play *Night of January 16th* (1934) was introduced. In this interactive play, audience members are chosen to play jury members. Those audience members have a chance to change the outcome of the story.

Finally, the most well-known example of nonlinear literature is the *Choose Your Own Adventure* book series by Edward Packard. These books are children’s fictional gamebooks written from a second-person point of view. After every few pages, the reader is asked to decide what the protagonist chooses to do next in the story. Depending on the decision, the reader is advised to jump to the corresponding page number and continue the story from there.

Packard's first interactive book, *Sugarcane Island*, was published in 1976. The *Choose Your Own Adventure* series was published from 1979 to 1998, with 184 titles and over 250 million copies sold (Lodge 2007). The format of the series is often called "gamebook", and it has been used by many other authors inspired by the success of the *Choose Your Own Adventure* series. The book series was also adapted by McGraw-Hill Education for a *Choose Your Own Adventure Graded Reader Series* (Katz 2013).

This was all revolutionized by information technology and hypertext in the latter part of the 20th century. Hypertext, the standard content of the World Wide Web, is described by the World Wide Web Consortium: *"Hypertext is text which is not constrained to be linear. Hypertext is text which contains links to other texts."* Hypermedia, another key concept of Internet content, is defined as *"a term used for hypertext which is not constrained to be text: it can include graphics, video and sound"*. (W3.org n.d.)

Internet consisting of hypertext and hypermedia is therefore fundamentally nonlinear media. Also, computer games and mobile games with nonlinear structures have become mainstream entertainment. Nonlinearity in storytelling is part of mainstream culture nowadays. People are familiar with nonlinear content and they want to use it.

2.3 Nonlinear Storytelling in Education

The nonlinear structure of information and its relation to learning has intrigued many researchers. Already Bush and Think (1945) described ideas about archives that work by associate links, inspired by how the human mind works. Ford and Chen (2000) have reviewed the literature on how hypertext and hypermedia will advance learning and describe their main findings:

a) The types of knowledge representation hypermedia afford is arguably closer than text-based representation to human associative and schema-based memory structures (Jonassen 1988, 1991; Marchionini 1988);
b) it has been argued that, in supporting such knowledge structures, hypermedia is well suited to facilitating learning processes, particularly as proposed by constructivist learning models (Cobb et al. 1992; Jonassen 1991; Nunes and Fowell 1996); and
c) the transformation and reconstruction of information that characterizes deep learning is facilitated by hypermedia-based information presentation (Jacobson and Spiro 1995)

More practically oriented, Dillon and Gabbard (1998) have listed possible benefits of intrinsically nonlinear hypermedia in education:

a) hypermedia enables nonlinear access to vast amounts of information (Nielsen 1995);
b) users can explore information in-depth on-demand (Collier 1987);
c) interaction with the instructional material can be self-paced (Barrett 1988);
d) hypermedia is attention-capturing or engaging to use (Jonassen 1989); and
e) hypermedia represents a natural form of representation with respect to the working of the human mind (Delany and Gilbert 1991)

Here we are interested specifically in nonlinear storytelling applications. In the following, a short literature survey on nonlinear storytelling in education is presented to provide an overall picture of the field. Google Scholar found 15 articles with *"nonlinear storytelling"* in the title, and 14 articles with the alternative spelling *"non-linear storytelling"*. Excluding three publications from the research presented in this thesis, the total number of results is 26.

This is a relatively small number compared to, for example, *"digital storytelling",* which produces 2740 results or *"storytelling",* with 16600 results[6]. A list of all nonlinear storytelling articles (both spellings) is included in Appendix A.

Only six of these articles present studies relevant to education. These articles are presented in the following. Other cases of educational research involving nonlinear storytelling most likely exist, but these studies give a general overview of the existing research.

2.3.1 Tawfik et al. (2018). Case-based learning Game

Tawfik et al. (2018) introduce a game-like, case-based learning environment in the form of a narrative text adventure game. In this learning environment, learners are presented with the challenge of solving a hiring problem in a sales management scenario set in a health equipment sales company. The game in this study was created by the researchers, and the participants only played the game. Therefore, on their part, this was only using nonlinear learning material, and they did not create stories themselves.

The focus of the study was on the usability of the learning environment. But the authors also indicated five themes that underscored the differences between the nonlinear format they used and more traditional narrative formats: (1) decision-making, (2) perspective taking, (3) problem representation, (4) case retrieval, and (5) case retention.

They report that participants were positive in that the nonlinear learning environment afforded perspective-taking and allowed them to take ownership of the material by making decisions. They also reported that decision-making was beneficial because it afforded the exploration of alternative solution paths. An interesting observation was that participants

[6] Search results including only articles, excluding patents and citations. Search performed Nov. 9th 2019.

were not satisfied with just playing through the game, but wanted to know how their decisions affected the story and what stories were possible in the game.

2.3.2 Maleki and Sajjadi (2012). Nonlinear language teaching in medicine

Maleki and Sajjadi (2012) studied English language teaching in Medicine. They compared traditional language teaching to nonlinear language teaching, which includes storytelling and small-group, problem-based learning. The nonlinear storytelling approach compared very favorably to traditional language teaching. In the nonlinear storytelling group, students were 52.7% more motivated, 65.2% more interested in-class activities and their class participation was 73.9% higher. Learning results were 47.4% better.

According to them, storytelling in language teaching is nonlinear by default. They do not provide a formal analysis of the method but discuss some possible factors. They refer to Fraser and Greenhalgh's (2001) idea that students must be educated not only for competence, but for capability (meaning the ability to adapt to change, generate new knowledge, and continually improve performance). Researchers suggest that this is achieved by nonlinear methods in this research. They also suggest that the zone of complexity (Plsek and Greenhalgh 2001) created by nonlinear storytelling may be a significant factor in the enhancement of learning.

Maleki and Sajjadi also list some other findings that correlate to these results supporting nonlinear storytelling:

- people access, express, and retain information and knowledge through storytelling (Caine et al. 2005);
- storytelling makes learning more meaningful, challenging, and stimulating (McDrury and Alterio 2003); and
- storytelling is enjoyable, entertaining, and interesting, which helps improve the overall learning, listening, and interacting skills of students (Mello 2001).

2.3.3 Liu et al. (2010). Children's collaborative storytelling platform

Liu et al. (2010) present a storytelling platform that allows linear and nonlinear storytelling. Stories are created using graphic sketchpads, animation tools, and commenting. The story structure is composed using a linear or nonlinear story editor. Story editors are presented in Figure 3.

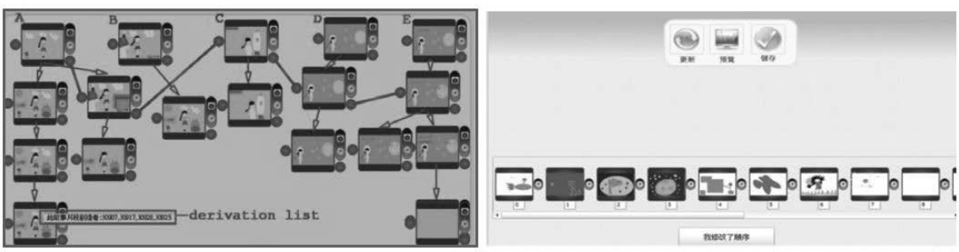

Figure 3. Nonlinear (left) and linear (right) story editor used by Liu et al. (2010).

Linear stories include only one beginning, one sequence of events, and one end. The story was created by children in a relay style. Liu et al. (2010, p. 4788) refer to Rettberg (2005) and describe relay-style linear story creation: *"Children [thus] deeply rely on evaluating the relationship, continuality and coherence of story path before sequentially participating in building up the story"*. In nonlinear stories, children can develop branches to the storyline, orchestrate different ideas, and integrate others' episodes.

Liu et al. hypothesize that this might change their perception and interaction in story creation as well as affecting their motivation and success in story creation. To study this, they observed story creation by 57 elementary school children, mean age 9 years, in Taiwan. The study is based on a four-step approach to facilitating children's collaborative storytelling, proposed by Shneiderman et al. (2002) and Robin (2006).

Results were measured in four categories: derivation, remix, ownership, and positive independence. The nonlinear group performed better in all categories. According to Liu et al., children in the nonlinear group were not afraid of spoiling others' work, and were therefore more motivated. The nonlinear interface better facilitated the children's ability to manage and remix stories. Nonlinearity also allowed children to develop their own stories and increased their feeling of ownership.

In this study, practices associated with linear and nonlinear applications were highly different by nature. Therefore a direct comparison may not be valid in all aspects. Their conclusion to the study is: *"Researchers and educators may find it of value to evaluate and assess linear and nonlinear approaches when designing platforms or activities to enhance children's collaborative storytelling performance in similar contexts."* (Liu et al., p. 4791)

2.3.4 Göbel et al. (2008). StoryTec digital storytelling platform

Göbel et al. (2008) report the development of the StoryTec Digital Storytelling platform. The purpose of the platform is to provide an authoring tool for creating interactive, nonlinear stories. It is aimed especially at users who do not have the programming skills required in traditional, programming-based game development.

StoryTec software platform is a comprehensive authoring environment that consists of several components such as a Story Editor to create, organize and manage stories, a Stage Editor to create and manipulate story units, an Action Set Editor to define transitions, and an Asset Manager and Property Editor for accessing and manipulating story objects.

This research article is concerned with maintaining an efficient workflow for nonlinear story authoring, and usability features such as intuitiveness of the graphical user interface. It describes only the development of the method and does not discuss the pedagogical aspects of the storytelling process.

The research presented in this paper is based on earlier research projects such as the European Union-funded project U-CREATE and the Integrated Project INSCAPE. INSCAPE project was a large project with 14 research and technology development partners in Europe (Göbel et al. 2008). These projects are also related to another European Union-funded project, 80Days[7], which researched game-based pedagogy. The result of this research was a psycho-pedagogical framework that explores how games can provide adaptiveness for the learners' needs (Kickmeier-Rust et al. 2011).

StoryTec project has a website[8] but the latest updates are from 2011. The INSCAPE project predating StoryTec also has a live website[9] but the latest updates on the website are from 2008. It must therefore be concluded that these projects have been abandoned.

StoryTec has similarities with some modern storytelling software such as Alice by Carnegie Mellon University and the Fungus storytelling tool for the Unity 3D game development environment. Alice and Fungus are briefly introduced in Study III of this thesis.

2.3.5 Winkler et al. (2006). Nonlinear mixed-reality storytelling room.

Winkler et al. (2006) present a study on a nonlinear mixed-reality storytelling room. It is based on the evaluation platform for mixed-reality and high interaction media projects by Melzer et al. (2005). Together these are part of the Kids in Media and Motion (KiMM) initiative of the University of Luebeck, Germany. Researchers describe the initiative as being a "*holistic approach of media literacy*" promoting "*action-based and body-based forms of instruction for students and teachers in the classroom environment*".

Authors note that children are used to playing computer games and are exposed to nonlinear content. They explain: "*The impact on the children's reality from playing with these commercial applications is problematic: the commercial 3D-applications do contain non-linear structures, but the children are not aware of them*" (Winkler et al. 2006, p. 743)

In this study, a group of 15 girls and 10 boys from the sixth grade of the school participated in a nonlinear storytelling mixed-reality project. The project was a cross-disciplinary class of art, German language, mathematics, and computer science. In the project, students first visited a museum ship. Children collected their experiences into story drafts. Their multiple narratives were refined (see the original article for details) and combined into a nonlinear narrative.

The nonlinear story was implemented with computers. 3D avatar models were created with AvatarLab software. Stories were organized using a web-based KiMM-CommSy system. Sounds and voices were edited with Audacity software. Finally, a 3D virtual space combining stories, avatars, and sounds was created using KiMM-Studio software. A mixed-reality installation was created using the virtual space.

The participating group of students was compared to a comparison group (13 girls, 10 boys) that covered the same subjects and software, taught with traditional classroom methods. The assessment was performed with a 62-question pattern and showed highly positive outcomes for learning, motivation, and social interaction. Feedback from teachers was collected using a 73-question pattern and it reflected matching observations.

---

[7] EU Cordis project information for "Around an inspiring virtual learning world in eighty days" project, https://cordis.europa.eu/project/id/215918

[8] StoryTec project website http://storytec.de/

[9] Inscape project website: http://inscapers.com/

The research setup was highly complex and used a great variety of techniques. Therefore the effect of a single technique, such as nonlinearity in storytelling, is difficult to evaluate.

Improvement of systematic thinking is expressed in the article in two instances. This is interesting for this thesis and its objective of linking storytelling to computational thinking. First, the researchers state the learning goals (p. 744) as

> *"The lessons were meant to support and improve their linguistic, technical and artistic skills, as well as systemic thinking and acting."*

In conclusions (p. 748), researchers conclude

> *"The results of the project (the independent creation of a complex mixed reality space) and the findings of the inquiries show that this way of instructing students is not only very efficient in improving their learning of subjects at school, but helps to build their ability in systemic thinking."*

Systematic thinking is not further discussed in the article, so these mentions may refer to questionnaire answers such as questions 40 ("it was clear to them, what they need the applications and devices for and how they functioned") and 41 ("could generally effectively use the applications/devices and could use them for their work"). For these questions, the average scores for the test group were 4.56 and 5.0 respectively (scale from 1 = "does not apply at all" to 6 = "applies fully and completely.")

The special software KiMM-CommSy and KiMM-Studio used in the project are no longer available.

2.3.6 Nasir and Effendi (2012). Non-linear and interactive storytelling for children's book

Nasir and Effendi (2012) review different storytelling techniques in children's books. They take a special interest in the nonlinearity of storytelling. They define nonlinearity as follow:

> *"Non-Linear is a single logic sequence which also involve with flashback supported with multiple story line progress such as jump back to the very beginning of the storyline and could jump to forth"*

This definition includes both the definition of nonlinear narrative used in literature and film studies, as well as branching stories as defined in this thesis.

The researchers review types of storytelling techniques briefly and make three conclusions:

1) it is possible to introduce cause-and-effect narration to stories using nonlinear storytelling;
2) children can learn from choices provided by nonlinearity by exploring options and comparing the outcomes of their decisions; and
3) interaction in nonlinear stories must be chosen carefully as navigation can be

confusing.

In conclusion, they emphasize the importance of not using interactivity just as a *"fancy thing"* but to *"maintain the rhythm of the storyline."* They also conclude that technology can be utilized to make stories more interesting and to *"bring storytelling up to the next level"*.

2.4 Computational Thinking

Computational thinking is one of the key concepts in integrating information technology with education. It is used in contexts ranging from primary education to university-level computer science courses. This thesis explores the use of properties of nonlinear storytelling to develop computational thinking skills. Therefore, an overview of computational thinking is presented as follows.

2.4.1 Definition

The term "computational thinking" was first used by Papert (1980). It is related to some earlier terms, such as "algorithmizing", used by Alan Perlis, and "algorithmic thinking", used by Donald Knuth (Tedre and Denning 2016). Papert also used an alternative term, "procedural thinking" (Papert 1980, p. 155). The development of these concepts of novel computational thinking can be understood in the context of academic computer science in the mid-20th century. Computer science originated in mathematics and electronics, but it was developing into an independent field of science. This growth process required justification, and some pioneers, like Edsger W. Dijkstra (1974) and Donald Knuth (1974), considered unique mental processes to be a part of the special identity of computer science (Tedre and Denning 2016, 2017).

The concept of computational thinking became popular in 2006 when Jeanette Wing published an essay in which she suggested that computational thinking was a beneficial skill set for everyone, not only computer scientists (Wing 2006). In the original article, five components of computational thinking can be found (Shute et al. 2017), although they are not explicitly listed:

1) problem reformulation;
2) recursion;
3) problem decomposition;
4) abstraction; and
5) systematic testing.

In her later work, Wing uses the definition of computational thinking by Jan Cuny, Larry Snyder, and Alfred Aho: *"The thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent."* (Wing 2011; Aho 2012)

The concept has been examined and reiterated by many researchers. It is used from preschool level (for example Lavigne et al. 2018; Papadakis 2016) to universities and adult education (Perković et al. 2010; Günbatar 2019), so different interpretations of the concept arise naturally depending on the context.

An oft-cited source is the United States National Research Council workshop in 2009, which finally stated in their report: *"Even though workshop participants generally did not explicitly disagree with views of computational thinking that were not identical to their own, almost every participant held his or her own perspective on computational thinking that placed greater emphasis on particular aspects or characteristics of importance to that individual."* (National Research Council 2010, p. 59) It was considered in the report (p. 59) that computational thinking could be regarded as *"the union of these different views – a laundry list of different characteristics"*.

In this thesis, I will principally use the definition by the United States Computer Science Teachers Association (CSTA) and the International Society for Technology in Education (ISTE), if not otherwise stated. Their definition is used here because it is widely cited in academic research relevant to the topic of this thesis. According to the definition by CSTA and ISTE (Stephenson and Barr 2011), computational thinking includes the following elements:

- formulating problems in a way that enables us to use a computer and other tools to help solve them;
- logically organizing and analyzing data;
- representing data through abstractions such as models and simulations;
- automating solutions through algorithmic thinking (a series of ordered steps);
- identifying, analyzing, and implementing possible solutions with the goal of achieving the most efficient and effective combination of steps and resources; and
- generalizing and transferring this problem-solving process to a wide variety of problems

As stated, several other definitions exist for computational thinking that mainly agree with each other but differ in scope and detail. When other definitions are used, they are pointed out, and their differences are examined as required by the context. The concept of computational thinking is examined critically, especially in study IV where the topic is further extended.

2.4.2 Methods in education

Computational thinking and computer programming are being integrated into primary school education curricula in most European countries (Balanskat and Engelhardt 2014). For example, in Finland, starting computational thinking and computer programming from the first grade of elementary school has been part of the Finnish national curriculum since 2016 (Opetushallitus 2016). This has motivated the development of methods to develop computational thinking skills.

Methods used for developing computational thinking can be divided into those that are used in computer science and those used in non-computer science-related education. As explained earlier, the concept originates from academic computer science research and is used to understand what cognitive processes computer scientists and software developers use in their work.

At present, the concept is more often understood in the context presented by Wing (2006), as skills useful for everybody. This is also the context discussed in this thesis. Computational thinking skills are closely related to 21st-century skills and it is beneficial to include them as part of general education. Most of the students will not become computer scientists or computer programmers, but they will find these skills useful in their lives.

Therefore, a variety of methods and interventions have been developed to develop these skills. These methods are not easy to classify because many methods overlap. Shute et al. (2017) review methods and categorize them as using programming tools, using robotics, and using game design and other intervention tools.

Programming tools used in general education can be real computer programming languages. For example, Python[10] is considered to have a clean and human-understandable syntax even though it is a professional text-based programming language and is used in elementary school computer programming (e.g. Miller 2018, Tabet et al. 2016). Some programming languages have been designed for educational purposes, such as Logo (Kahn et al. 2020), or have been modified to be more child-friendly, such as Kids Ruby[11].

Repenning et al. (2010, p. 265) report the experience of using traditional text-based programming language at elementary school. A child expressed his earlier experience of computer programming saying *"Programming, oh no! I know what is going to happen. The teacher writes a program onto the blackboard, we type it into the computer and it never works."* This emphasizes how computer code has a precise syntax and getting the code typed correctly can be a difficult task in itself.

Visual programming languages have become popular in education to overcome the tedious task of typing code correctly. A common implementation of visual code is using code blocks. This is used by many well-known programming languages such as Scratch by Massachusetts Institute of Technology[12], Snap! by the University of California[13], App Inventor by Google and the Massachusetts Institute of Technology[14], and Alice by Carnegie Mellon University[15]. All these introduce the common procedural programming language paradigm, where the syntax of the language is simplified by using visual blocks. In addition, they are specifically targeted towards creating animations and games by having built-in graphics, sound, and animation tools.

In robotics, Lego[16] is particularly popular (Shute et al. 2017). The core programming language used with Lego MindStorms robots is the LabView visual programming language. It differs in appearance from the previously mentioned block-based languages as

---

[10] Python programming language homepage, https://www.python.org/
[11] Kids Ruby homepage, http://kidsruby.com/
[12] Scratch homepage, https://scratch.mit.edu/
[13] Snap! homepage, https://snap.berkeley.edu/
[14] App Inventor homepage, http://appinventor.mit.edu/
[15] Alice homepage, https://www.alice.org/
[16] Lego Robots for Kids, https://www.lego.com/en-fi/categories/robots-for-kids

blocks are not attached next to each other but connected by signal wires. The basic principle of replacing text-based syntax with predetermined blocks still holds. Alternatively, Scratch-based programming is possible, and a mobile app is available for simple experimentation.

Berland and Wilensky (2015) report that physical robots give an alternative perspective to learning. According to their study, using physical robots make students more likely to adopt a bottom-up or agent-based perspective, while students using the virtual system are more likely to use a top-down or aggregate perspective.

The third category of methods, according to Shute et al. (2017), is game design-based tools. They explicitly mention AgentSheets[17], an authoring tool used to design games and to teach computational thinking. This tool is also based on block programming and therefore has a large overlap with the previously mentioned visual programming tools. Basawapatna el. al (2011) present a study where five games were designed to study the development of computational thinking skills. These were traditional action (Frogger, Centipede, Space Invaders) and puzzle (Sokoban, Sims) games. The computational thinking patterns identified differ from earlier descriptions, for example, the CSTA definition, and include more game mechanics-type higher-order patterns of generation, absorption, collision, and transportation.

In addition to computer-based methods, computational thinking can be approached using physical methods. Kim et al. (2013) present a paper-and-pencil method to representing problems logically and solving them. Another well-known example of training computational thinking without computers is CS Unplugged[18]. In Figure 4 an example of CS Unplugged KidBots is presented. The goal of KidBots is to find control sequences or programs to guide a person in a grid. This is a familiar example of game programming and robotics.

In the study by Kim et al. (2013), it was found that students gained higher learning outcomes with paper and pencil compared to computer programing with Logo language. On the other hand, Grover and Pea (2013, p. 40) express their concern:

> *"Noteworthy efforts like CS Unplugged that introduce computing concepts without the use of a computer, while providing valuable introductory activities for exposing children to the nature of CS, may be keeping learners from the crucial computational experiences involved in CTs common practice"*

As seen, the core idea in all these methods is very similar. The methods are all based on procedural programming. That is, they contain a series of computational steps that are executed in order. Certain sequences can be organized as routines or subroutines, and together they form algorithms to solve problems. The problems are commonly chosen to be interesting to children and include animations and games. Robots are often used not only

[17] AgentSheets homepage, https://agentsheets.com/
[18] CS Unplugged homepage, https://csunplugged.org

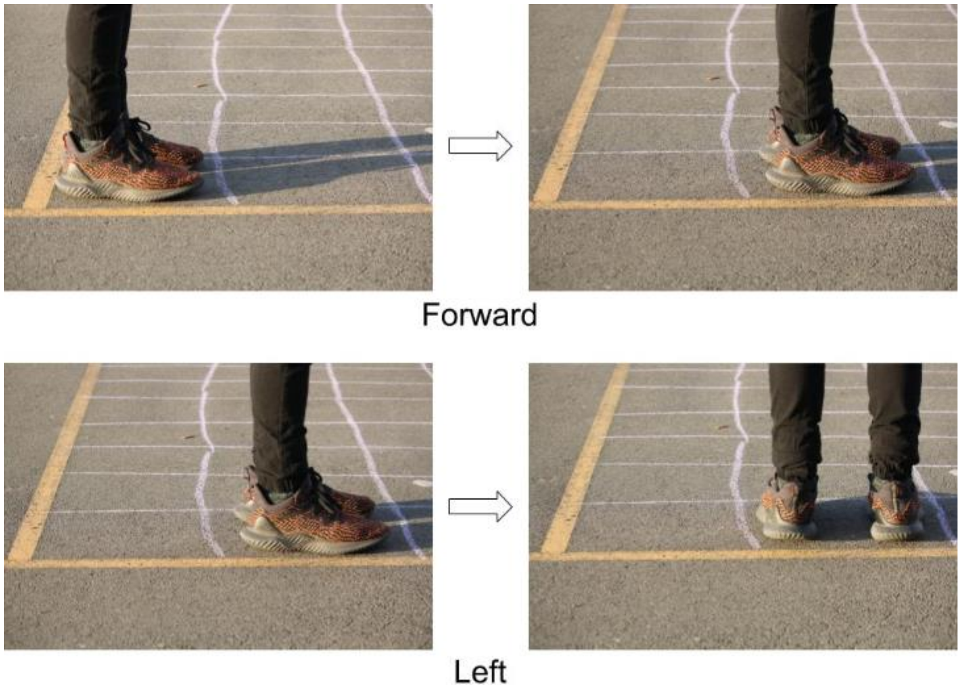

Figure 4. An example of training computational thinking without computers. KidBots from CS Unplugged, https://csunplugged.org/en/at-home/kidbots/. (CS Unplugged, Computer Science Education Research Group at the University of Canterbury, New Zealand, CC BY-SA 4.0).

as physical tools but characters in games and puzzles, and even non-computer-based methods like KidBots.

3 The Research Outline

As I have described, nonlinear storytelling is a specific type of branching storytelling that has received less attention in research and educational practice. Nonlinear stories are structurally similar to computer programs. This connection may provide a method for using nonlinear storytelling in teaching computational thinking skills. In this research, I will study this possibility. Here I describe in detail the aim and the structure of this research.

3.1 Aim of the Research

The aim of this research is to develop an understanding of how nonlinear storytelling can be used in education. I am especially interested in the relationship of nonlinear storytelling to computational thinking. The hypothesis of this study is that nonlinear stories and nonlinear storytelling have specific properties, in contrast to those of traditional linear stories and linear storytelling. These properties can be utilized in educational applications to develop computational thinking skills. To study these claims, I seek to answer the following research questions:

1) What characterizes the basic properties of nonlinear stories in educational applications?
2) How can nonlinear storytelling be integrated into teaching computational thinking skills?

Existing research on this topic often depends on certain software tools or special applications. In this research, I try to approach these research questions using general storytelling processes as much as possible. The focus is on the definition of nonlinearity and the specific properties of nonlinearity. To emphasize this, the methods have been chosen to be simple and universal. Storytelling in particular is kept a separate feature from multimedia production, such as graphics, sound, video, or game development. The separation of the storytelling function from other multimedia and game production helps us to study the specific properties and features of nonlinear storytelling in an application-independent manner.

3.2 The process of the research

The academic research presented in this thesis is closely related to two practical research projects carried out in the South-Eastern Finland University of Applied Science. The first project was titled *"Competent Employees for Digitalizing Work"* (2016-2017). In this project, an educational computer game was developed and methods for incorporating nonlinear storytelling methods into game development were studied.

The second research project was *"Digital Stories 2.0 – Nonlinear Storytelling in Edu-*

cation" (2018-2019). In this project, some study materials for teaching nonlinear storytelling were developed. Several small-scale pilot studies were executed, and classroom experiences were collected to evaluate the usability of study materials. The study materials were published under open Creative Commons licenses[19].

Together these two research projects form the backbone of this academic work, as they have made it possible to study storytelling in classroom environments. The stories analyzed in this academic work were mostly collected from the participants of these projects.

Academic research has been mostly independent of practical work. It has built on the projects mentioned above and extended the theoretical scope, exceeding the actual project requirements. The ongoing academic research has also provided some feedback and guidance to these projects.

For example, study II gives a theoretical justification for the application of nonlinear storytelling in teaching computational thinking. Study IV shows that the boundaries of computational thinking are somewhat arbitrary and reflect the way we are used to interpreting the concept in traditional applications. It therefore gives reasoning and a rationale for a re-interpretation of the concept of computational thinking in storytelling applications.

The research papers in this thesis are largely independent works but they also form a concrete and logical sequence by developing the ideas of previous research. The rationale and motivation for each paper are described in the following.

The research articles related to these projects are presented in Table 2. In addition to the four papers included in this thesis, two closely related papers are also presented in this table to give a full picture of the process of the research. In the table, A1 (Letonsaari and Selin 2018) is a more technical paper and therefore excluded from this thesis. A2 (Letonsaari and Selin 2019) is a report of a practical test and was excluded because of its less formal nature.

[19] Materials are available at https://www.mv.helsinki.fi/home/mletonsa/twinekoodi/ Materials used in the research were in Finnish language and only available in Finnish at the time of writing this thesis.

Table 2. Research articles of the project and their rationales. A1 and A2 are additional articles excluded from this thesis.

Study	Rationale
I	A method for co-creation for participants of different skill levels was required for a serious game development project. Nonlinear storytelling as a co-creative method is presented. The key advantage of this method from the academic research point of view is the separation of storytelling from the other game development processes. This allows an independent analysis of the storytelling process, which results in a more generalizable outcome. Some educational properties of nonlinear storytelling are discussed.
II	The first research article raises the question of what the relationship between nonlinear storytelling and computer programming is. Writing a nonlinear story requires some understanding of computer game logic. Creating a complex game plot is somewhat similar to some programming tasks. In this article, it is shown that computational algorithms can be presented as nonlinear stories. In theory, this connection enables the use of nonlinear storytelling as a teaching method for computer programming and a method for developing some general computational thinking skills.
A1	The open-source software used in our research is not designed for educational use but for hobbyists who write interactive fiction and multiple-choice games. The contribution of this paper is to present social media integration for storytelling software. This allows greater opportunities for sharing, remixing, and collaboration, which is needed in classroom teaching activities.
III	During studies I and II, we investigated several methods of writing nonlinear stories. We also examined some technical properties of the stories created and collected during those studies. Both studies had a narrow focus and much of this background research exceeded the scope of those articles. In this article, we present some of this research and analyze the stories from a technical perspective. From this analysis, we pinpoint some practical considerations that must be taken into account in using nonlinear storytelling.
A2	This study reports the first pilot study in which we use the nonlinear storytelling method in practice. A single class of first-year business information technology students in a vocational college is given a short introduction course to some programming concepts and simple electronics using the nonlinear storytelling method developed previously. The rationale for this study was to discover practices for using the nonlinear storytelling method in education.
IV	Our experiences with the nonlinear storytelling method gave us reason to criticize how the concept of computational thinking is interpreted and used in educational sciences. In many ways, the concept is closely rooted in traditional information technology education. This point of view is sometimes too narrow to provide a basis for discussing the approach of the nonlinear storytelling method. In this article, the concept of computational thinking is extended by using the philosophical concept of extended cognition. This challenges the traditional approach to understanding the concept of computational thinking.

4 Study Designs and Methods

The aim of this research was to study how nonlinear storytelling can be used in education, and what the relationship of nonlinear storytelling to the concept of computational thinking is. While some small-scale pilot studies were carried out, most of the research is qualitative and related to the development of the storytelling method. The methods used in each study in this thesis are explained in the following sections.

4.1 Study I: Co-creative serious games design process using nonlinear storyline editing

In this study, an implementation of a practical co-creative storytelling process for game development was examined and developed. The purpose of the study was to create a game development process that enabled people with low-level information technology skills to participate actively. This problem was approached using a design science research method (Hevner et al. 2004, Vaishnavi et al. 2004). Participant information was collected using a questionnaire. Observations with open interviews were used to collect feedback. The study is part of an educational research project in which a serious game was designed and implemented. In the following, design goals and principles for the study are presented.

4.1.1 Serious Game Definition and Design

A serious game is a game whose main purpose is not entertainment. There is no single definition, but the general idea is that the game has some purpose other than entertainment, for example in the fields of training, advertising, simulation, or education (Susi et al. 2007). Djaouti et al. (2011) have presented a history of the concept of the serious game. They note its similarity to the Renaissance term *"serio ludere"*, serious play. This term, which can be understood as an oxymoron, indicates *"deal[ing] with serious things in a playful manner"* (Manning 2004, p. 154).

Djaouti et al. identify the first modern use of the term "serious game" to the book *Serious Games* by Clark Abt (1970). For a current definition, they refer to Michael and Chen's *"games that do not have entertainment, enjoyment or fun as their primary purpose"* (Michael and Chen 2005, p. 11). Susi et al. point out some related and overlapping domains like *"e-learning, edutainment, game-based learning, and digital game-based learning"* (Susi et al. 2007, p. 2).

In this project, the purpose of the game was to educate users on the skills and knowledge needed in modern working life. The knowledge was identified to be non-cumulative by nature, so an episodic structure (Baranowski et al. 2008) was chosen. The game world was constructed to be an open-world type (Juul 2002) and included stories as learning minigames (Van Geit et al. 2015).

4.1.2 Game Development

The study describes a method for co-creative serious game development. In this method, nonlinear storytelling, also called game writing, is separated from other, more technical, game development tasks.

Usually, when game development is used for educational purposes, students create the whole game by themselves. The integrated game development is done using, for example,

- a general programming language such as C++, Java, or Python (e.g. Begosso et al. 2012; Leutenegger and Edgington 2007; Wang 2009)
- an education-oriented, low-threshold language such as Scratch (e.g. Ouahbi et al. 2015; Ozoran et al. 2012)
- specialized game development software or a game editor (e.g. Fowler and Cusack 2011; Kahn 1996; El-Nasr and Smith 2006)

With integrated development processes like these, students do much more than just storytelling. Depending on the application, they may need to learn, for example, computer programming, software design, software usability, computer graphics, and sound engineering in addition to many other skills.

Since game development involves learning many technical tasks, it is most efficient if all these tasks are part of the curriculum itself. For example, computer game development projects that use general programming language are suitable for information technology students who already have a technical background. In addition to computer programming, software usability, computer graphics, and sound editing are often part of the curriculum in information technology studies. Such a comprehensive project is therefore an exercise to practice a diverse set of skills.

On the other hand, a low-threshold language such as Scratch or specialized game development software is suitable for general introductory information technology courses such as information technology classes in elementary school. In these classes, computer programming and advanced technologies are not important; the objective is to expose students to many aspects of computer usage. These aspects can even be introduced superficially, but together they provide students with a good general understanding of information technology.

In this study, learning technical information technology skills was not the goal. The goal was instead to inspire students to use storytelling to create learning experiences about working life skills and knowledge. The goal was also to integrate these activities with the broader education of the students. The storytelling or game writing task was therefore designed to be an independent part of the game development. To assess participation, student activities were observed and evaluated by the author.

It can be noted that the specialized task of game writing is common in commercial game development projects where specific game writers play an independent role (Robison 2008; McDevitt 2010). But to our knowledge, game writing has not been separated from game development in serious or educational game development in earlier research.

4.1.3 Execution of the Study

This study was carried out as a part of the Competent Employees for Digitalizing Work project, organized by the Workers' Educational Association of Finland and funded by the European Union Regional Development Fund. The target group participating in storytelling tasks was people over the age of 30 years who were either unemployed or in job loss transition, and who possessed no vocational qualification, or whose vocational qualifications were no longer coherent with the requirements of work.

The Workers' Educational Association of Finland organized training in three different cities, Mikkeli, Savonlinna, and Pieksämäki. A part of the training was to participate in the development of a serious game.

The game development was coordinated by teachers at the South-Eastern University of Applied Sciences. Much of the actual software development was done in co-operation with university information technology students. The University of Eastern Finland participated in the project by providing feedback on the pedagogical feasibility of the working methods.

The Unity 3D game engine was used as a technical tool in game development. It was chosen because participating university students had already used it in their computer game programming course. It is also free of charge for non-profit game development, enables the creation of modern-looking 3D games, and has high portability. Portability in this context means that the product can be used on different platforms, such as operating systems, on a web browser, and mobile devices.

Nonlinear stories were written with Twine, which is a tool for writing interactive nonlinear stories. It has been used with 10 to 12 year old children (Tran 2016; Davis 2013) so it has been demonstrated to be easy to learn even with low information technology skills. It is a text-only tool that lets the user concentrate on the content. Stories written with Twine can be integrated with Unity 3D game development using the Cradle plugin[20] (known earlier as the UnityTwine plugin).

4.2 Study II: Modeling computational algorithms using nonlinear storytelling methods of computer game design

Nonlinear storytelling bears certain similarities to computer programming. Firstly, nonlinear stories in computer software and on the Internet, commonly known as interactive fiction, are a type of game in themselves. Computer games generated from nonlinear stories are computer programs, which means that they are analogous in a certain sense. Secondly, there are similarities on the structural level: branching of execution is a fundamental part of the logic of both computer programs and nonlinear stories.

These similarities justify considering that there may be a connection between nonlinear storytelling and computer programming. But more formal verification is needed. In this research, I am especially interested in educational applications in computational thinking

[20] Cradle plugin and its documentation is available at https://github.com/daterre/Cradle

skills and computer programming. Therefore, in this study, appropriate computational models are chosen to be validated in algorithm implementation using nonlinear storytelling. The research method follows constructive research in computer science, where solutions are constructed to provide objectively argued, but not necessarily formal, validations (Guimaraes 2011, p. 31).

The open source storytelling software Twine was chosen as a tool in writing nonlinear stories. Several other possible options for writing nonlinear stories were examined, but their analysis exceeds the scope of this study and is presented separately in study III. For all nonlinear storytelling tools, basic branching functionality is a common feature. In addition, many types of storytelling software have advanced scripting capabilities, such as variables and conditional branching.

Twine software was already used in Study I and we had good experiences using it with students. Standard Twine 2.x stories[21] do not include any graphics, video, or sound. This makes the story-writing process and the stories themselves explicit and clean. Nonlinear stories written with different tools are generally homologous with each other. More general results are often achieved if a simple format is used at first and extended to more complex cases only when needed, as it is technically easier to prove homology in simpler cases.

In addition to its clear text format, Twine includes visualization tools that are useful in comparing stories to visual programming such as finite-state machine state diagrams. Figure 5 presents an example of such a visualization of a simple turnstile machine (left) presented in the state diagram (middle) and as a Twine story (right).

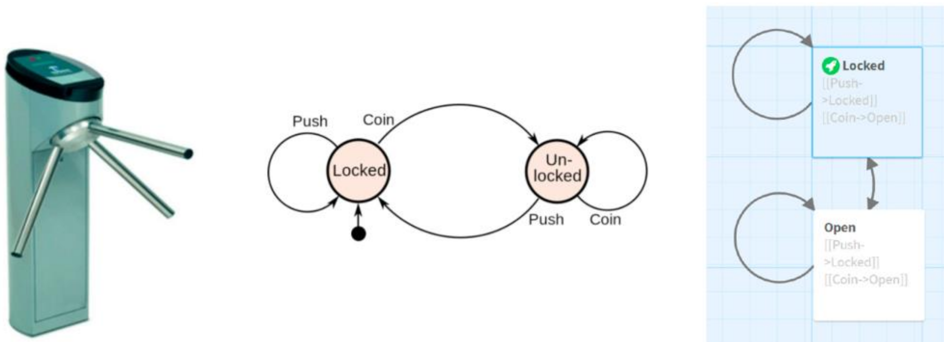

Figure 5. Left: A turnstile machine (Wikimedia Commons, Sebasgui, GFDL). Middle: State diagram for a turnstile machine (Wikimedia Commons, Chetvorno, CC0). Right: A turnstile machine function implemented in the Twine story.

[21] Older Twine versions with version numbers 1.x included the ability to include multimedia in stories. The Twine software was streamlined and many features were removed in version 2.0. In this thesis, Twine software refers to the latest version of the software, which is version 2.2.1 as of February 28th, 2019.

In this study, some classic algorithms and programming tasks are re-created using nonlinear stories in Twine. In general, there is no predefined set of tasks that a programming language must be able to implement to be considered a valid programming tool. Many programming languages are specialized in certain types of tasks. It is thus reasonable to implement programming tasks that are suited to the tool. Nonlinear stories are most useful for interactive programming tasks requiring user interaction, such as computer interfaces, control systems, and games.

First, finite-state machines are used. Finite-state machines are simple computational models with no memory (Wright 2005). But they have many important practical and theoretical applications, such as the turnstile machine and syntax parser simulated in this study.

Second, program flow control and loop structure are demonstrated with the use of variables and conditional branching. This is a classic implementation of abstraction. Here it is presented as an interactive version of the song *99 Bottles of Beer*. The website *99-bottles-of-beer.net* presents the implementation of the song with over 1500 programming languages to enable a comparison of different languages.

The last application presented in the study is the simulation of the famous Prisoner's Dilemma in game theory. The original formulation of the problem is used (Aumarm 1959). In this example, the structure of a logical problem is represented as a nonlinear story.

Implementations of these example tasks demonstrate whether nonlinear stories, here created with Twine storytelling software, can meaningfully represent some of the classic fundamental computer programming tasks. These tasks also feature many core ideas of computational thinking, such as abstraction, program flow control, conditional logic, structured problem decomposition, and debugging (Grover et al. 2013).

Presenting these properties in nonlinear stories would indicate that nonlinear storytelling is potentially a practical method for teaching computer programming and computational thinking skills.

4.3 Study III: Nonlinear Storytelling Method and Tools for Low-Threshold Game Development

This case study reports our practical experiences in choosing, testing, and using storytelling software tools in our research. First, software for nonlinear storytelling is reviewed from the point of view of educational research. This evaluation includes three main aspects:

1) We expect research to target people with low information technology skills. This target audience can be young novice users, students with learning disabilities, or adults who have difficulties in adapting to new technologies. Because of this, we are interested in easy-to-use, low-threshold tools.
2) While nonlinear storytelling can be used as-is, it is often more flexible and enables more possible use-cases if stories can be integrated with other tools, such as game engines. For example, in Study I, we integrated nonlinear storytelling into

professional game development. In other cases, we may want to improve the sharing and remixing of stories and create integration with the physical world such as Arduino electronics or robots. This is an essential feature for research where we want to explore new possibilities and use-cases.

3) We consider licensing of the software because in many research situations we require openness and freedom of use. Open-source licensing also allows modification of the software for specialized research purposes, such as logging user actions for analysis.

After the review, we analyzed the stories we acquired from our pilot studies. An educational computer game was developed by university information technology students and professionals. This pilot research was reported in Study I. Stories reported in Study I were collected from a single class of students at an early stage of the project. In this Study III, the stories analyzed were submitted at a later stage of the project by students from several classes at the end of the project.

The group of adults (age 30-64 years) participated in a basic information technology training course intended to enhance their employment opportunities. Submitting a story was not compulsory, so the number of stories is less than the total number of students participating in the research.

During one learning session, the serious game development project was introduced to the students, and students had an opportunity to play the game. The game is an open-world 3D game in which the player's avatar can walk around and complete educational tasks. During a second learning session, the basic use of the nonlinear storytelling tool Twine was introduced to the students.

After the introduction, students were instructed to write a sample story at the computer classroom in the school where they attended the training. Students were free to choose subjects for their stories, but they were advised that the stories were to be used for teaching skills and knowledge relevant to modern working life. The students were able to continue their stories on their own time after the session. Submitting their story for the game development research project was voluntary and not everybody wanted to include their work in the game.

4.4 Study IV: Extended Cognition Hypothesis Applied to Computational Thinking in Computer Science Education

One of the main questions of this thesis is to consider how nonlinear stories are related to the concept of computational thinking in the educational context. From the first study as well as from practical experience (Letonsaari and Selin 2018) it was noted that the storytelling approach to computer programming has some fundamental differences from the traditional approach to computer programming.

The concept of computational thinking, as it is used in education, has a close relationship to computer science and the traditional approach to computer programming, influenced by practical software engineering. Therefore, while computational thinking is considered a neutral set of skills for solving computational problems, it is perhaps unbiased only within a certain set of methods that have a close affiliation with software engineering.

There is little research on nonlinear storytelling but a large amount of research literature on computational thinking. Because of this discrepancy, there is little academic experience of how to interpret the concept of computational thinking in the context of nonlinear storytelling. In this Study IV, conceptual analysis is used to find a reasonable and justified extension to the applicability of the concept of computational thinking. Specifically, the extended cognition hypothesis (Clark and Chalmers, 1998) is used.

During the mid-20th century, information processing emerged as the standard method of understanding human cognition. The idea of human cognition as information processing has grown together with our understanding of the possibilities of information technology. The extended cognition hypothesis explains how the human mind interacts with its environment and the technology we use. The idea is that the cognitive processes of the human mind are not bound to our physical brains or our body, but extend to our environment when examined from the point of view of information processing. For example, a person with a pen and paper or with a calculator can perform more complex calculations than the same person without those tools.

The idea of extended cognition has made a major impact in many disciplines ,such as cognitive science (Arnau et al. 2014), psychology (Wilson 2014), and biology (Mikhael 2007). In the educational sciences, the extended cognition hypothesis and the related concept of enactivism are closely related to constructivist learning theories (Holton 2010).

The roots of computational thinking, on the other hand, lie in computer science and software engineering. Software engineering has been an independent, technology-driven field of study since the mid-1950s. Its long history can be seen in how information technology, computer science, and computer programming are taught today. In my approach, I seek to demonstrate how this might have defined the use of the concept of computational thinking as it is applied in modern education.

In this study, the history and main ideas of both of these concepts are laid out. I then proceed to examine the concept of computational thinking from the viewpoint of extended cognition. I expect this analysis to reveal some novel ideas, since software engineering generally has a very distinct separation of the user and the computer, a view that is challenged by the extended cognition hypothesis.

For a formal review of the subject, I use the Computer Science Teachers' Association's (CSTA) definition of Computational thinking (Stephenson and Barr 2011). Their definition includes the following skills:

- formulating problems in a way that enables us to use a computer and other tools to help solve them;
- logically organizing and analyzing data;
- representing data through abstractions such as models and simulations;
- automating solutions through algorithmic thinking (a series of ordered steps)
- identifying, analyzing, and implementing possible solutions with the goal of achieving the most efficient and effective combination of steps and resources; and
- generalizing and transferring this problem-solving process to a wide variety of problems.

5 Main Findings

In the following, the main findings of the studies are presented.

5.1 Study I: Co-creative serious games design process using nonlinear storyline editing

In this study, a co-creative process for serious game development was created. A modern 3D open-world computer game was developed in cooperation with information technology experts and non-ICT students with very low information technology skills. The students with low information technology skills participated by writing nonlinear stories. These stories were semi-automatically integrated into the game as part of the game plot.

The main result of the work is the development process, in which nonlinear storytelling is separated from the rest of the game development. This enables people with practically no information technology skills to partake in the game development. To the best knowledge of the author, this separation was implemented to a greater extent than in earlier existing research.

The separation in the method is based on writing dialogs in hypertext tools that enable linking storyline branches. Storyline branching, accomplished here using narrative methods, is one of the main techniques in gamification. It enables a concrete way of participating in the creation of a game. The specific implementation of the process developed in this study is presented in Figure 6.

The game development process was developed using agile methods (Beck et al. 2001; Laanti et al. 2013). This means that the finer details of the process are developed and updated during the process based on feedback and experience. In the educational context, this most commonly means observing students, their challenges and progress, and adjusting teaching according to what is required.

None of the participants (N = 14) had any earlier experience with console gaming, including handheld consoles, downloaded games, or multiplayer online games. Some participants did have some experience with single-player computer games (42%), mobile gaming (34%), Facebook games (20%), and other games played with an Internet browser (25%). However, the students carried out most of the storytelling tasks successfully, albeit with high individual variance in the amount of assistance they required.

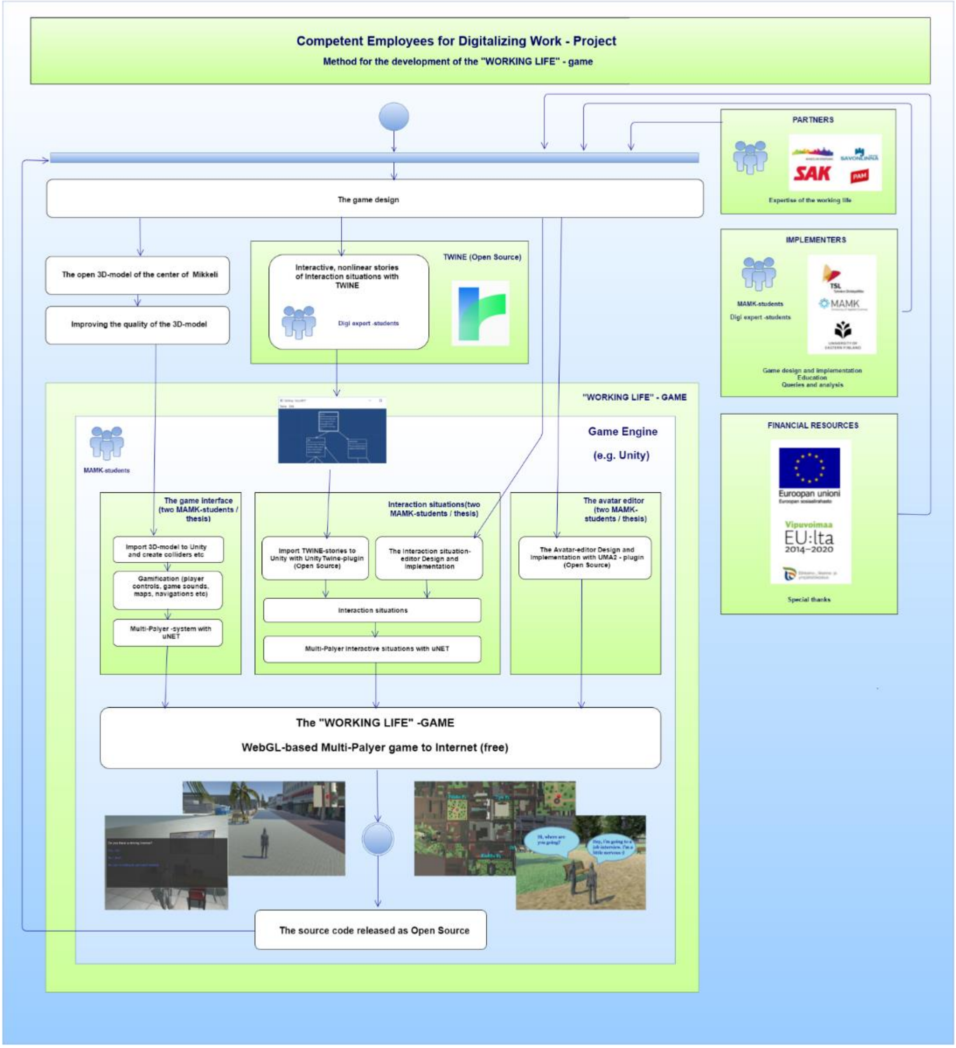

Figure 6. Flowchart of the process of developing a serious game used in this study.

It was noted that participation in the game development project generated excitement amongst the participants. Unlike most other activities included in their training, game development allowed freedom of expression. For this to be utilized in its entirety, the low-threshold learning curve in getting started is essential.

The students utilized their personal expertise in storytelling. In traditional game development, there are usually certain principles and limitations on what results are achievable by novice game creators. Students often learn by example, and the possibilities are controlled by what examples are provided. In storytelling, there are also certain rules, such as multiple-choice answers rather than free-form input, but students can generally use their creativity in storytelling. This made the end-product richer in content compared to those produced by traditional development methods. In this aspect, the co-creative process can be compared to the results of crowdsourcing research.

Students using storytelling also felt their participation in game development. Dialogs created by students for computer-controlled characters are important plot devices. Interactive computer-controlled characters create another, social dimension to the game world, even if the computer-controlled characters work with seemingly simple logic. The involvement in game development is deeper and more conceptual with formal storytelling tools than, for example, simply informally creating ideas for the game logic or character behavior.

By taking part in the development, students gained a perspective on game logic and how computer games function. This was not a set goal but something students learned as a byproduct of storytelling activity. An important observation regarding this thesis is that active nonlinear storytelling might generally help students to understand the logic needed in working with computers. This conclusion was based on informal observations of student activity and was not studied further in this study.

5.2 Study II: Modeling computational algorithms using nonlinear storytelling methods of computer game design

In this study, some fundamental programming tasks were implemented in nonlinear stories using the Twine storytelling software.

First, two examples of finite-state machines were implemented. Both the code for a turnstile machine logic and a number syntax parser express that nonlinear stories greatly resemble the state diagrams usually utilized in finite-state machine programming. Both nonlinear stories and finite-state machines also have similar application domains.

As the visual representation is important for evaluating the feasibility of the implementation, the story view presentation of the number syntax parser is presented in Figure 7 as an example.

The third example, the implementation of the *99 Bottles of Beer* song, provides a clear and concise solution to the task. While modern computer languages have specialized commands for looping, the basic abstraction of a loop using conditional branching is justified from the pedagogical perspective as it bridges the fundamental computer machine code to higher programming languages.

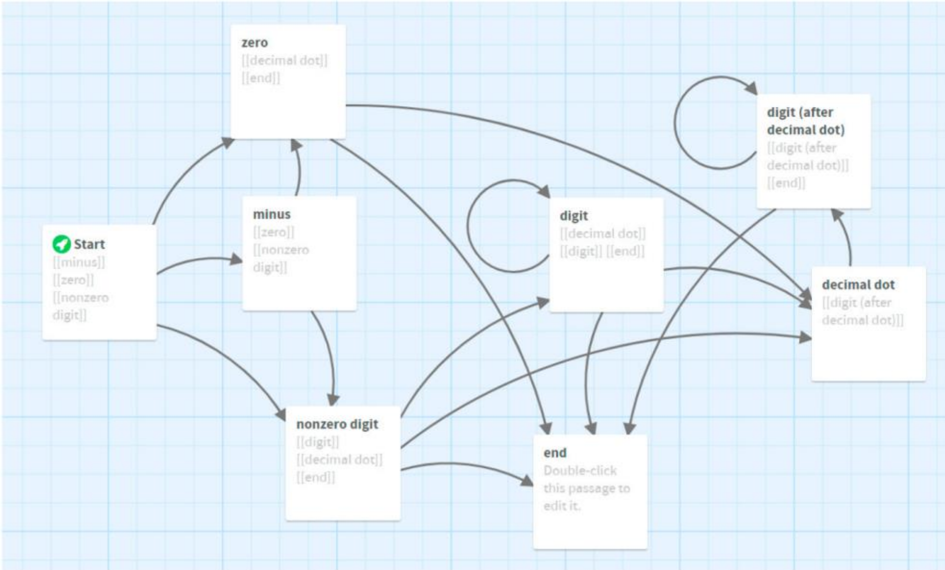

Figure 7. Parser for legal numbers implemented as a nonlinear story in Twine.

Lastly, the Prisoner's Dilemma is simulated using nonlinear stories. In this implementation, the visual structure of the problem is evident in Twine. In Figure 8 the Prisoner's Dilemma is presented in Twine in a form where the story flow is easily identifiable. By manipulating the structure of the story, just by dragging passages in the story edit window, we can find that actions can be transformed into a form that corresponds to the payoff matrix. The payoff matrix is an example of a normal form, sometimes known as a strategic form, which is formally used in game theory to present strategic interactions (Leyton-Brown et al. 2008). This correspondence is presented in Figure 9.

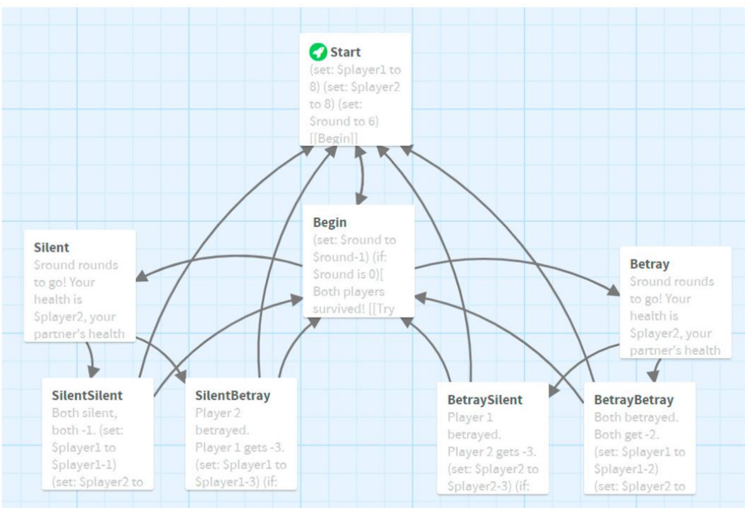

Figure 8. Prisoner's Dilemma as a nonlinear story in Twine.

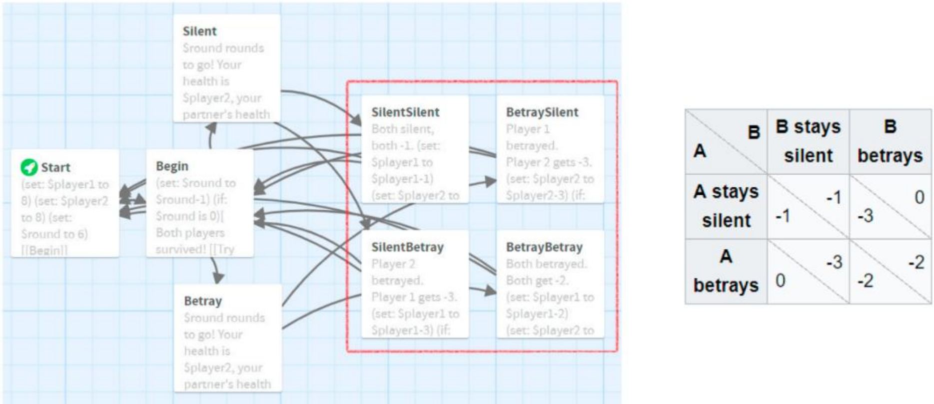

Figure 9. Left: Prisoner's Dilemma as a nonlinear story, as presented in Figure 8, transformed to emphasize the similarity to the formal payoff matrix presentation. Right: Standard payoff matrix presentation of Prisoner's Dilemma.

In the study, similarity to the Unified Modeling Language (UML) was identified. The UML is a widely used modeling language, especially in the field of software engineering. The UML has been characterized as having three general use cases (Fowler 2003, 2004):

1) sketching ideas and communicating them;
2) making blueprints and checking model validity; and
3) as a programming language.

All these features are also intrinsic to Twine. Twine can be used informally to write a story of the idea. The story can be validated by checking that there are no dead-ends (branches without a proper ending) in the story flow. And finally, an executable game, or another computer program, can be created from the story.

These example implementations of common computer programming tasks provide validation that nonlinear stories can be used for realizing many common algorithmic computations and computer programming.

5.3 Study III: Nonlinear Storytelling Method and Tools for Low-Threshold Game Development

In Study III, our experiences of using nonlinear storytelling software in teaching are reported. Here the main findings are reported.

5.3.1 Classification of Storytelling Software

Writing nonlinear stories does not need specialized software or tools. As explained earlier in section 2.2, the history of nonlinear texts predates computers and can be traced as far back as ancient Egypt (Aarseth 1997). But choosing the correct tool is still an essential decision and influences many practical aspects of writing stories.

In our earlier research, we reviewed software suitable for nonlinear storytelling. In this study, Study III, we present our experiences and a comprehensive list of nonlinear storytelling tools. The main finding here is that software can be categorized into three classes by its intended use:

1) **Text editing** – Any text editing software that is not intended for writing hypertexts can still be used for writing hypertext content. Most text editors do not have hyperlinking features, so the user must manually jump to the selected page or chapter on branching. Some text editors, such as word processors, have bookmarking and cross-referencing features that can be used to automate this.
2) **Hypertext and Wiki Software** – Hypertext editors such as HTML editors and Wiki software are intended for writing hypertext. Many of these editors have specialized functions for writing hyperlinked content and support syntax for hyperlinking as well as file formats that support hyperlinking.
3) **Game-oriented Software** – Software intended for game-oriented use has functions related to writing and controlling game plots. They may include visualization of the story structure or variables and script programming ability. This supports the creation of content in story format.

In this categorization, we explicitly concentrated on textual storytelling. These categories can be generalized for other types of stories such as video storytelling, audio storytelling, or multimedia storytelling, which combine various types of media for storytelling.

5.3.2 Observations on Collected Stories

In this study, students were trained briefly on nonlinear stories and the software used to write them. The teacher's availability to supervise and help with technical problems was limited because students continued their work unsupervised after school hours. The goal of the study was to discover what errors students make and what instructions are needed.

Several observations were made:

- some students didn't understand the difference between traditional dialog and dialog presented with a nonlinear story;
- hyperlink text can be confused with hyperlink target; and
- some stories were unfinished, and it was not clear if this was intentional or accidental.

In all these mistakes, the primary method is to improve the introduction and the instructions given to students. The relevance of the instructions depends on the students' background. It is important to explain the idea of nonlinearity and how it enables choice and storyline branching. If the students have a background in computer games, this idea may be trivial. But for students who have not played computer games, this might be a novel idea that needs to be explained with example game scenarios.

The relevance of correct syntax and hyperlink naming conventions can be easy for students who have a background in using desktop computers to understand. Related concepts include files and file systems, command line usage, and computer programming. For students who have less experience with desktop computers, the syntax may be presented in greater detail and with example use-cases. It may be useful to present some common mistakes and explain methods for checking the correctness and validity of syntax.

Some of the misconceptions and errors can be attributed to the properties of nonlinearity, while others can be related to the software or applications. Most errors can also be identified by examining the structure of the data. A lack of valid links in a story usually means that student has not understood the hyperlink syntax or did not comprehend the concept of nonlinearity, the effect of choice, in stories. Confusion between hyperlink texts and hyperlink targets can be detected from infinite loops and illogical structures. Unfinished stories can be found in empty passages.

5.4 Study IV: Extended Cognition Hypothesis Applied to Computational Thinking in Computer Science Education

In this study, the concept of computational thinking was analyzed from the perspective of the extended cognition hypothesis. It is common to both theories that they describe the relationship and interaction between human cognition and technology. But they address this connection very differently.

Cognitive science considers human cognition as an information processing process. Sensory information and information stored in memory are processed by complex and advanced cognitive processes in the brain and in the nervous system to produce human behavior.

Sometimes these cognitive processes seem to extend outside the human nervous system. For example, a child can use their fingers in simple calculations. Or a student can use a calculator to solve a mathematical problem. Raised fingers or numbers in a calculator have no meaning by themselves, but they are used in the cognitive problem-solving processes.

It can be considered that information processing is extended from the brain to an external and more functionally suitable medium. In the modern world, this external medium is often technology we have built ourselves. Technology is developed to enhance our functional capabilities in the environment, and these functions are integrated into our cognitive processes.

It is possible to define the limits of human cognition without including anything outside the human brain, the human nervous system, or the human physical body. But from the point of view of information processing in cognitive science , this is an arbitrary definition that has no factual meaning. Analogously, it is not computationally relevant to a computational algorithm if it is run on a single computer or a computational cluster, or if it uses cloud services to execute the algorithm.

Computational thinking, on the other hand, is rooted in computer science and software engineering. It considers the technical aspects of how computers work and how computers execute tasks. The concept of computational thinking is based on the idea that computers use another type of logic and technology, something that is not natural for humans. Computational thinking is seen as a method or a skill set that humans use to understand and utilize external technology.

For example, in the original article by Wing (2006), she emphasizes that computational thinking is *"a way that humans, not computers, think"*. This is a valid viewpoint, but it underlines the difference between the human and the machine. It externalizes the technology and enforces a boundary where one might not be necessary.

From the point of view of external cognition, the technical aspect of utilizing a tool is not the core idea of the relationship between humans and technology. Rather it is about how we can extend our ability to function and solve problems together with technology. Therefore, there is a clear contradiction between these two views. In this study, Study IV, the main aspects of this dichotomy are explored. In the following, the main findings are explained.

5.4.1 Extended Limits of Computational Thinking

When we talk about computational thinking, we often think of the modern computers found in computer classrooms. But where does the concept of computational thinking come from? Is it something specifically related to modern computers or did we possess computational thinking skills even before computers existed? When technology evolves, does this thinking change in some way?

In this section, the author presents a two-dimensional space where the limits of the computational thinking concept can be located. The first dimension is the level of technology. It starts from human prehistory without technology, going through different technological advances to the modern day with its digital computers. From here it is also possible to estimate and extrapolate what may be achieved in the near future. In addition to contiguous improvements such as higher computational performance and diminishing costs, more radical advanced technologies such as computational implant technology are often considered in the research literature (for example, Clark and Chalmers 1998).

The other dimension is the application domain. Computational thinking was defined as the skills required for solving problems with computers. Certain application or tool independent skills, such as abstraction, analysis, and automation, are needed in formulating problems adequately and implementing algorithms to solve them (see section 2.4.1). These skills are application and tool independent, and thus what kind of computational device or which programming language is used in problem-solving is of little concern.

These ideas are presented and discussed in this section of the article. Algorithms in mathematics are procedures for performing calculations and solving problems and their use long predates the invention of computers. It is noted that the earliest known algorithms were created as early as 4000 years ago (Knuth 1972). Creating and using algorithms as well as the related algorithmic thinking has existed in human history in many different forms. Complex tasks such as engineering, practicing business, and governing communities require similar features of thought as the computational thinking associated with computer programming tasks.

It is generally implicitly assumed that only the current level of technology is concerned. This excludes, for example, using human servants or employees as computational units. Sometimes it is not sure if state-of-the-art technology is taken sufficiently into account, and future technology is rarely considered in defining and evaluating computational thinking skills. The level of technology and the human cultures built around it specify and define the requirements and possibilities in applying computational thinking in everyday practices, and how computational thinking manifests itself.

5.4.2 Levels of computer programming

The concept of computational thinking is often used in relation to computer programming, where computer programming generally means solving problems and automating tasks using computers. How programming is done in practice has changed in many ways through history. Some important aspects are briefly explored in this section of the article.

In addition to the development of computer hardware, there are mainly two kinds of changes. Firstly, tools for software development have become smarter and more efficient. The simple low-level languages of early computers have gained many features that automate programming tasks and make programming easier, thus developing into high-level languages. Even high-level languages have changed greatly during the decades they have existed.

Integrated development environments let the programmer keep better track of resources such as variables, functions, and library files. They may also have features such as intelligent code completion to help with programming tasks.

The second important change, in addition to programming tools, is how computer code and information are shared in programming communities. This aspect includes code libraries and frameworks that allow the programmer to work on a higher level of code abstraction, and social information sharing such as online discussion forums.

In early computer literature, learning low-level language programming began with technical details such as memory handling. With high-level languages, such as the then-popular Basic and Pascal, computer programming consisted of learning the basic syntax of the programming language and some programming logic. With object-oriented languages such as Java, teaching often begins with the concept of an object and higher conceptualization on how computer programs are structured. Nowadays, less formal methods of learning are often preferred.

While these approaches share some features such as strict logic, their differences clearly require different kinds of thinking skills. For example, low-level skills are very mechanistic, while object-oriented programming tries to use terms inspired by natural language to describe a higher level of abstraction. Some languages such as visual Scratch hide some of the strict logic of textual computer code. In the future, technologies such as artificial intelligence and proof systems will probably further change programming work in many ways, following how it has already developed. While some aspects of computer programming have remained the same, I conclude that computer programming is constantly changing and not a skill set that can be defined statically.

5.4.3 Modes of Technology

In this section of the article, the relationship between multimodality and programming is discussed. Computer programming is traditionally a highly textual activity. Just as mathematics or physics describe the real world in symbols, computer programs use textual data to process events and phenomena related to computation.

Here the argument lies in two ideas. The first is the extended cognition hypothesis. Human cognition is not separate from the computer; rather the computer performs a part of the information processing process generated by human cognition. Human cognition is multimodal, so the human-computer interaction should reflect this.

Currently, the most common practical use of information technology is textual and graphical because of technological limits. Multimodality is expressed mostly in computer games and digital arts. Voice recognition and speech recognition are steps towards multimodality in everyday applications. It may not be technologically feasible to expect multimodality in computer programming today, but there is no reason in theory for not realizing multimodality.

The other aspect is the efficiency of multimodality. When using simple technology such as a toaster or a calculator, it is clear that more complex organisms must adapt to the simple one. This was the case with early computer programming. But we can see that the more complex information technology has become, the more adapting and anticipatory it has become. Part of this adaptation is the adoption of human multimodal function.

Finally, I present a fundamental example of how spatial thinking is related to skills in mathematics and science (e.g. Newcombe, 2010). Computer games and physical activities that enhance spatial thinking seem not to have any connection to the symbolic representation of mathematics. Mathematics can be performed entirely based on symbolic notations that are not dependent on the physical world. But human cognitive models benefit from spatial multimodality in symbolic processing. The effectiveness of multimodal human-computer interaction is therefore not only a theoretical hypothesis but has real-world plausibility.

5.4.4 Technology Synchronized Thinking

The final result of Study IV is the conclusion that the current use of the computational thinking concept reflects only a small part of the human-computer relationship. The concept is bound to computer science and software engineering that does not express the role of information technology in our everyday lives. Nor does it help us understand the challenges and possibilities that future technology presents to humanity.

In this study, the concept of technology synchronized thinking is suggested to complement the idea of computational thinking. By replacing computation with technology, we acknowledge that information processing is not limited to computation as understood by computer science i.e. algorithmic problem solving and data processing. Technology is a broader concept that encompasses tasks of everyday life, entertainment, and technological aspects of human social interaction in society.

Synchronization emphasizes the bi-directionality of the interaction. We do not expect technology to be an external thing that human cognition learns to operate. Instead, technology is a symbiotic phenomenon. Adaptation is reciprocal and the symbiosis pursues a state in which the human cognitive domain and technological domain are efficiently synchronized for purposes of cognitive information processing.

6 Discussion and Conclusions

In this thesis, two research questions were answered. Firstly, I examined the basic properties of nonlinear stories, especially in educational applications. Secondly, I studied how the concept of computational thinking can be approached from nonlinear stories for educational applications. I evaluated nonlinear storytelling along several aspects, specifically

- Study I: Nonlinear storytelling can be used as an independent method, but it can be more meaningfully integrated as a part of a more complex project. This study presented a practical co-creative serious game development project that used nonlinear storytelling. Storytelling was a modular part of the game development process, so it could be studied and analyzed as an independent task. In this study, the process of implementation was designed and discussed.
- Study II: This study showed that nonlinear storytelling can be used to implement many fundamental programming tasks meaningfully. Regarding the first research question, it was demonstrated that nonlinear stories have similar functional properties as computer programs. This similarity implies that nonlinear storytelling is a valid method of developing computational thinking and computer programming. This also provides an understanding of how nonlinear stories and computational thinking are related.
- Study III: In this paper, the properties of nonlinear stories were studied further. First, various tools for nonlinear storytelling were analyzed. The method of nonlinear storytelling itself is application-independent but the tools can influence the storytelling process significantly. Three fundamental software categories were discovered: (1) software that does not have hyperlinking as one of its main features (such as text editing), (2) software for hypertext editing (such as HTML editors and Wiki systems), and (3) game-oriented software. A small-scale study was conducted, and nonlinear stories were collected from novice users. These stories were analyzed and challenges in creating nonlinear stories were exposed.
- Study IV: Some understanding of the relationship between nonlinear stories and computational thinking was gained from the previous studies I-III. In particular, it was shown in Study II that nonlinear stories can be used to present computer programs. But nonlinear stories are significantly different from the traditional computer programming approach to computational thinking. Even methods such as visual programming (such as Scratch) and computer science unplugged (Bell et al. 2015) are mostly targeted at merely understanding traditional computer programming in simplified form. This raised the question of whether we can extend the concept of computational thinking, or if we are bound to its current definition. In this study, I examined the concept from the viewpoint of the extended cognition hypothesis. According to this well-established hypothesis, thinking is not something that happens only in our brains or inside our bodies, but cognitive

processes that can extend into our environment and especially into the technology we use. In this study, it was shown that computer science and software engineering have had a significant influence on how computational thinking is understood and taught in schools. It is not self-evident that the current interpretation of computational thinking is the only valid interpretation. Interpretations exist that are less computer science-oriented and it is possible to choose which of them to use with our nonlinear storytelling approach.

Together these four studies provide answers to the research questions. The results are not definitive, yet they build some modest but fundamental ground for using the method of nonlinear storytelling in the development of computational thinking skills. In the following, further discussion and some conclusions are presented to summarize the results in detail and reflect on them within the broader topic of nonlinear storytelling in educational applications.

6.1 Storytelling activity

Storytelling as an activity was investigated in Studies I and III. In the following, I discuss the main topics of storytelling activity.

6.1.1 Nonlinearity

What differentiates storytelling here from most educational storytelling is its nonlinearity. As a term, nonlinearity is not intuitively comprehensible for everyone. In practice, it is better to introduce this type of storytelling as gamified stories, interactive stories, or stories with multiple alternative plots. This choice of wording often provokes an association with already familiar concepts such as games and interaction. But nonlinearity is a single distinct term to describe the essential property of storyline branching.

In many cases, gamified and interactive stories are created without recognizing the nonlinearity that is created. To summarize this research, there are three main reasons why nonlinearity should be recognized in storytelling:

1) The nonlinear structure allows for richer and more complex stories. These in turn enable more natural gamification. Understanding nonlinearity in storytelling is required to realize and utilize this potential.
2) Creating gamification is not an intuitive task for everybody. Some of the difficulties arise because gamified stories have nonlinear structures that are more complex than those of traditional linear stories. It is therefore important to recognize nonlinearity to understand and overcome the difficulties.
3) To utilize information technology efficiently, computational thinking skills are needed. Understanding abstraction and dealing with complexity are computational thinking skills. These skills are also needed in writing nonlinear stories, but storytelling is a less technical task compared to many other activities used to train computational thinking.

Considering the first aspect, nonlinearity can be seen as a tool to build the world (sometimes called the universe) where the story happens. Nonlinearity allows a world in which decision-making is enabled. Defining the nonlinear structure of the story decides what plots are possible, what structures plots form, and what complexity is created in the story.

Gamification and interaction can be implemented in a linear story. For example, a certain task must be completed before the story continues. But this is a world where the user cannot make decisions that have consequences. In gamification, we decide what rewards and penalties we wish to assign to the player. Nonlinearity defines how these can be accomplished in terms of different plot outcomes. Therefore, nonlinearity supports natural gamification and interaction in the story.

In addition to understanding the new possibilities that nonlinearity creates in storytelling, the other important aspect to consider is the added layer of abstraction. In writing nonlinear stories, these elements are storyline branching, including possible syntax and function, and alternative plots. Study III discussed the challenges caused by this abstraction.

The most problems are caused when nonlinearity is not explicit but a hidden implicit feature. For example, game editors may assume that the user is familiar with possible game structures. In computer programming, the user is responsible for designing the story structure. If the user is not familiar with such ideas, they can easily become confused. It seems that these challenges can be more easily understood and solved when the nonlinear nature of the gamified story is realized, although more detailed research is needed.

6.1.2 Tools for Nonlinear Storytelling

In Study III nonlinear storytelling software was reviewed. The main result was three practical categories of storytelling software, as presented in section 5.3.1. While there is not much software intended exclusively for nonlinear storytelling, nonlinear stories can also be created with many suites of software that were not created primarily for nonlinear storytelling. This is the second main result of the study: the realization that there is no shortage of suitable software tools.

In our research, simplicity of software was chosen as one of the favored key features. For research purposes, this can be reasoned as a good base for software-independent, generalizable results. We were interested not in a specific piece of software but the core ideas of nonlinearity in storytelling.

The same principle of simplicity has also found support in education. For example, Roschelle (2003, p. 24) summarizes the research in that we should move *“towards simple, well-honed technology and rich, pedagogically developed social practices”*. He continues to go so far as to claim that *“'pedagogical applications' are often led down the wrong road by complex views of technology and simplistic views of social practices.”* This is a good guideline for selecting educational software generally: software should support pedagogy.

That being said, the software review in this research was limited to textual multiple answer storytelling. There is also non-textual software as well as more free-form storytelling software. For example, presentation software such as Microsoft PowerPoint, Google

Slides, or OpenOffice Impress can be used for nonlinear multimedia storytelling, including images, sound, and video. Multimedia and digital technologies have been used with storytelling to provide engagement and co-creation.

There has also been much research in interaction technologies. This has recently received practical applications such as voice recognition, facial recognition such as emotion detection, and the use of movement and expressions by video recognition and accelerometers with mobile technology. These have been omitted from our current research, but they provide interesting prospects for future applications.

6.1.3 Integration of Storytelling with Game Development

In Study I, the integration of nonlinear storytelling with a game development project was presented. This was a highly integrated process of game development as novice users could design game plots and dialogs for the game.

An example of a previous, less integrated method of co-development is to collect stories and ideas before the development process. These stories are either traditional linear stories or often just abstract ideas. They are collected and later integrated by developers into the final product (e.g. All et al 2013).

Experts and end-users can also operate in software testing and validation and provide feedback to the developer team. This kind of testing can take many forms, but it generally provides fewer opportunities for creative development by end-users (Bossavit and Parsons, 2016; Cosma et al. 2015).

Co-operation with low skill level participants in the game development can be implemented on a visual level. Users may, for example, modify characters by customizing physical attributes or choosing their clothing and accessories. They might also be able to edit the game world with graphical editors such as level editors.

Graphical editing of characters and the game world is an engaging activity, as the results are easily recognizable in the game. Graphics are easily separated from the game logic and require very little understanding of the game design. In contrast, writing a game plot is a more conceptually involving process that goes deeper than the appearance of the game.

6.1.4 Development of Storytelling Tools

In this research, Twine storytelling software was used as a tool for writing stories. The reasons for selecting this software were explained in Studies I and III. One of the selection criteria was being open-source software. This allows modification of the software tool to adopt advanced features.

The most important development feature that was experimented with was social media integration. In early pilot studies, it became evident that game development is a highly social activity. Students want to share their creations and let other students test their games. Most storytelling tools are built not for educational classroom situations but for individual developers. Therefore, story sharing is not always present.

To address this lack, social media sharing features were developed. This is presented in Figure 10. Instead of publishing the game on file, it can be sent to various shared social media accounts or copied as a hyperlink. Stories were stored at the project server. The development of this feature is presented in Letonsaari and Selin (2018).

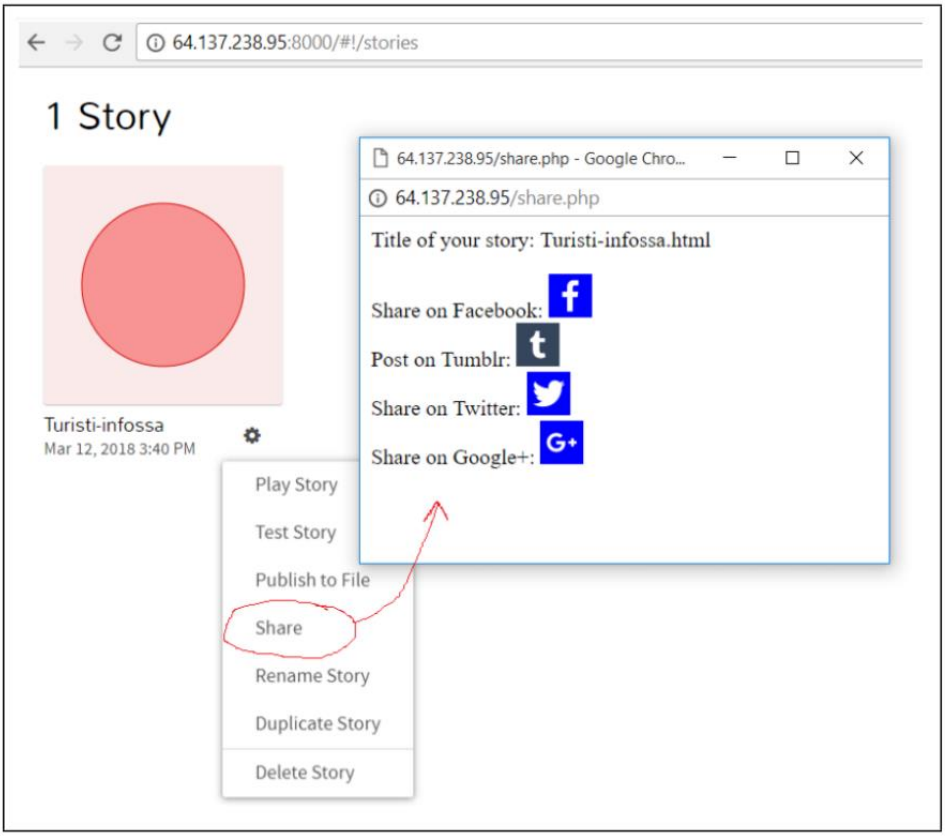

Figure 10. Social media sharing features developed for Twine storytelling software.

Another experiment with the possibilities of storytelling was developing mobile apps. Games developed with Twine can be played with mobile phone browsers but installing a game as an app gives a greater feeling of ownership in the product. Ownership as quality has been discussed, for example, by Liu et al. (2010). Creating a mobile app is a somewhat more technical task and not suitable for all audiences. Instructions for creating mobile apps were created for advanced users[22] in the project and some games are available as Android APK install packages[23].

[22] Available in the Finnish language:
https://www.mv.helsinki.fi/home/mletonsa/twinekoodi/indexfe70.html?page_id=253
[23] Games "Detective" and "Origins of Apollyon". The games are in English, but the website for downloads is available only in Finnish:
https://www.mv.helsinki.fi/home/mletonsa/twinekoodi/indexdc18.html?page_id=279

In my research plan, I prepared to integrate storytelling into electronics and robotics. To do this, I acquired Arduino development boards and small humanoid robots. A robot is shown in Figure 11. The idea behind using electronics and robots was to make storytelling more concrete by providing physical applications.

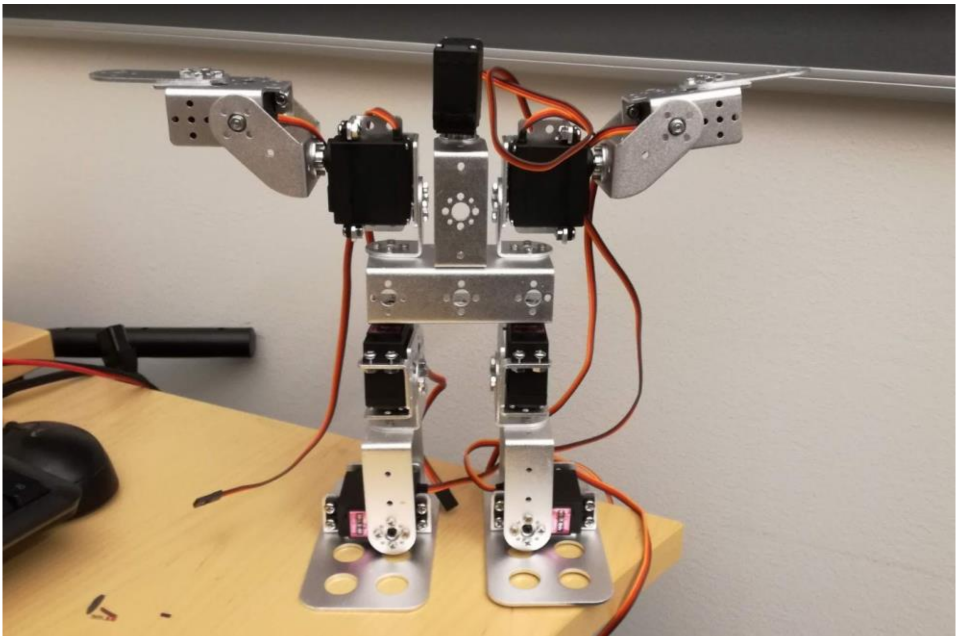

Figure 11. Humanoid robot tested in the project.

In my tests, it was possible to control a robot through commands embedded in the story. I developed a syntax where control commands were inside parentheses so that the robot could act out the story being told. For example, the following code inside Twine story turns the robot head to the right and takes a step forward:

(move head-right)

(move stepforward)

Unfortunately, while the storytelling software itself does not require installation and can be used in a web browser, sending a command program to the robot requires installing Arduino software, downloading story files to the disk, and running a command-line script for converting the story into Arduino language. This made the use of robots overly complicated, and it was used mostly by the teacher for demonstration purposes only.

In our experience, it is better to keep technology simple and focus on a single task. This means focusing on writing nonlinear stories, which proved to be an engaging task in itself. Storytelling also provides some advanced features and has the potential to become a deeper learning process. In most cases, this appeared to be a more productive topic than the use of robots.

6.1.5 Philosophical Considerations about Nonlinear Stories

Traditional linear stories do not have branching. This is how we see things happen in the world. We can see only one history and present. If an event happens, for example, France won the football world cup in 2018, it is unambiguously true, and no other country won. The day before the final match, nobody could know the result. But once things happen, they form a chain of incidents that cannot be changed – and this is how traditional linear stories present the world (Liu et al. 2010, Adams 2009).

Causality is the idea that everything that happens has a cause. In its strict form, incompatible determinism claims that there is no room for free will but everything that happens is predetermined by the previous state of the world (Clarke and Capes 2000). We do not need to consider this problem, because humans subjectively have a concept of free will and that is enough for us.

But it is worth considering how well nonlinear storytelling reflects our decision-making and our physical world. In nonlinear storytelling, the user makes discrete decisions and there is a limited number of possible plots as opposed to an infinite amount. This usually fits well with our mental model of how we act in the world. In the morning we may have a cup of tea or a glass of orange juice with breakfast. The third option can be water or maybe nothing to drink at all. It seems that we have a few options that we can choose from.

With this idea, we can imagine that we can simulate even very complex events with nonlinear stories. For example, work processes or problem-solving have this high level of causality. Causality is also something we expect, so it is easy to consider these discreet plot structures as a good representation of the real world. In the human brain, decision-making in discrete choices is done by the so-called winner-take-all mechanism (Coultrip et al. 1992).

But in the real world, some things and actions cannot be described as being discrete. For example, the time of events is a continuous variable. By an accurate measurement, we can find small time differences in seemingly similar events. Another similar example is the amount of matter, for example, liquid. We can mostly ignore these small differences. But sometimes they can affect the outcome of events significantly. For example, a very small time difference can change whether an accident happens or how severe the consequences are. Generally, this phenomenon is called the butterfly effect (Lorenz 2000).

Nonlinear stories, therefore, simulate some aspects of reality better than others. When we experience an event such that we can contribute to decision-making, it is amenable to credible simulation. Examples of conscious and unconscious actions in psychology and our subjective experience are presented, for example, by Csikszentmihalyi (1991).

6.2 Computational Thinking and Nonlinear Storytelling

Jeannette Wing famously said, *"Computational thinking is a fundamental skill for everyone, not just for computer scientists"* (Wing 2006). At the end of her article (p. 35), she makes a practical suggestion: *"Professors of computer science should teach a course called "Ways to Think Like a Computer Scientist" to college freshmen, making it available to non-majors, not just to computer science majors."*

Wing probably referred to college students and educated professionals when she used the word "everyone". The idea of computational thinking in education has evolved since. The principles of computer programming are now introduced to even preschoolers and used in education at all levels of elementary school. Computational thinking is not just for higher education occupations but really for everyone and everyday life.

In Study IV I explored some ways to extend the concept of computational thinking in this regard: Who are the students learning computational thinking skills? In what situations will they use these skills? What tasks do they perform using these skills? Here I will discuss these findings.

6.2.1 Technology-driven change

The acceleration in the growth of technological development has been documented in many sources. The most well-known is Moore's law, which states that the number of transistors in integrated circuits doubles about every two years (Moore 1965). The original observation was made in 1965 and the trend has continued since. Moore's law is presented in Figure 12. Notice the logarithmic scale on the vertical axis. In fifty years, the number of transistors has grown from a few thousand to tens of billions.

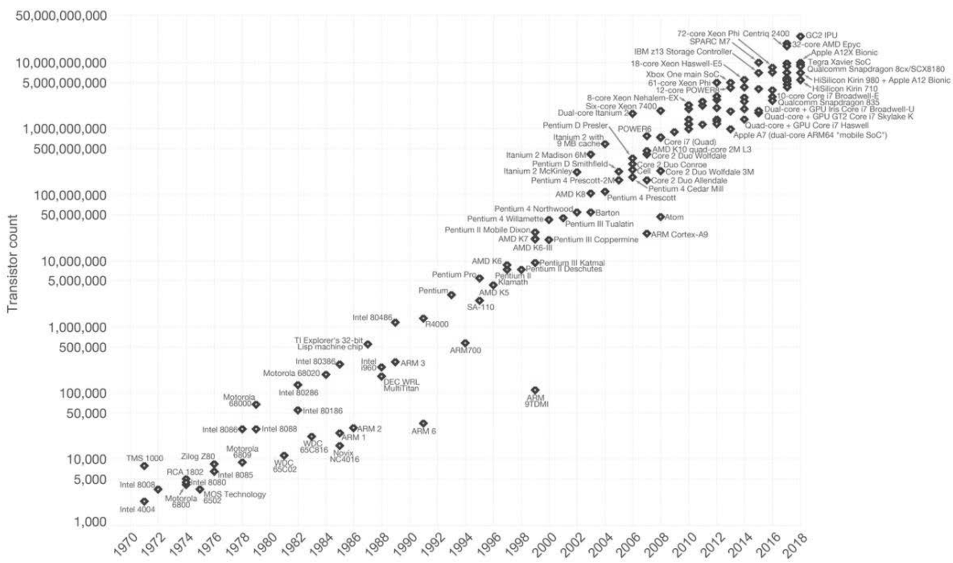

Figure 12. Moore's law (Wikimedia Commons, Max Roser, CC BY-SA 4.0)

Koomey's law offers an alternative viewpoint comparing energy per computation. According to Koomey et al. (2010) the number of computations per unit of energy has doubled every 1.57 years from 1946 to 2009. As an exponential growth rate, this means about a hundred-fold increase per decade. Exponential growth rates have also been observed in computations per unit of price (Flamm 2002).

Yet another perspective on the accelerating change is the adoption time of different technologies. For example, in the 19th century, it took over 30 years for new technologies such as electricity, telephone, and radio to achieve a 25% adoption rate in the USA. Television, invented in the early 20th century, had an adoption time of 26 years. But newer technologies such as the personal computer, mobile phone, or the Internet, have had adoption times as low as seven years. See Table 3 for more detailed data.

Table 3. Mass use of inventions (Kurzweil 2005).

First year	Years until 25% adoption	Invention
1873	46	Electricity
1876	35	Telephone
1897	31	Radio
1926	26	Television
1975	16	PC
1983	13	Mobile Phone
1991	7	The Web

The US Bureau of Economic Analysis calls these technologies digital-enabling infrastructure, as they enable the digital economy (Barefoot et al. 2018). In addition to computer hardware, the infrastructure also includes software, telecommunications, physical structures, and the internet of things.

The digital economy has several definitions, discussed, for example, by Bukht and Heeks (2017). As a reference, the definition used by OECD is stated (Dahlman et al. 2016, p 11) as

> *"The digital economy is the amalgamation of several general purpose technologies (GPTs) and the range of economic and social activities carried out by people over the Internet and related technologies. It encompasses the physical infrastructure that digital technologies are based on (broadband lines, routers), the devices that are used for access (computers, smartphones), the applications they power (Google, Salesforce) and the functionality they provide (IoT, data analytics, cloud computing)"*

The topology of digital economy formation is illustrated in Figure 13, as described by Heeks (2008).

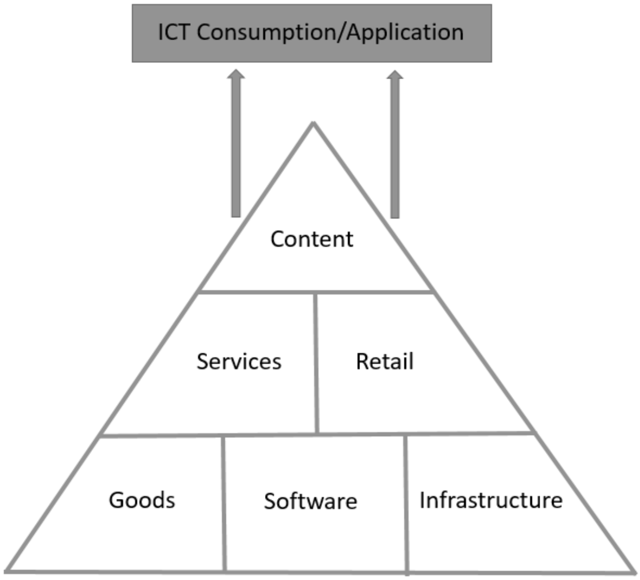

Figure 13. The topology of ICT-based enterprises within information and communication technology sub-sectors (reproduced from Heeks 2008).

The World Economic Forum calls the rise of the digital economy the fourth industrial revolution, and estimates that, within a decade, about 60–70% of new value will be based on data-driven digitally enabled networks and platforms (Smith et al. 2018). This has a considerable impact on labor markets (Dengler and Matthes 2018), education (Gleason 2018), and all aspects of human life.

Technological development seems to be continuous and accelerating. This technological development drives the digital economy and how people work and live in societies. This brief review examined the quantitative change. Qualitative changes such as more intelligent technology or social media use are harder to detect, but have even greater impact by challenging traditional social structures.

Therefore, it is appropriate to continually revise concepts and ideas used in the 20th century and update them if needed. In Study IV, I examined some aspects of this change and found that the concept of computational thinking may benefit from such a revision. Issues to be examined could include intelligent development tools such as utilizing artificial intelligence, social aspects of programming, functional programming paradigms, non-textuality, and multimodality.

6.2.2 Multimodality and Non-Textual Programming Languages

In this research, I proposed nonlinear storytelling as a method of writing simple computer programs. As in Study II, the most natural programs for the storytelling method are finite state machine programs and programs utilizing interactive multiple-choice decisions.

Such domain specificity is not necessarily a bad thing. For example, Scratch is most natural with programming simple sprite-based animations or games. By natural, I mean that writing this kind of application is efficient using the language in question. A task that is not natural for a language is such that it would be significantly easier with some other language. In particular, being natural for some tasks does not limit a language's use to only those tasks.

Let us review some programming languages that might be considered multimodal or non-textual in the sense that was implicated in Study IV.

6.2.2.1 Cellular automaton: Conway's Game of Life

A cellular automaton is a system that consists of cells that have a finite number of states. Cells are connected, for example in a grid. After a specified initial condition, a new state for each cell is calculated as a function of its earlier state and its neighboring cells.

The most well-known cellular automaton is John Conway's The Game of Life (Gardner 1970). It consists of a two-dimensional grid. Each cell can be alive or dead, so they have two states. Each cell has eight neighbors. The rules for generating an update is as follows:

1) Any live cell with fewer than two live neighbors dies, as if by underpopulation.
2) Any live cell with two or three live neighbors lives on to the next generation.
3) Any live cell with more than three live neighbors dies, as if by overpopulation.
4) Any dead cell with exactly three live neighbors becomes a live cell, as if by reproduction.

An example of this is presented in Figure 14. The initial condition of the automaton is three vertical live cells. Following the rules, two live cells die because they have only a single live neighbor. Two dead cells become live cells because they have exactly three live neighbors. With this automata, Step 3 is identical to Step 1. Therefore three vertical cells and three horizontal cells alternate ad infinitum.

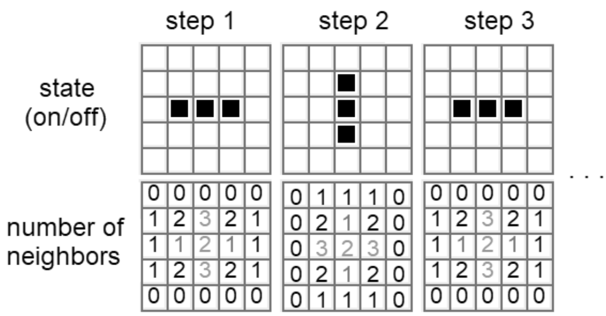

Figure 14. An example of Conway's The Game of Life. On the top row, living cells are presented with black squares and dead cells are empty. For each cell number of live neighbors is counted. This count is presented on the bottom row. Cells that stay alive or will become alive are colored green. Dying cells are colored red. The state of Step 2 is generated from state Step 1. Step 3 is generated from Step 2. Generating further generations would continue without limit.

Conway's The Game of Life is not a practical programming language, but it has been shown that it is Turing complete (Chapman 2002). Its advantage is that it is easy to execute without a computer, using just a pen and paper. It also emphasizes the mechanical nature of computing by exposing a very minimal method of computation. Other cellular automata and their practical and philosophical implications have been studied by Wolfram (2002).

6.2.2.2 Esoteric languages: Piet

Esoteric programming languages are languages that are not intended for practical programming but are designed to test the boundaries of computer programming language design. Here I present the Piet language. It is designed so that programs written with it resemble abstract paintings. The rules of the language can be found on its homepage[24]. Some example codes are presented in Figure 15.

Figure 15. Three computer programs are written in the Piet language. a) Hello world program prints "Hello world!", b) program to solve the problem of Tower of Hanoi by Sylvain Tintillier, c) program to calculate an estimate for the value of pi, by Richard Mitton.

Piet demonstrates that an image can be an executable computer program. Programming with Piet can be considered an activity similar to solving mathematical puzzles. It, as well as many other esoteric languages, has entertaining qualities which may attract some people. While it is not a practical language, it may develop an understanding of complex discreet systems and abstract thinking as a problem-solving activity.

6.2.2.3 Live music coding: Sonic Pi

Sonic Pi is a live coding language to enable the generation of live music performances. It is very specialized for generating live music. It was originally designed as an educational tool to engage schoolchildren in the UK's then newly drafted computing curriculum (Aaron 2016). Its idea is to illustrate core aspects of computing such as sequences, iteration, and functions. It is also designed to expose students to functional programming concepts (Aaron and Blackwell 2013).

Aaron (2016) observed that students understood the coding environment more as a musical instrument than a programming language. This led to its development in a more instrumental direction. Sinclair (2014) has compared Sonic Pi to another Ruby implementation, Kids Ruby, with first-year undergraduate art students. The students in the study favored the Sonic Pi language.

There are also several other languages and systems for producing music, such as Pure Data, UPIC, HighC, and Vuzik, which allows the creation of music through painting (Pon

[24] Piet homepage: https://www.dangermouse.net/esoteric/piet.html

et al. 2011). These examples introduce a multimodal approach to computer programming. While many of the traditional computational thinking skills are still required with these languages, they provide an engagement with programming experience that is not attainable using traditional textual programming languages.

As with nonlinear storytelling, these languages may not be suitable for all students. But they have potential for those students who have trouble engaging with regular programming languages. Storytelling activity with a natural language approach, different sensory modalities, or a student's personal attraction to some special aspect such as the esthetics of Piet language may provide a starting point for learning programming and computational thinking skills.

6.3 Nonlinear storytelling in education

The context of this research is how nonlinear storytelling can be utilized in education. Some groundwork has been done: in Study I, an experimental game development project with nonlinear storytelling was reported. In Study III, available software was reviewed, and stories created by students were analyzed. In Study II we provided evidence that nonlinear storytelling is a feasible method to teach simple programming tasks. Finally, nonlinear storytelling differs from traditional computer programming and the corresponding idea of what computational thinking is. Study IV explored how computational thinking ideas can be adapted to a more general context.

There is no general guide on how to evaluate new teaching methods. Sometimes an old method is applied in a new context or a new technology is used to replace an old technology while keeping the pedagogical content. In such gradual advancement, the old system can sometimes be used as a reference point or a benchmark for evaluation. With nonlinear storytelling, this is not available.

In earlier research, nonlinear storytelling has mostly been used as a novelty item. There are only separate studies but no continuation. The studies are also mostly software-centric rather than methodological. For example, the only comparative study by Liu et al. (2010) is the only study using that piece of software. The software is also very specific and not available anymore. This seems to be a common trend with research software. Much of the research describes research projects where software is short-lived and often not maintained after the project ends, since computer software is hard to maintain and computer technology evolves quickly.

This is why I have adopted a more basic research-oriented methodology and concentrated on producing generalizable results that might be useful in future research.

6.3.1 The novelty value of storytelling, gamification, and robotics

Both students and educators often express interest in including storytelling, games, and robots in teaching. For students any activity that might differ from regular learning activities can be experienced as a positive change. Teachers, on the other hand, may have heard about these methods and become interested in trying them. This interest can be a good source of motivation for the students, and it can provide high participation and focus on learning tasks.

But utilizing novelty value can also be seen as a risk. Students may have high expectations for these activities. For example, gamification may be expected to include excitement. Students may have mental images of unrealistic capabilities for robots. In storytelling, high expectations can be met with difficulties in coming up with stories and the task of generating a detailed story. Good preparation for the teacher is required to cope with these pitfalls.

Han et al. (2005) emphasize high-quality learning content with various options for learners throughout the learning process with robots. They also advise continuous supervision and supporting the learner's reflective activities as well as providing an environment to support learners' various needs.

This is in line with our experience that study materials should be planned carefully. Often, more is less. For example, storytelling can be used in any kind of story and game. Popular options are dialogs, multiple-choice adventure games, quizzes, and humoristic stories and games. Any one of these can provide a good activity in learning the basics of nonlinear storytelling, but providing too many options can be overwhelming. Therefore we usually pick just one category of stories at first. For an advanced group of students, it can be an adventure game, but usually something simpler, such as a quiz or a dialog, is suitable. After mastering one activity, the students can then move on to exploring others.

In our training, we usually had from three to five study sessions. We originally planned to introduce robots in addition to storytelling. But after some experience, the robots were left out to leave more room for storytelling activities. After the first session of the introduction, it was good to have more time to go deeper into the subject.

This also saves the novelty value of robots to be used another time. There is some evidence that robots are not only novelty toys, and that interaction level with the robots continues after introduction (Standen 2014, Fridin 2014), but this only comes from limited-duration studies. We might expect that students in the future may have educational robots introduced many times even during their elementary school career. They may also have exposure to robots at home, for example, robot vacuum cleaners, robot lawnmowers, and entertainment or educational robot toys.

6.3.2 Nonlinear storytelling for women and girls

Girls and women are known to have different media preferences compared to boys and men. Yee (2017a) studied the gaming preferences of 250,000 gamers. According to the study, for male gamers, two primary motivations are competition and destruction. For female gamers, the primary motivations are completion and fantasy. The story is the 5th most important motivation for women and only the 8th most important for men.

According to another study (Yee 2017b), the most common games for women, after casual puzzles, match-3 games, and family/farm simulations, are atmospheric explorations and interactive drama. The least popular games among women are sports, tactical shooters, and racing games. Fromme (2003) gives similar results for game genres but also different media use styles. For example, 49 percent of girls (12 to 19 yo youth in Germany) reported writing texts as a computer activity performed at least once a week. For boys, such activity was reported by 36 percent.

Girls are often a special target group for storytelling applications. Tran (2016) reports a study titled "Her story was complex: A Twine workshop for ten-to twelve-year-old girls." Hess (2013) studied digital storytelling as a tool for empowering young women's feminist and womanist faith formation. Hlalele and Brexa (2015) used digital storytelling as a method for empowering female university students in South Africa. Şimşek (2012) reports a project, "Digital Stories from Amargi Women", where digital storytelling is used to promote agency and self-expression with feminist activists in Turkey. These are just some examples of storytelling applications specifically targeted at women.

So-called STEM (Science, Technology, Engineering, and Mathematics) field occupations are particularly underrepresented by women. According to PISA 2015 survey results, 4.8% of boys expected to work in the ICT sector in the future but only 0.4% of girls (OECD 2015). A report by Google (2014) investigated women's perception of computer science. The top three adjectives they associated with computer science were "boring", "hard", and "difficult".

The traditional approach to computer science and computational thinking is known to be technical and perhaps distant from the social aspects of human life. This can be one of the reasons for the negative views that women have of these concepts. Even with tools like Scratch and Alice, characters and animations are rather superficial and do not hide the technical aspects.

Very little technical knowledge is needed to get started with nonlinear storytelling. The story is the main role of the process and it can be used as a tool for expression with very few technical constraints. Therefore, it is possible that nonlinear storytelling could provide a method to encourage women and girls to adopt a more active creative role in the use of informational technology. At the same time, this activity can develop their computational thinking skills and provide better readiness for future studies where adopting information technology skills is needed.

7 Future work

This research has built some foundation for using nonlinear storytelling in developing computational thinking skills. Here I outline some possible future research.

7.1 Understanding nonlinear information technology tasks

In our research, I noticed that there is great variance in how quickly and how confidently students apprehended the idea of storyline branching and how it is related to new possibilities in storytelling such as interaction and gamification.

During the research, we experimented with the storytelling method with different focus groups to develop a better understanding of its challenges and possibilities. Generally, many younger students (under 30 years old) seemed to feel that nonlinearity was a very natural concept. Correspondingly, many older students (over 30 years old) seemed to have difficulties with the concept of nonlinearity.

Our first hypothesis was that younger students might be more familiar with information technology since most of our older students were selected to participate in the study according to their need to learn the basics of information technology. But one of our studies (Letonsaari and Selin 2019) included older students who already had a long work history with computers. Different age groups were not quantitatively or formally compared, but our experience was that younger students adapted to the concept of nonlinearity more quickly here as well.

Several factors may explain this. For example, younger people can generally be faster at adopting new ideas, or they might have a different experience with information technology, involving computer games and social media instead of the office software used in working life.

Nevertheless, I feel that this issue deserves further investigation. Generally, the idea is that some people seem to prefer clear instructions for everything, while some are more aware of the nonlinear nature of tasks.

For example, when writing an email, it is possible to add another recipient and attach a new file to the email. Some people prefer to have stepwise instructions, for example

1) Select a new email,
2) add recipient,
3) write a title,
4) write the message, and
5) send the email.

When instructed to add an attachment, they wish to know the correct step to add an attachment (for example, after step 4). They might discover a more flexible method of working with experience, but even then, their mental representation of the task might be that it is possible to "go back" to the previous step. For other people, it is more natural to understand that the process of starting a new email and sending it is actually nonlinear, and the user can freely switch between selecting recipients, writing title or text, and adding attachments.

Other examples include, for example, understanding multitasking operating systems (several programs running simultaneously) and file systems (where files are not in a list but in a hierarchical directory and lists can be sorted according to file properties such as filename, file type, or size). Many more examples exist.

This idea is also a very fundamental skill in the efficient learning of information technology. Students can memorize a series of actions, or they can learn the process which also gives signification for the process.

For this topic, several research topics are possible:

- Can we identify how people understand the ways in which information technology processes work? Are there variables that explain their thinking, such as their history of computer use, gaming history, age, education, etc.? If we can identify people who have specific difficulties, we may be able to offer specialized training to improve their understanding of information technology processes.
- Can we explain how these different ways of understanding affect learning? Several theoretical tools and frameworks are available to analyze this question. The cognitive dimensions framework (Green 1989) provides methods of using Cognitive Load Theory in analyzing information technology tasks. It is, for example, widely used in computer language analysis (Green 1996). Using the instructional design based on this theory, cognitive load in learning tasks could be evaluated (Plass et al. 2010). Alternatively, there are models for the comprehension of computer programs (such as Brooks 1983) and cognitive tutors (Koedinger et al. 2006) that might be applicable in this task.
- If we can identify people whose understanding of information technology is not optimal for learning more, is nonlinear storytelling a feasible method to improve their understanding? As mentioned earlier, nonlinear storytelling can be implemented with a very low learning threshold and tools that require very little previous skills or knowledge. Liu et al. (2010) used a video-based nonlinear storytelling system successfully with children with the mean age of nine years. The Twine storytelling software has been used with children as young as ten years old (Tran 2016; Davis 2013). A quantitative study should be designed and carried out to examine this possibility.

Currently, these research plans are just plans and suggestions for further research.

7.2 Validation of the method

In this research, I did not conduct studies to directly validate nonlinear storytelling as a teaching method. There are several reasons for this. Firstly, the method of nonlinear storytelling was a novel topic. There were no ready-to-use instructions and materials. Therefore basic research was needed to evaluate its general feasibility as an educational method. This research took us to the basics of information technology in Study II, to review and categorize tools in Study III, and to the fundamental question of whether the concept of computational thinking can be used in the context of nonlinear stories in its present form in Study IV.

If we wish to validate the nonlinear storytelling method against computational thinking skills tests, there is no single authoritative test available. Román-González et al. (2019) provide a review and classification of computational thinking tests. Their classification includes:

- CT diagnostic tools aimed at measuring the CT aptitudinal level of the subject;
- CT summative tools to evaluate if the learner has achieved sufficient content knowledge after receiving some instruction in CT skills;
- CT formative–iterative tools aimed at providing feedback to the learner, usually automatically, in order to develop and improve their CT skills;
- CT data-mining tools retrieve and record the learner activity in real-time providing data and learning analytics from which cognitive processes of the subject can be inferred;
- CT skill transfer tools' objective is to assess to what extent the students are able to transfer their CT skills onto different kinds of problems, contexts, and situations;
- CT perceptions–attitudes scales aimed at assessing the perceptions and attitudes of the subjects not only about CT, but about related issues such as computers, computer science, computer programming, or digital literacy; and
- CT vocabulary assessment intends to measure several elements and dimensions of CT verbally expressed by the subjects.

They recognize several tests in each category. They also analyze these along three dimensions of computational thinking: computational concepts, computational practices, and computational perspectives.

Most importantly, they recommend certain types of tests for several research designs. In their classification, a quasi-experimental research design aims to verify that a certain intervention is effective in developing the computational thinking skills of the students. This research design has a treatment and control group, each of which is administered pre-tests and post-tests.

For summative research, they suggest four tools:

- Meerbaum-Salant el. al (2013) assessment in the Scratch context;
- Quizly1 (Maiorana et al. 2015) for assessing content knowledge in the context of App Inventor;
- Fairy Assessment (Werner et al. 2012), a performance-based tool that runs in the Alice environment; and
- summative tools for measuring understanding of computational concepts relevant to Israel's computing curriculum (Zur-Bargury et al. 2013).

These are all application-dependent to some extent and probably need some adapting to be used with nonlinear storytelling.

For effective validation, more experience in the practical use of nonlinear storytelling is needed. Assessment tools are dependent on teaching content and learning objectives, and the method should be tested in a classroom environment with relevant target groups to form a preliminary understanding of the learning process. Experience is also needed for the teacher so that they can familiarize themselves with the method and prepare for the task.

To conclude, I have conducted some basic research with my co-workers to evaluate the general feasibility of nonlinear storytelling in education. We have also surveyed tools to use in educational storytelling and evaluated how the concept of computational thinking applies to nonlinear storytelling. As this is a novel research topic, more experience and research knowledge are needed to enable a well-designed validation.

References

Aaron, S. and Blackwell, A.F., 2013, September. From sonic Pi to overtone: creative musical experiences with domain-specific and functional languages. In *Proceedings of the first ACM SIGPLAN workshop on Functional art, music, modeling & design* (pp. 35-46).

Aaron, S., 2016. Sonic Pi–performance in education, technology and art. *International Journal of Performance Arts and Digital Media*, 12(2), pp.171-178.

Aarseth, E.J., 1997. *Cybertext: Perspectives on ergodic literature*. JHU Press.

Abt, C.C., 1970. *Serious games*. The Viking Press.

Adams, E., 2009. *Fundamentals of game design*. New Riders Publishing.

Aho, A.V., 2012. Computation and computational thinking. *The Computer Journal*, 55(7), pp.832-835.

All, A., Van Looy, J. and Castellar, E.P.N., 2013. An evaluation of the added value of co-design in the development of an educational game for road safety. *International Journal of Game-Based Learning* (IJGBL), 3(1), pp.1-17.

Alismail, H.A., 2015. Integrate digital storytelling in education. *Journal of Education and Practice*, 6(9), pp.126-129.

Arnau, E., Estany, A., Gonzalez del Solar, R. and Sturm, T., 2014. The extended cognition thesis: Its significance for the philosophy of (cognitive) science. *Philosophical Psychology*, 27(1), pp.1-18.

Aumarm, Robert J., 1959. Acceptable points in general cooperative n-person games. In *Contributions to the Theory of Games* (AM-40), 4:pp. 287–324.

Balanskat, A. and Engelhardt, K., 2014. *Computing our future: Computer programming and coding-Priorities, school curricula and initiatives across Europe*. European Schoolnet.

Baranowski, T., Buday, R., Thompson, D.I. and Baranowski, J., 2008. Playing for real: video games and stories for health-related behavior change. *American journal of preventive medicine*, 34(1), pp.74-82.

Barefoot, K., Curtis, D., Jolliff, W., Nicholson, J.R. and Omohundro, R., 2018. Defining and measuring the digital economy. *US Department of Commerce Bureau of Economic Analysis*, Washington, DC, 15.

Barrett, E. (Ed.), 1988. *Text, ConText, and HyperText: Writing with and for the Computer*. Cambridge, MA: Mit Press.

Basawapatna, A., Koh, K.H., Repenning, A., Webb, D.C. and Marshall, K.S., 2011, March. Recognizing computational thinking patterns. In *Proceedings of the 42nd ACM technical symposium on Computer science education* (pp. 245-250).

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R. and Kern, J., 2001.

The agile manifesto.

Begosso, L.C., Begosso, L.R., Gonçalves, E.M. and Gonçalves, J.R., 2012, October. An approach for teaching algorithms and computer programming using Greenfoot and Python. In *2012 Frontiers in Education Conference Proceedings* (pp. 1-6). IEEE.

Bell, T., Witten, I. and Fellows, M., 2015. *CS Unplugged: An enrichment and extension programme for primary-aged students*.

Berland, M. and Wilensky, U., 2015. Comparing virtual and physical robotics environments for supporting complex systems and computational thinking. *Journal of Science Education and Technology,* 24(5), pp.628-647.

Biasutti, M. and Heba, E.D., 2012. Using Wiki in teacher education: Impact on knowledge management processes and student satisfaction. *Computers & Education*, 59(3), pp.861-872.

Boltman, A. and Druin, A., 2003. *Children's storytelling technologies: differences in elaboration and recall.*

Bossavit, B. and Parsons, S., 2016, May. "This is how I want to learn" High Functioning Autistic Teens Co-Designing a Serious Game. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (pp. 1294-1299).

Bromme, R., & Stahl, E., 2005. Is a hypertext a book or a space? The impact of different introductory metaphors on hypertext construction. *Computers & Education*, 44(2), 115-133.

Brooks, R., 1983. Towards a theory of the comprehension of computer programs. *International journal of man-machine studies,* 18(6), pp.543-554.

Bukht, R. and Heeks, R., 2017. Defining, conceptualising and measuring the digital economy. *Development Informatics working paper*, (68).

Bush, V. and Think, A.W.M., 1945. The atlantic monthly. As we may think, 176(1), pp.101-108.

Caine, R., G. Caine, C. McClintic and K. Klimic, 2005. *Brain/Mind Learning Principles in Action*. Morgan Press, Thousand Oaks.

Chapman, P., 2002. *Life universal computer*.

Clark, A. and Chalmers, D., 1998. The extended mind. *analysis*, 58(1), pp.7-19.

Clarke, R. and Capes, J., 2000. *Incompatibilist (nondeterministic) theories of free will*.

Cobb, P., Wood, T., Yackel, E., & McNeal, B. (1992). Characteristics of classroom mathematics traditions: An interactional analysis. *American Educational Research Journal*, 29, 573-604

Cole, M., 2009. Using Wiki technology to support student engagement: Lessons from the trenches. *Computers & education*, 52(1), pp.141-146.

Collier, G. H., 1987. Thoth-II: Hypertext with explicit semantics. In *Proceedings of the ACM conference on Hypertext* (pp. 269-289). ACM.

Cosma, G., Shopland, N., Battersby, S., Seymour-Smith, S., Pockley, A.G., Archer, M., Thompson, R., Khan, M. and Brown, D., 2015, October. Co-design of a prostate cancer serious game for African Caribbean men. In *2015 International Conference on Interactive Technologies and Games* (pp. 71-75). IEEE.

Coultrip, R., Granger, R. and Lynch, G., 1992. A cortical model of winner-take-all competition via lateral inhibition. *Neural networks*, 5(1), pp.47-54.

Csikszentmihalyi, M., 1991. *Flow: The psychology of optimal experience* (Vol. 41). New York: HarperPerennial.

Dahlman, C., Mealy, S. & Wermelinger, M., 2016. *Harnessing the Digital Economy for Developing Countries*, OECD, Paris.

Davis, G., 2013. Teaching Twine to Ten Year Olds >> *re/Action | Zine*. [ONLINE] Available at: http://www.reactionzine.com/teaching-twine-to-ten-year-olds/. [Accessed 10 April 2020].

Delany, P., and Gilbert, J. K., 1991. HyperCard stacks for Fielding's Joseph Andrews: issues of design and context. In *Hypermedia and literary studies* (pp. 287-297). MIT Press.

Dengler, K. and Matthes, B., 2018. The impacts of digital transformation on the labour market: Substitution potentials of occupations in Germany. *Technological Forecasting and Social Change*, 137, pp.304-316.

DeStefano, D. and LeFevre, J.A., 2007. Cognitive load in hypertext reading: A review. *Computers in human behavior*, 23(3), pp.1616-1641.

Dillon, A., and Gabbard, R., 1998. Hypermedia as an educational technology: A review of the quantitative research literature on learner comprehension, control, and style. *Review of educational research*, 68(3), 322-349.

Djaouti, D., Alvarez, J., Jessel, J.P. and Rampnoux, O., 2011. Origins of serious games. In *Serious games and edutainment applications* (pp. 25-43). Springer, London.

Dijkstra, E.W., 1974. Programming as a discipline of mathematical nature. *The American Mathematical Monthly*, 81(6), pp.608-612.

El-Nasr, M.S. and Smith, B.K., 2006. Learning through game modding. *Computers in Entertainment* (CIE), 4(1), p.7.

Flamm, K. 2002, "Microprocessors and computers: The Phenomenom of Price Declines" in *Measuring and Sustaining the New Economy: Report of a Workshop*. National Academies Press. pp. 83-88

Ford, N., and Chen, S. Y., 2000. Individual differences, hypermedia navigation, and learning: an empirical study. *Journal of educational multimedia and hypermedia*, 9(4), 281-311.

Fowler, A. and Cusack, B., 2011, June. Kodu game lab: improving the motivation for learning programming concepts. In *Proceedings of the 6th International Conference on Foundations of Digital Games* (pp. 238-240). ACM.

Fowler, M., 2003. *UmlMode*. [ONLINE] Available at: https://martinfowler.com/bliki/UmlMode.html [Accessed 10 April 2020]

Fowler, M., 2004. *UML distilled: a brief guide to the standard object modeling language*. Addison-Wesley Professional.

Fraser, S. and T. Greenhalgh, 2001. Coping with complexity: Educating for capability. *BMJ*, 323: 799-803.

Fridin, M., 2014. Storytelling by a kindergarten social assistive robot: A tool for constructive learning in preschool education. *Computers & education*, 70, pp.53-64.

Fromme, J., 2003. Computer games as a part of children's culture. *Game studies*, 3(1), pp.49-62.

Gardner, M., 1970. The Fantastic Combinations of Jhon Conway's New Solitaire Game'Life. *Sc. Am.*, 223, pp.20-123.

Gleason, N.W., 2018. *Higher education in the era of the fourth industrial revolution* (p. 229). Springer Nature.

Google CS Ed Research group, 2014. *Women who choose computer science–what really matters: The critical role of encouragement and exposure*. Technical report, Google.

Green, T. R. G. 1989. Cognitive dimensions of notations. In A. Sutcliffe and L. Macaulay (Eds.) *People and Computers* V. Cambridge, UK: Cambridge University Press, pp 443-460.

Green, T.R.G., Petre, M., 1996. Usability analysis of visual programming environments: a 'cognitive dimensions' framework. *Journal of Visual Languages & Computing*, 7(2), pp.131-174.

Grover, S. and Pea, R., 2013. Computational thinking in K–12: A review of the state of the field. *Educational researcher*, 42(1), pp.38-43.

Guimaraes, Fernando (edt.) 2011. *Research - Anyone Can Do It*. Mainz: PediaPress.

Gundlach, R., 1985. *Geschichtsdenken und Geschichtsbild im pharaonischen Ägypten*. Wiss. Verlag-Ges..

Günbatar, M.S., 2019. Computational thinking within the context of professional life: Change in CT skill from the viewpoint of teachers. *Education and Information Technologies*, 24(5), pp.2629-2652.

Göbel, S., Salvatore, L. and Konrad, R., 2008, November. StoryTec: A digital storytelling platform for the authoring and experiencing of interactive and non-linear stories. In *2008 International Conference on Automated Solutions for Cross Media Content and Multi-Channel Distribution* (pp. 103-110). IEEE.

Han, J., Jo, M., Park, S. and Kim, S., 2005, August. The educational use of home robots for children. In ROMAN 2005. *IEEE International Workshop on Robot and Human Interactive Communication*, 2005. (pp. 378-383). IEEE.

Heeks, R, 2008. Researching ICT-Based Enterprise in Developing Countries: Analytical Tools and Models. *Development Informatics*, Working Paper no. 30.

Hevner, A., March, S.T., Park, J. and Ram, S., 2004. Design science research in information systems. *MIS quarterly*, 28(1), pp.75-105.

Hess, Mary E, 2013. Digital Storytelling: Empowering Feminist and Womanist Faith Formation with Young Women. In *Media, Religion, and Gender: Key Issues and New Challenges*, edited by Mia Lövheim, 169–82. Media, Religion and Culture Series. London: Routledge.

Hlalele, D. and Brexa, J., 2015. Challenging the narrative of gender socialisation: Digital storytelling as an engaged methodology for the empowerment of girls and young women. *Agenda*, 29(3), pp.79-88.

Holton, D.L., 2010, April. Constructivism+ embodied cognition= enactivism: theoretical and practical implications for conceptual change. In *AERA 2010 Conference*.

Jacobson, M.J., & Spiro, R.J., 1995. Hypertext learning environments, cognitive flexibility, and the transfer of complex knowledge: An empirical investigation. *Journal of Educational Computing Research*, 12, 301-333.

Jenkins, H., 2006. *Confronting the challenges of participatory culture: Media education for the 21st century. An occasional paper on digital media and learning*. John D. and Catherine T. MacArthur Foundation.

Jonassen, D.H., 1988. Designing structured hypertext, and structuring access to hypertext. *Educational Technology*, 28, 13-16.

Jonassen, D. H., 1989. Hypertext/hypermedia. *Educational technology*.

Jonassen, D.H., 1991. Evaluating constructivistic learning. *Educational Technology*, 39, 28-33.

Juul, J., 2002, June. The Open and the Closed: Games of Emergence and Games of Progression. In *CGDC Conf.*

Kahn, K., 1996. ToonTalkTM—an animated programming environment for children. *Journal of Visual Languages & Computing*, 7(2), pp.197-217.

Kahn, K., Silverman, B., Minsky, M., Papert, A., Harvey, B., Miller, M., Solomon, C. and Lieberman, H., 2020. History of Logo. *Proceedings of the ACM on Programming Languages*.

Katz, D. (n.d.) Person - Webster, Doris - Demian's Gamebook Web Page. [ONLINE] *Gamebooks.org*. Available at: https://gamebooks.org/Person/4354/Show [Accessed 22 Jun. 2019].

Katz, D. (2013) Marcos Benevides Interview -- January, 2013. [ONLINE] *Gamebooks.org*. Available at: https://gamebooks.org/marcos_benevides.htm [Accessed 28 Feb. 2020].

Kickmeier-Rust, M.D., Mattheiss, E., Steiner, C. and Albert, D., 2011. A psycho-pedagogical framework for multi-adaptive educational games. International *Journal of Game-Based Learning* (IJGBL), 1(1), pp.45-58.

Kim, B., Kim, T. and Kim, J., 2013. and-Pencil Programming Strategy toward Computational Thinking for Non-Majors: Design Your Solution. *Journal of Educational Computing Research*, 49(4), pp.437-459.

Knuth, D.E., 1972. Ancient babylonian algorithms. *Commun. ACM* 15(7), 671–677

Knuth, D.E., 1974. Computer science and its relation to mathematics. *The American Mathematical Monthly*, 81(4), pp.323-343.

Koedinger, K.R. and Corbett, A., 2006. *Cognitive tutors: Technology bringing learning sciences to the classroom*. na.

Koomey, J., Berard, S., Sanchez, M. and Wong, H., 2010. Implications of historical trends in the electrical efficiency of computing. *IEEE Annals of the History of Computing*, 33(3), pp.46-54.

Kurzweil, R., 2005. *The singularity is near: When humans transcend biology*. Penguin.

Laanti, M., Similä, J. and Abrahamsson, P., 2013, June. Definitions of agile software development and agility. In *European Conference on Software Process Improvement* (pp. 247-258). Springer, Berlin, Heidelberg.

Lavigne, H., Orr, J. and Wolsky, M., 2018, March. Exploring Computational Thinking in Preschool Math Learning Environments. In *Society for Information Technology & Teacher Education International Conference* (pp. 38-43). Association for the Advancement of Computing in Education (AACE).

Letonsaari, M. and Selin, J., 2018, April. Social Media Integration with Nonlinear Educational Storytelling Application. In *2018 17th International Conference on Information Technology Based Higher Education and Training* (ITHET) (pp. 1-5). IEEE.

Letonsaari, M., Selin J., 2019. Nonlinear Storytelling Method for Interdisciplinary and Cross Expertise-Level Communication of Knowledge. Virendra Singh Nirban, Tanu Shukla, Anupam Yadav, Rajneesh Choubisa, *Interdisciplinarity: Vignettes on Contemporary Knowledge, Education & Research*, Pilani, India, 8-9 February

Leutenegger, S. and Edgington, J., 2007, March. A games first approach to teaching introductory programming. In *ACM SiGCSE Bulletin* (Vol. 39, No. 1, pp. 115-118). ACM.

Leyton-Brown, K. and Shoham, Y., 2008. Essentials of game theory: A concise multidisciplinary introduction. *Synthesis lectures on artificial intelligence and machine learning*, 2(1), pp.1-88.

Lin, D. Y. M., 2003. Hypertext for the aged: effects of text topologies. *Computers in Human Behavior*, 19(2), 201-209.

Liu, C.C., Liu, K.P., Chen, G.D. and Liu, B.J., 2010. Children's collaborative storytelling with linear and nonlinear approaches. *Procedia-Social and Behavioral Sciences*, 2(2), pp.4787-4792.

Lodge, S., 2007. Chooseco embarks on its own adventure. *Publishers Weekly*, 17

Lorenz, E., 2000. The butterfly effect. *World Scientific Series on Nonlinear Science Series A*, 39, pp.91-94.

Maiorana, F., Giordano, D. and Morelli, R., 2015, October. Quizly: A live coding assessment platform for App Inventor. In *2015 IEEE Blocks and Beyond Workshop* (Blocks and Beyond) (pp. 25-30). IEEE.

Maleki, A. and Sajjadi, S., 2012. The role of non-linear methods in teaching english for medicine: Example of storytelling. *Journal of Applied Sciences*, 12(18), pp.1972-1977.

Manning, J., 2004. *The Emblem*. Reaktion Books, United Kingdom

Marchionini, G., 1988. Hypermedia and learning: Freedom and chaos. *Educational technology*, 28(11), pp.8-12.

McDevitt, D., 2010. A Practical Guide to Game Writing. [ONLINE] Available at: https://www.gamasutra.com/view/feature/134542/a_practical_guide_to_game_writing.php?print=1 [Accessed 22 February 2019].

McDrury, J. and M. Alterio, 2003. *Learning Through Storytelling in Higher Education: Using Reflection and Experience to Improve Learning*. Routledge, UK.

Media College. 2017. Linear vs Non Linear Editing. [ONLINE] Available at: http://www.mediacollege.com/video/editing/linear/linear-vs-nonlinear.html [Accessed 8 March 2018].

Meerbaum-Salant, O., Armoni, M. and Ben-Ari, M., 2013. Learning computer science concepts with scratch. *Computer Science Education*, 23(3), pp.239-264.

Mello, R., 2001. The power of storytelling: How oral narrative influences children's relationship in the classroom. *Int. J. Educ. Arts*, Vol. 2.

Melzer, A., Hadley, L. and Herczeg, M., 2005, June. evaluation of a mixed-reality and high interaction media project in the classroom: Strategies and methods. In *EdMedia+ Innovate Learning* (pp. 3984-3991). Association for the Advancement of Computing in Education (AACE).

Michael, D.R. and Chen, S.L., 2005. *Serious games: Games that educate, train, and inform*. Muska & Lipman/Premier-Trade.

Mikhael, J., 2007. *The Philosophy of Bioinformatics: Extended Cognition, Analogies and Mechanisms*. PhD thesis, University of Waterloo, Waterloo.

Miller, J., 2018, February. Benefits of Exposure Programs to K12 Student Interest in Computer Science. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (pp. 1108-1108).

Morgan, K. 2017. The Literary Term 'Nonlinear Narrative'. [ONLINE] Available at: http://penandthepad.com/literary-term-nonlinear-narrative-1816.html [Accessed 3 August 2017]

Moore, G.E., Cramming more components onto integrated circuits, *Electronics*, Vol. 38, No. 8, April 1965.

Nasir, M.N.F.B. and Effendi, W.M.S., 2012. *Non-linear and interactive storytelling for children's book.*

National Research Council, 2010, Report of a Workshop on the Scope and Nature of Computational Thinking, Washington, D.C.: The National Academies Press, p. 59

Newcombe, N.S., 2010. Picture this: increasing Math and Science learning by improving spatial thinking. *Am. Educ.* 34(2), 29

Nielsen, J., 1995. *Multimedia and hypertext: The Internet and beyond*. Morgan Kaufmann.

Notari, M., 2006, August. How to use a Wiki in education: 'Wiki based effective constructive learning'. In *Proceedings of the 2006 international symposium on Wikis* (pp. 131-132).

Nunes, J.M., and Fowell, S., 1996. Hypermedia as an experiential learning tool: A theoretical model. *Information Research*, 2(1).

OECD, 2015. *PISA 2015 Database*, Tables I.3.11a-d.

Opetushallitus, 2016. *Perusopetuksen opetussuunnitelman perusteet 2014*. Helsinki: Opetushallitus. Helsinki: Edita.

Ouahbi, I., Kaddari, F., Darhmaoui, H., Elachqar, A. and Lahmine, S., 2015. Learning basic programming concepts by creating games with scratch programming environment. *Procedia-Social and Behavioral Sciences*, 191, pp.1479-1482.

Ozoran, D., Cagiltay, N. and Topalli, D., 2012. Using scratch in introduction to programming course for engineering students. In *2nd International Engineering Education Conference* (IEEC2012) (Vol. 2, pp. 125-132).

Papadakis, S., Kalogiannakis, M. and Zaranis, N., 2016. Developing fundamental programming concepts and computational thinking with ScratchJr in preschool education: a case study. *International Journal of Mobile Learning and Organisation*, 10(3), pp.187-202.

Papert, S., 1980. *Mindstorms: Children, computers, and powerful ideas*. Basic Books Inc.

Perković, L., Settle, A., Hwang, S. and Jones, J., 2010, June. A framework for computational thinking across the curriculum. In *Proceedings of the fifteenth annual conference on Innovation and technology in computer science education* (pp. 123-127).

Petrides, A.K., 2014. *Plautus between Greek comedy and Atellan farce: Assessments and reassessments.* The Oxford Handbook of Greek and Roman Comedy, pp.424-443.

Plass, J.L., Moreno, R. and Brünken, R., 2010. *Cognitive load theory*. Cambridge University Press.

Plsek, P. and T. Greenhalgh, 2001. The challenge of complexity in health care. *BMJ*, 323: 625-628.

Pon, A., Ichino, J., Sharlin, E. and Eagle, D., 2011. *Vuzik: Music Creation and*

Comprehension through Painting. University of Calgary.
Rettberg, S., 2005. *All together now: Collective knowledge, collective narratives, and architectures of participation*. Digital Arts and Culture.
Repenning, A., Webb, D. and Ioannidou, A., 2010, March. Scalable game design and the development of a checklist for getting computational thinking into public schools. In *Proceedings of the 41st ACM technical symposium on Computer science education* (pp. 265-269).
Ritsema, R. and Karcher, S., 1994. *I Ching*. Rockport, MA: Element Books.
Robin, B., 2006, March. The educational uses of digital storytelling. In *Society for Information Technology & Teacher Education International Conference* (pp. 709-716). Association for the Advancement of Computing in Education (AACE).
Robison, A.J., 2008. The design is the game: Writing games, teaching writing. *Computers and Composition*, 25(3), pp.359-370.
Román-González, M., Moreno-León, J. and Robles, G., 2019. Combining assessment tools for a comprehensive evaluation of computational thinking interventions. In *Computational Thinking Education* (pp. 79-98). Springer, Singapore.
Roschelle, J., 2003. Unlocking the learning value of wireless mobile devices [keynote paper]. *Journal of Computer Assisted Learning*, 19(3), pp.260-272.
Salmerón, L., Strømsø, H.I., Kammerer, Y., Stadtler, M. and van den Broek, P., 2018. *Comprehension processes in digital reading. Learning to read in a digital world*, pp.91-120.
Shneiderman, B., Fischer, G., Czerwinski, M., Resnick, M., Myers, B., Candy, L., Edmonds, E., Eisenberg, M., Giaccardi, E., Hewett, T. and Jennings, P., 2006. Creativity support tools: Report from a US National Science Foundation sponsored workshop. *International Journal of Human-Computer Interaction*, 20(2), pp.61-77.
Shute, V.J., Sun, C. and Asbell-Clarke, J., 2017. Demystifying computational thinking. *Educational Research Review*, 22, pp.142-158.
Şimşek, B., 2012. Enhancing women's participation in Turkey through digital storytelling. *Journal of Cultural Science*, 5(2), pp.28-46.
Sinclair, A., 2014. Educational Programming Languages: The Motivation to Learn with Sonic Pi. In *PPIG* (p. 10).
Smith, J., St Amour, L., O'Halloran, D., 2018. Our Shared Digital Future Building an Inclusive, Trustworthy and Sustainable Digital Society. *Insight Report*, World Economic Forum.
Standen, P., Brown, D., Roscoe, J., Hedgecock, J., Stewart, D., Trigo, M.J.G. and Elgajiji, E., 2014, June. Engaging students with profound and multiple disabilities using humanoid robots. In *International Conference on Universal Access in Human-Computer Interaction* (pp. 419-430). Springer, Cham.

Stephenson, C. and Barr, V., 2011. Defining computational thinking for k-12. *CSTA Voice,* 7(2), pp.3-4.

Susi, T., Johannesson, M. and Backlund, P., 2007. *Serious games: An overview.* Technical Report. University of Skövde, Sweden

Tabet, N., Gedawy, H., Alshikhabobakr, H. and Razak, S., 2016, July. From alice to python. Introducing text-based programming in middle schools. In *Proceedings of the 2016 ACM Conference on innovation and Technology in Computer Science Education* (pp. 124-129).

Tawfik, A.A., Schmidt, M.M. and Msilu, F., 2018. Stories as decision scaffolds: Understanding nonlinear storytelling using case-based reasoning and educational design research. In *Educational Technology and Narrative* (pp. 21-38). Springer, Cham.

Tedre, M. and Denning, P.J., 2016. The long quest for computational thinking. In: *Koli Calling*, pp. 120–129

Tedre, M. and Denning, P.J., 2017. Shifting identities in computing: From a useful tool to a new method and theory of science. In *Informatics in the Future* (pp. 1-16). Springer, Cham.

Tran, K.M., 2016. "Her story was complex": A Twine workshop for ten-to twelve-year-old girls. *E-learning and Digital Media*, 13(5-6), pp.212-226.

Tulving, E., 2002. Episodic memory: From mind to brain. *Annual review of psychology*, 53(1), pp.1-25.

Vaishnavi, V., Kuechler, W., and Petter, S. (Eds.) (2004/19). "Design Science Research in Information Systems" January 20, 2004 (created in 2004 and updated until 2015 by Vaishnavi, V. and Kuechler, W.); last updated (by Vaishnavi, V. and Petter, S.), June 30, 2019. URL: http://www.desrist.org/design-research-in-information-systems/.

Van Geit, K., Cauberghe, V., Hudders, L. and De Veirman, M., 2015, October. Using Games to Raise Awareness. How to Co-Design Serious Mini-Games?. In *European Conference on Games Based Learning* (p. 532). Academic Conferences International Limited.

W3.org. (n.d.). What is Hypertext?. [ONLINE] Available at: https://www.w3.org/WhatIs.html [Accessed 22 Jun. 2019].

Wang, H., 2009. Engaging students with game programming in python. In *Proceedings of the World Congress on Engineering and Computer Science* (Vol. 1).

Werner, L., Denner, J., Campe, S. and Kawamoto, D.C., 2012, February. The fairy performance assessment: measuring computational thinking in middle school. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education* (pp. 215-220).

Wilson, R.A., 2014. Ten questions concerning extended cognition. *Philosophical psychology*, 27(1), pp.19-33.

Wing, J.M., 2006 Computational thinking. *Commun. ACM* 49(3), 33–35

Wing, J., 2011. Research notebook: Computational thinking—What and why. *The link magazine*, 6.

Winkler, T., Herczeg, M. and Goldmann, A., 2006, March. Why and what children learn while creating an interactive, non linear Mixed-Reality-Storytelling-Room. In *Society for Information Technology & Teacher Education International Conference* (pp. 742-749). Association for the Advancement of Computing in Education (AACE).

Wolfram, S., 2002. *A new kind of science* (Vol. 5, p. 130). Champaign, IL: Wolfram media.

Wright, David R., 2005. *Finite state machines*. Carolina State University.

Yee, N., 2017a. 7 things we learned about primary gaming motivations from over 250,000 gamers. *Quantic Foundry*.

Yee, N., 2017b. Beyond 50/50: Breaking down the percentage of female gamers by genre. *Quantic Foundry,* 19, p.2017.

Zausner, T., 2007. Process and meaning: nonlinear dynamics and psychology in visual art. *Nonlinear dynamics, psychology, and life sciences*, 11(1), p.149.

Zur-Bargury, I., Pârv, B. and Lanzberg, D., 2013, July. A nationwide exam as a tool for improving a new curriculum. In *Proceedings of the 18th ACM conference on Innovation and technology in computer science education* (pp. 267-272).

Appendix A.

List of publications including “nonlinear storytelling” or “non-linear storytelling” in the title, excluding patents and citations. Sorted by year (starting from the most recent) and author (alphabetical). Results have been retrieved from Google Scholar on Nov. 9th 2019. Publications with educational topic have been highlighted with grey color.

Cites	Author	Title	Year	Type	Topic
0	KE Assal, E Thomas, A Gabriel…	Viewport-driven DASH media playback for interactive storytelling: a seamless non-linear storyline experience	2019		Multimedia technology
1	M Letonsaari	Nonlinear Storytelling Method and Tools for Low-Threshold Game Development	2019		This reseach
1	AA Tawfik, MM Schmidt, F Msilu	Stories as decision scaffolds: Understanding nonlinear story-telling using case-based rea-soning and educational design research	2018		Education
0	M Ecker, H Rieß, M Schuster, M Uhlig, P Klein	“Journey”–guided, non-linear storytelling supported by eye tracking	2018		Multimedia, Healthcare
0	M Letonsaari, J Selin	Social Media Integration with Nonlinear Educational Story-telling Application	2018		This reseach
0	OI Psomadaki, CA Dimoulas, GM Kalliris…	Technologies of Non Linear Storytelling for the Manage-ment of Cultural Heritage in the Digital City: The Case of Thes-saloniki	2018		Multimedia, Arts
0	P de Lange, P Nicolaescu, J Benscheid…	Collaborative Non-linear Story-telling Around 3D Objects	2018		Multimedia technology
0	F Coyl	Trace Elements: Non-linear multi-narrative storytelling as a means of deconstructing American mythologies	2017		Multimedia, Arts
3	M Kosek, B Kon-iaris, D Sinclair, D	IRIDiuM+: deep media story-telling with non-linear light field	2017		Multimedia technology

	Markova…	video			
3	M Letonsaari, J Selin	Modeling computational algorithms using nonlinear storytelling methods of computer game design	2017		This reseach
0	AK Μουλαδούδη	Designing non-linear storytelling environments for non-developers: The case of the SKG Parks page.	2017		Multimedia technology
3	SR Meyer	Right, Left, High, Low Narrative Strategies for Non–linear Storytelling	2016		Multimedia, Arts
5	J Schumann, T Buttler, S Lukosch	An approach for asynchronous awareness support in collaborative non-linear storytelling	2013	HTML	Multimedia, Arts
2	A Maleki, S Sajjadi	The role of non-linear methods in teaching english for medicine: Example of storytelling	2012	HTML	Education
0	C Bolewski	Viewers' engagement in non-linear storytelling:'Journeys in Travel'.	2012	PDF	Multimedia, Arts
31	AG Bosser, MO Cavazza…	Linear logic for non-linear storytelling	2010		Artificial intelligence
9	CC Liu, KP Liu, GD Chen, BJ Liu	Children's collaborative storytelling with linear and nonlinear approaches	2010	PDF	Education
0	J Schumann	Supporting Workspace Awareness by Visualizing the Story Graph Evolution in Non-Linear Collaborative Storytelling	2010	PDF	Information technology
86	S Göbel, L Salvatore, R Konrad	StoryTec: A digital storytelling platform for the authoring and experiencing of interactive and non-linear stories	2008	PDF	Storytelling techonology, educational
0	A Sehmi	Nonlinear digital storytelling through multiple perspectives.	2006		Arts
3	T Winkler, M Herczeg, A Goldmann	Why and what children learn while creating an interactive, non linear Mixed-Reality-Storytelling-Room	2006	PDF	Education
0	Y Abe	APPLYING NON-LINEAR STORYTELLING FORMS TO	2005	PDF	Multimedia, Arts

		MULTIMEDIA DOCUMENTARIES			
29	BL Massey	Examination of 38 web newspapers shows nonlinear storytelling rare	2004		News
8	DF Abawi, S Reinhold, R Dörner	A Toolkit for authoring non-linear storytelling environments using mixed reality	2004		Multimedia technology
18	O Bangsø, OG Jensen, FV Jensen…	Non-linear interactive storytelling using object-oriented Bayesian networks	2004	PDF	Computer games
12	B Button	Nonlinear editing: storytelling, aesthetics, and craft	2002	BOOK	Arts
10	M Redmond, N Sweeney	Multimedia production: nonlinear storytelling using digital technologies	1997	BOOK	Multimedia technology
0	MNFB Nasir, WMS Effendi	non-linear and interactive storytelling for children's book		DOC	Education
0	R Klamma, Y Cao	Non-linear Digital Storytelling for the Battleship			Multimedia, Arts
0	S Kougioumtzis, NN Karanikolas, T Panayiotopoulos	Augmented reality environments for immersive and non linear virtual storytelling		PDF	Multimedia, Arts

Number or works by Topic:

Topic	**Number of works**
Arts	10
Multimedia	8
Multimedia technology	6
Education	5
News	1
Computer games	1
Information technology	1
Artificial intelligence	1
Healthcare	1
Storytelling technology	1