



# Nonlinear Storytelling Method and Tools for Low-Threshold Game Development

**Mika Letonsaari, Leena Karjalainen and Jukka Selin**

South-Eastern Finland University of Applied Sciences

Email: [mika.letonsaari@xamk.fi](mailto:mika.letonsaari@xamk.fi)

## Abstract

*Nonlinear storytelling is a method in digital storytelling where the storyline is not linear but has branches. This is typical for interactive stories such as computer game plots. In this study we present a review of nonlinear storytelling tools and educational computer game development software. Special attention is paid to usability, software licensing models, and integration with modern game engines. We present a pilot study where nonlinear stories ( $n = 14$ ) were collected during a game development project. Participants were adults with low information technology skills and who were participating in an information technology training course intended for enhancing their employment opportunities. Collected stories are analyzed to show an example of what kind of content can be expected when nonlinear storytelling is used. In particular some common errors and misconceptions in nonlinear storytelling are explored. Usability of the collected stories in the context of game development is examined. These observations and findings can be used to develop storytelling process more meaningful to students and to harness the nonlinear storytelling method more effectively in education and educational game development.*

**Keywords:** digital storytelling, nonlinear storytelling, educational game, storytelling software

## Introduction

Storytelling has become a very popular method of education. In storytelling, the learner creates or tells a story that is related to the subject matter. Learner's personal point of view

©2019 (author name/s). This is an Open Access article distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), allowing third parties to copy and redistribute the material in any medium or format and to remix, transform, and build upon the material for any purpose, even commercially, provided the original work is properly cited and states its license.

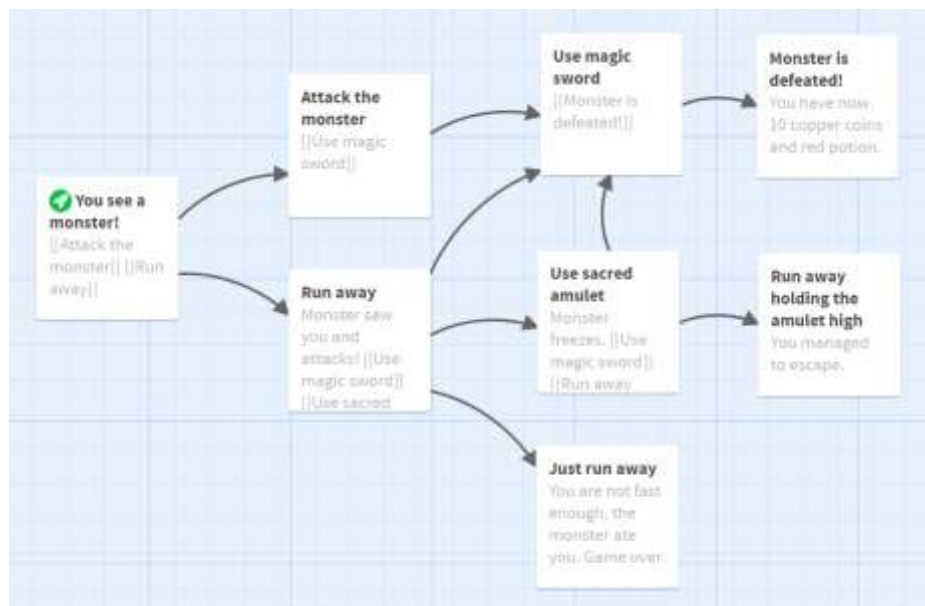
connects the subject matter to a larger context. Storytelling as a learning method has many positive cognitive, social, and emotional benefits including improvement in language skills, concentration, imagination, comprehension, and critical thinking (Boltman 2001, pp. 17-18).

Due to the arrival of computers in school classrooms, digital storytelling has become a widely used and studied form of educational storytelling. Digital storytelling, sometimes called multimedia storytelling, is storytelling using methods of digital multimedia such as images, sound, and video. Often computers are used to compose the story. Digital storytelling is a method to learn digital information processing and multimedia production using modern digital tools. It also provides engagement to the learning process and encourages students in multimedia literacy (Viitanen et al. 2014, Çıralı et al. 2015). Most of the educational storytelling, including digital storytelling, is linear in structure. Stories present only one predetermined sequence of events. The reader of the story has no control of the story. An example of such a story is presented in Figure 1.



**Figure 1.** A story with a linear structure.

Gamified stories or interactive stories provide the user of the story, a reader or a player, a possibility to affect the storyline. Such a storyline has branches and the structure is not linear but nonlinear. An example of a nonlinear gamified story is presented in Figure 2. The story starts from the left. The player is given two options which provide the player with more options. This creates a complex structure for the story with several possible outcomes.



**Figure 2.** A gamified story with nonlinear structure.

There is little research focused on the nonlinear nature of gamified stories in education. Computer game development as a form of nonlinear storytelling is most often just a tool for teaching computer programming (for example Leutenegger et al. 2007, Topalli et al. 2018, Kalelioğlu 2015). Sometimes game development or interactivity of the story is used as a way to engage students but nonlinear nature of stories is not considered as a special feature (Tran 2016).

A notable exception is a study by Liu et al. (2010) where linear and nonlinear storytelling interfaces were compared. They measured co-creative performance in four factors: derivation, remix, ownership, and positive interdependence. In all four factors, the nonlinear group performed superior to the linear group. Researchers reported a positive outcome for the nonlinear storytelling.

In our earlier work, we have shown that nonlinear stories have similarity to computational algorithms (Letonsaari & Selin, 2017a). Based on this, we speculate that practicing nonlinear storytelling may have positive synergy with learning computational thinking skills.

In this article, we review some available types of nonlinear storytelling tools and software. These include general software that is not intended for writing nonlinear stories, as well as specialized nonlinear storytelling software. We present their features regarding the requirements of an educational game development.

We also present an analysis of stories collected in our earlier study to give an example of what kinds of results can be expected when nonlinear stories are collected from novice

computer users.

In the research project an educational computer game was developed by information technology students and professionals. We wanted also to enable the participation of another group of adults who had very low information technology skills. The aim was to engage them in a computer game development, to empower them regarding the use of information technology, and to provide their non-information technology related expertise to the game development process (Letonsaari, Selin & Lampi, 2017b).

In this article the collected stories are analyzed in detail. We are interested in common technical problems that novice users have. Participant in this project had only a short introduction to nonlinear storytelling and limited help from their teacher as they continued their work after school hours. This method of study tries to uncover the critical learning challenges and possible misconceptions in nonlinear storytelling.

## Nonlinear storytelling software

Game plots can be understood as nonlinear stories or hypertext documents. There are many tools to produce such documents with internal hyperlinking. There also exists many types of game development processes from simple experimentations to complex professional projects. Therefore both the requirements of game development processes and the features available in tools must be considered when storytelling tools are chosen.

In our example game development project we chose the following key features for consideration when choosing the solution:

1. easy to use for people with no special information technology skills,
2. possibility to export stories in a simple form, or an integration with advanced game engines, and
3. open source licensing.

The first requirement is given because the idea is to enable a low-threshold participation for everyone. The second requirement is important for us because the stories are used in another professional game engine. It is also generally good to prevent vendor-locking by using software with full export functionality.

The third requirement was chosen because we develop an educational software in an academic research project and it is important to be able to share the results. We also wanted to use external open source licensed resources and therefore open source licensing was a natural choice to avoid licensing problems and enable wide adoption.

In the following, some of the basic types of nonlinear storytelling software are introduced. Different types of hyperlinking methods are explained and the software's selected key features regarding game development are discussed.

## Text editing

The simplest way to write nonlinear stories is to add references to a text file for internal linking. For this purpose, any text editor can be used. Notable examples of texts produced in this manner are encyclopedias with internal linking. There also exist gamified storybooks, for example, Choose Your Own Adventure book series (Friedman, 1995) which have numbered chapters. At the end of each chapter, there are usually choices given to the reader. For example, the text might say *“If you want to fight, jump to chapter number 13. If you want to run away, jump to chapter 85.”*

A more advanced option is to use bookmarking and cross-referencing features in word processing or presentation software such as Microsoft Word, Microsoft PowerPoint, OpenOffice Writer, or OpenOffice Impress. This can be a good method in nonlinear storytelling for people who are familiar with word processing and presentation software. In addition gamification can also be used as an engaging method to teach the advanced features of office software.

## Hypertext and Wiki Software

For hypertext documents, Hypertext Markup Language (HTML) is the standard markup language, especially used in Internet WWW pages (Berners-Lee, 2001). Basic hyperlinking of documents with HTML language requires only a small subset of HTML language (W3C, 2017). Hyperlinking is created with A HREF attributes and some additional attributes are required for forming valid HTML documents. The biggest drawback for utilizing HTML language with novice users is the need for understanding files and file systems. Each passage in the nonlinear story must be stored in its own file. For example, in our project, students did not have the required skills in file handling, and this makes the task of using HTML difficult.

Wiki software is an improvement in this matter as documents are stored in a database and the users do not need to manually manage them. Also, the markup language is simplified from HTML and users do not need to define document structure.

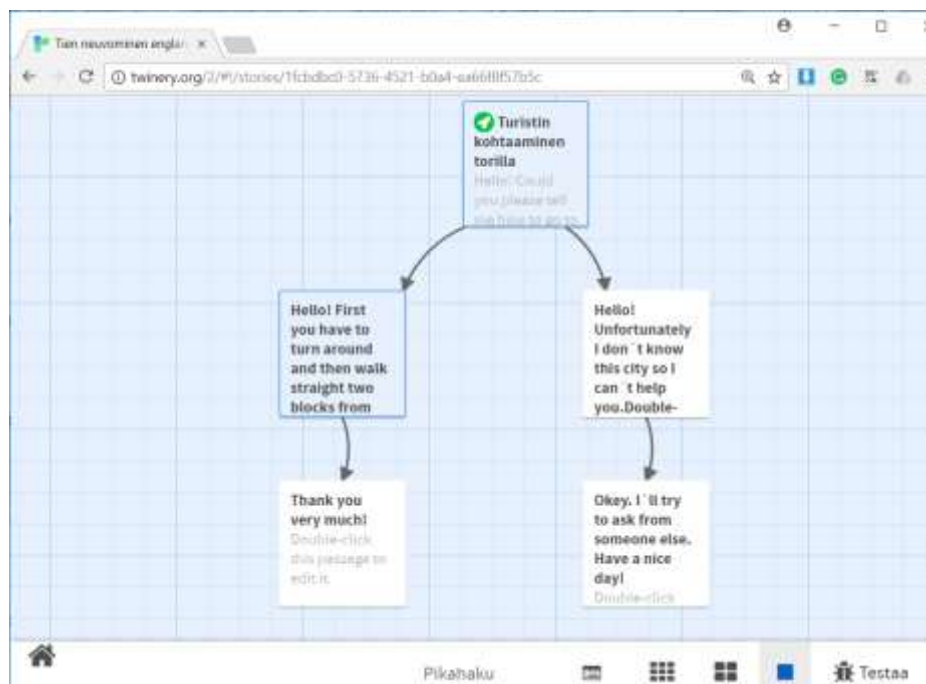
There are several different Wiki software and also many open source options exist. Documents are computer readable and can be exported in simple formats. This makes Wiki documents a very simple and efficient tool for writing nonlinear stories. Using Wiki

for writing also teaches Wiki markup to students so that they become familiar with how the online encyclopedia Wikipedia and other online Wiki resources work.

## Game-oriented software

There exist some game oriented software for nonlinear storytelling. These software have the advantage that they are designed for the purpose of writing game plots. They often have some means to control the story structure and maybe to visualize the story structure in a form of a flowchart. They often have features for a more complex game mechanics such as variables and scripting programming option. Let us review some of the most popular game oriented nonlinear writing tools.

*Twine* is described on its website ([twinery.org](http://twinery.org)) as a tool for telling interactive, nonlinear stories. Its user interface is a flowchart-based. Nodes of the flowchart are known as passages. They contain the text presented to the player as well as hyperlinks and possible macros for controlling the plot. An example of Twine flowchart is presented in Figure 3.



**Figure 3.** Twine flowchart with five passages and one branching.

There are two methods for creating hyperlinks in Twine. The basic syntax is double brackets around the link text. This creates a link to the passage with the same title as the link text. If no passage exists with the same name, a new passage with the name is created. The second method has separate link text and target passage title. There are three syntaxes for this in Twine and its default Harlowe story format:

- `[[linktext|title]]`
- `[[linktext->title]]`
- `[[title<-linktext]]`

*RenPy* ([renpy.org](http://renpy.org)) is a tool for creating visual novels. It has many options designed to integrate visual images into nonlinear storytelling. A story is written in a text file and story structure is created with menus, jumps, and labels. An example of a RenPy code, where two options are provided to the player, is presented in Figure 4. RenPy is a popular and mature software. It is at its best when the story is strongly visual.

```
s "Sure, but what's a \"visual novel?\""  
menu:  
    "It's a videogame.":  
        jump game  
    "It's an interactive book.":  
        jump book  
label game:  
    m "It's a kind of videogame you can play on your computer or a console."
```

**Figure 4.** Story structure control mechanisms for RenPy visual novel engine. Menu item gives options to the player. The Jump attribute moves to the given label in the storyline. This example is taken from RenPy website [renpy.org](http://renpy.org).

*Inkle* is a software for combining visual art with nonlinear stories. It uses a scripting language *Ink*. Ink website ([github.com/inkle/ink](https://github.com/inkle/ink)) describes it as “scripting language for writing interactive narrative, both for text-centric games as well as more graphical games that contain highly branching stories.” An example of Ink scripting is presented in Figure 5.

Another scripting type of software is *ChoiceScript* ([github.com/dfabulich/choicescript](https://github.com/dfabulich/choicescript)). Both *Inkle* and *ChoiceScript* are open source software. *Inkle* also has Unity 3d integration option by a plugin. As scripting languages, these are both textual in nature. This

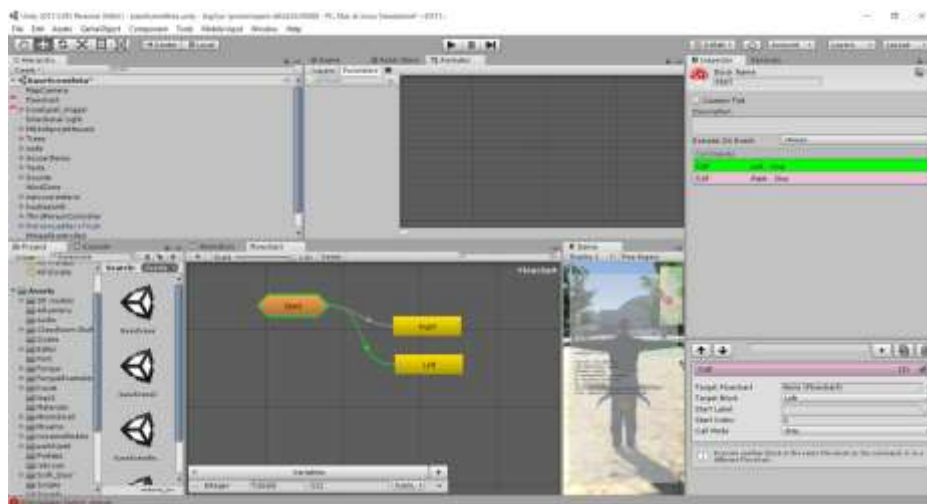
emphasizes the story logic and the story script is very concise. Both of these languages have clear syntaxes and are good for text-based game design when no structure visualization is required.

```
=== paragraph_1 ===  
You stand by the wall of Analand, sword in hand.  
* [Open the gate] -> paragraph_2  
* [Smash down the gate] -> paragraph_3  
* [Turn back and go home] -> paragraph_4  
=== paragraph_2 ===  
You open the gate, and step out onto the path.
```

**Figure 5.** An example of branching in Ink scripting language.

*Fungus* ([fungusgames.com](http://fungusgames.com)) is an open source software for writing stories inside Unity 3d development environment. It is a flowchart-based tool and usability of the tool has been highly paid attention to. The usability is especially good in using variables and conditional branching but these advanced features were not used in our pilot project.

The drawback of *Fungus* is that it requires Unity 3d installation. While Unity 3d is free for personal use, it is a significantly more complex software. The native Unity 3d integration can make the user interface difficult to use for people with low information technology experience. An example view of the user interface is presented in Figure 6.



**Figure 6.** An example view of Fungus user interface inside Unity 3d environment.

*Alice* ([alice.org](http://alice.org)) is a storytelling tool that is designed to be used in education. It is developed by Carnegie Mellon University and its aim is to provide an introduction to computer science concepts and computer programming in an engaging way of telling



stories and creating games. Alice is a rather comprehensive storytelling tool. A 3d scene view of Alice is shown in Figure 7. As an educational tool, it is designed to be easy to use even if it has many advanced features. As an independent storytelling tool Alice is an engaging and compact tool. But it has many internal features in addition to the storytelling features and it cannot be easily integrated into external processes such as general game development processes. Alice has extensive learning resources. For example there is material for designing nonlinear narrative (<https://www.alice.org/resources/lessons/design-interactive-narrative/>).



Figure 7. The Scene view of Alice storytelling tool.

## Methods

An educational computer game was developed by university information technology students and professionals. To acquire content for the game, nonlinear stories were collected from a group of novice computer users. The group of adults (age 30-64 years) participated in a basic information technology training course intended for enhancing their employment opportunities.

During one learning session the serious game development project was introduced to the students and students tried the game. The game is an open world 3d game where the player's avatar can walk around and complete educational tasks.

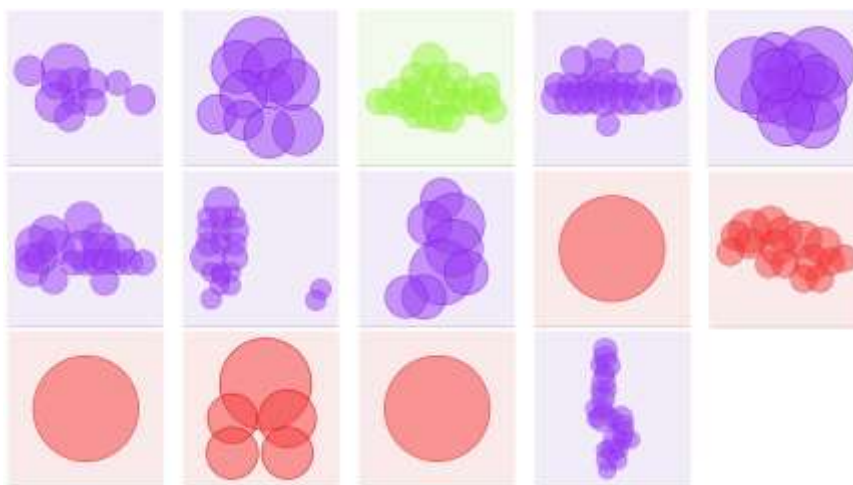
During a second learning session, the basic usage of nonlinear storytelling tool Twine was introduced to the students. Twine was selected for the project because it is an open source software and because it's good integration with Unity 3d game engine.

After the introduction, students were instructed to write a sample story on personal computers at the computer laboratory of the school. Students were able to continue their

stories at home 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.

## Results

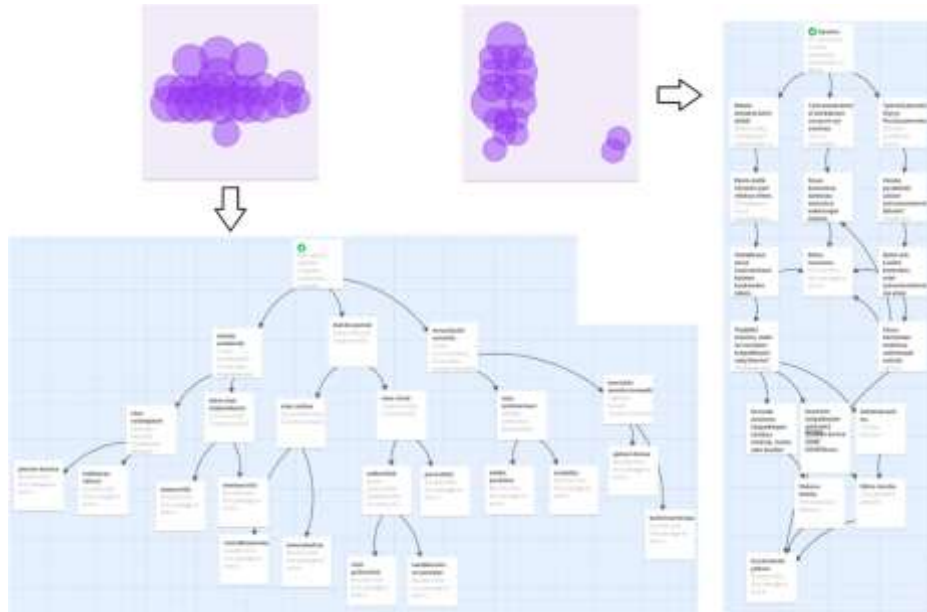
Fourteen stories ( $n = 14$ ) were submitted to the project. The default visualization of Story List view by Twine is presented in Figure 8. In this visualization, each circle presents one passage in the story. Size of the circles correlates with the amount of text in a passage. Colors are only for visual purposes. The overall structure of the story is also visually presented by positions of circles.



**Figure 8.** Visual presentation of the collected stories in Twine.

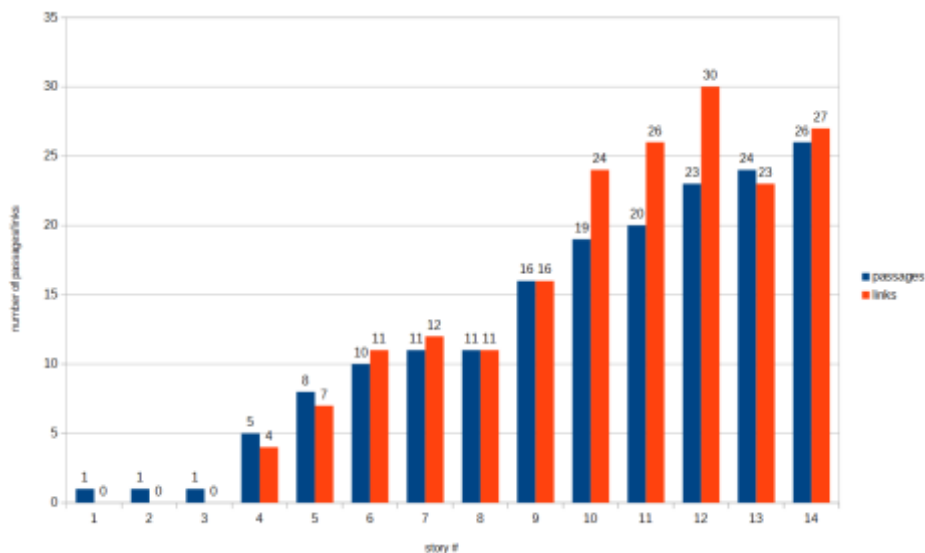
It can be seen from the visualization that there are three stories with only one passage (just a single sphere in the visualization). These stories have obviously some technical problems. Other stories seem structurally good. So the default visualization of the Story List view presented in Figure 8 can be used as a simple inspector of story structure.

A more detailed view of two of the stories, as presented by the Story Edit view of Twine, is presented in Figure 9 for comparison. As can be seen, the visualization in Figure 8 gives an approximate structure of the story. But the Story Edit view also presents the story flow and transitions between passages.



**Figure 9.** A more detailed view of two stories in Figure 8. Two extra passages on the lower right corner are omitted from the latter story.

It is possible to collect certain metrics from the stories to describe the story structure in analytical means. The number of passages and the number of links between passages in each story is presented in Figure 10. Here we can see again the three stories with only a single passage and no links.



**Figure 10.** The number of passages and the number of links between passages in collected stories.

We analyzed the three stories with only one passage and found that they all had the same problem: students had not correctly understood the syntax for creating hyperlinks. An example of the contents of one story is presented in Figure 11. It can be seen that the dialog

has been written as a linear manuscript without any interaction with the player.

In all of these three cases, students had put double brackets over the whole dialog. Double brackets are used for creating hyperlinks, but they should include only the link text provided to the player or a link text and a link title. Double bracket notation works only with single line text so no hyperlink is created in this case.

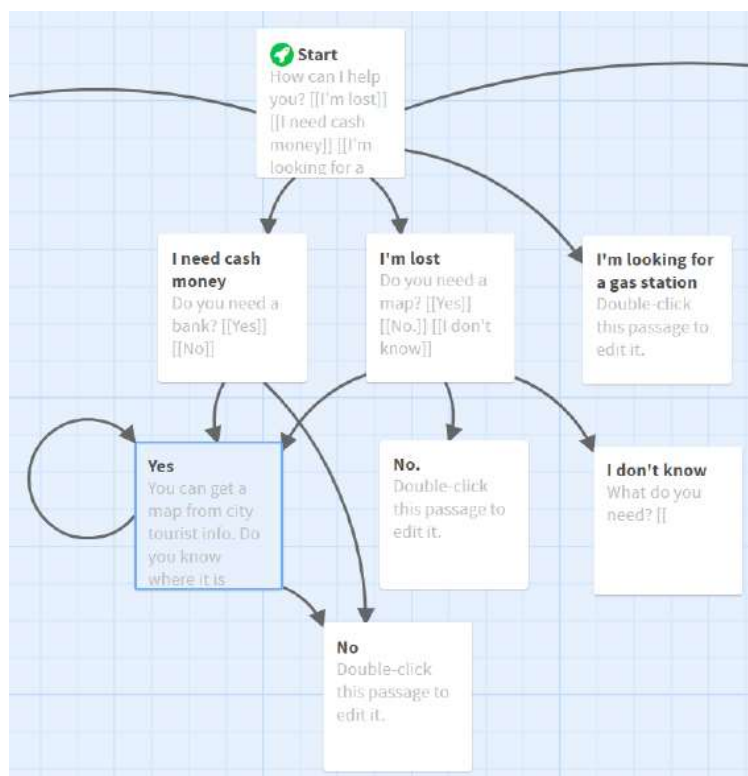
```
[[BARTENDER1  
What would you like to have?  
CUSTOMER1/1  
A large dark beer  
BARTENDER2  
Would you like to have German or Belgian, sir?  
CUSTOMER1/2  
German, please.  
BARTENDER3  
And you, sir?  
CUSTOMER2/1.  
A Scotch and a small beer.  
BARTENDER4  
Okay. Here you are!]]
```

**Figure 11.** An example of a story with only one passage. The student has not understood the idea of nonlinear and hyperlinks properly but has written the dialog in a traditional format. The text has been translated into English from the original Finnish language used in the project.

By default, Twine generates new passages using the link text as a new title. This was a somewhat difficult concept to learn in the introduction phase. The problem is that given two similar link texts, same continuation passage is assumed. This is not always the case and simple hyperlink syntax cannot be used.

In collected stories, only one story had this problem. A part of the story is presented in Figure 12. In this story, there are several questions with Yes and No answers. Questions are very different, “Do you need a bank?” and “Do you need a map?”, so the story should not continue in the same passage.

In the case of negative answer, the student has solved the problem by using “No” and “No.” which are interpreted as two different texts by Twine. But in the positive case both answers lead to the same passage. The problematic passage has been selected in Twine user interface so it is shown as light blue instead of regular white color in Figure 12. This passage has even a further Yes and No question which leads back to the passage itself in the positive answer case leading to a possible infinite loop in the story.



**Figure 12.** An example of a part of a story where the concepts of link text and link target were not fully understood and this has caused a problem with the story structure.

There are several questions to the player with Yes and No answers and automatically incorrectly generated hyperlinks cause an illogical structure. The text has been translated into English from the original Finnish language used in the project.

It is worth noting that most of the stories were not finished completely. Out of the eleven stories that had more than one passage, only two had no unfinished passages. Some of the unfinished passages were story branches that were not written, but some were missing just the closure or the ending for the story.

Three stories had extra passages that were not connected to the story. This is not an error or a problem but can tell something about the creative process of the story. Unconnected passages can tell for example about the editing of the story or experimentation with the story structure.

## Discussion and conclusions

In this article, nonlinear storytelling was introduced as a low-threshold method for co-creative educational game development. Some of the most important characteristic features of nonlinear stories were presented. Nonlinear stories can be seen as hypertext documents, interactive stories or game plots. These are all used in education in different

contexts but there does not exist much research literature on the specific topic of creation of stories with nonlinear structure.

We presented several methods for writing nonlinear stories. Stories can be written with very commonplace tools such as HTML editors, wikis, word processors or presentation software. There also exist several tools for specifically writing nonlinear interactive stories. These tools often have features that make writing nonlinear stories easier such as flowchart visualization and integration with more advanced game engines. They often also have an easy use of scripting, such as using variables and conditional branching, although these advanced features were not utilized in this pilot project and were not introduced in detail in this article.

In our pilot study, 14 stories were collected from adult students with low information technology skills. Students were given only a short introduction to nonlinear storytelling and Twine software used for writing stories. The teacher was only limitedly available to supervise and help in technical problems because students continued their work unsupervised after school hours. This experimental setup was used to reveal common learning challenges and possible misconceptions in nonlinear storytelling.

Out of the 14 stories collected, three stories had some fundamental flaws so that stories contained no hyperlinks. Creating a nonlinear story with hyperlinks was introduced to students beforehand but based on these observations, more emphasis should be put on explaining the user interaction and the concept of branching in nonlinear stories.

The difference between the link text and the target passage title was emphasized during the introduction of the tool as it is maybe the most difficult concept in using hyperlinks. Making efficient passage titles should be considered an important subject in storytelling as it can cause confusion. In this study, only one student had technical problems with the independent link text and the target passage title.

Submitted stories had many unfinished passages. The process of writing a nonlinear story differs from traditional linear story as nonlinear story can have several branches and all of them must be finished. There are several techniques for this. For example, there can be several different endings or branches can be pieced together in a single ending. This idea could be made clear by practicing with small gamified stories first.

We did not include computer programming concepts such as variables and conditional branching in this study. Still, hyperlinks and complexity of the story structure presented challenges for the students. This is something that we are interested in our future research: how do these challenges, recognizing them, learning to solve them and gaining fluency in dealing with them, develop their thinking skills generally.

These skills are also very close to so-called computational thinking skills which are considered to be of great importance in computer science education. It would be therefore interesting to know if useful transferable skills are accumulated during the activity of nonlinear storytelling.

Most of the collected stories were of good quality and ready to be implemented in our game. The chosen software tool was found functional and students were able to use it very independently, although it would be better to give more support during the writing process especially in the early phases of learning.

Students' views were not inquired formally but computer game development embedded in a regular information technology basic training was considered as a refreshing change for their study routines.

This study was a small scale pilot study to better understand nonlinear storytelling process and to increase understanding of how to implement nonlinear storytelling in education. In our future work we will develop teaching methods and materials for nonlinear storytelling in education based on the findings of this study. More research is needed to identify the special features and possible applications of nonlinear storytelling in education. This study will provide some guidance for designing these studies.

Most of the storytelling tools reviewed were open source licensed software. This allows the modification of software. In our future work, we plan to further modify some of the software features to answer the requirements of educational use. These modifications will include features such as enabling better co-creative processes through sharing and remixing stories.

## **Acknowledgment**

This work was supported by European Union Regional Development Fund and the Leverage from the EU 2014-2020 programme.

## References

- Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web. *Scientific American*, 284(5), 34–43. <https://doi.org/10.1038/scientificamerican0501-34>
- Boltman, A. (2001). *Children's Storytelling Technologies: Differences in Elaboration and Recall*. Dissertation. Faculty of the Graduate School of the University of Maryland, Maryland.
- Çıralı, H., & Usluel, Y. K. (2015). *A Descriptive Review Study about Digital Storytelling in Educational Context*. Presented at the 7th International Conference on Education and New Learning Technologies (EDULEARN15), Barcelona, Spain. Retrieved from [https://www.researchgate.net/profile/Hatice\\_Cirali\\_Sarica/publication/284177441](https://www.researchgate.net/profile/Hatice_Cirali_Sarica/publication/284177441)
- Friedman, T. (1995). Making sense of software: Computer games and interactive textuality. In *Cybersociety; Computer-Mediated Communication and Community*. Retrieved from [https://www.researchgate.net/profile/Ted\\_Friedman/publication/277013129](https://www.researchgate.net/profile/Ted_Friedman/publication/277013129)
- Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code.org. *Computers in Human Behavior*, 52, 200–210. <https://doi.org/10.1016/j.chb.2015.05.047>
- Letonsaari, M., & Selin, J. (2017a). Modeling computational algorithms using nonlinear storytelling methods of computer game design. *Procedia Computer Science*, 119, 131–138. <https://doi.org/10.1016/j.procs.2017.11.169>
- Letonsaari, M., Selin, J., & Lampi, M. (2017b). *Co-creative Serious Games Design Process using Nonlinear Storyline Editing*. 582–588. <https://doi.org/10.5220/0006320305820588>
- Leutenegger, S., & Edgington, J. (2007). A games first approach to teaching introductory programming. *ACM SIGCSE Bulletin*, 39(1), 115–118.
- Liu, C.-C., Liu, K.-P., Chen, G.-D., & Liu, B.-J. (2010). Children's collaborative storytelling with linear and nonlinear approaches. *Procedia - Social and Behavioral Sciences*, 2(2), 4787–4792. <https://doi.org/10.1016/j.sbspro.2010.03.771>
- Tedre, M., & Denning, P. J. (2016). The long quest for computational thinking. *Proceedings of the 16th Koli Calling International Conference on Computing Education Research*. <https://doi.org/10.1145/2909541.2909542>
- Topalli, D., & Cagiltay, N. E. (2018). Improving programming skills in engineering education through problem-based game projects with Scratch. *Computers & Education*, 120, 64–74. <https://doi.org/10.1016/j.compedu.2018.01.011>



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), 212–226.

<https://doi.org/10.1177/2042753016689635>

Viitanen, K., Harju, V., Niemi, H., & Multisilta, J. (2014). Digitaalisen tarinankerronnan monet mahdollisuudet. In *Rajaton luokkahuone. Opetus 2000* (pp. 187–211). Jyväskylä: PSkustannus.

W3C. (2017). HTML 5.2, W3C Recommendation, 14 December 2017. Retrieved May 14, 2019, from <https://www.w3.org/TR/html/>

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

<https://doi.org/10.1145/1118178.1118215>