Detecting plot holes in visual novel type games

A visual novel definition language and its application in detecting plot holes

Naomi J.F.L. Vogelpoel
Bachelor thesis Game Development,
HBO-ICT Amsterdam University of Applied Sciences
General information

Name: Naomi Vogelpoel
Student number: 500733490
Email: NaomiVogelpoel@hotmail.nl
Bachelor thesis HBO-ICT: Game Development
Amsterdam University of Applied Sciences

Date: October 25, 2019

Supervisor: dr Anders Bouwer
Advisor: ir. Riemer van Rozen
Internship: Lectoraat Play & Civic Media
Amsterdam University of Applied Sciences
Abstract

This research was conducted to create insight into how story structure and plot holes correlate to one another in the story of visual novels. Most games have a story of some sort, but visual novels have a particularly great focus on the story. Visual novels are a type of interactive fiction and are therefore text-based games. They contain very little in terms of gameplay elements. The gameplay elements in visual novels usually present themselves in the form of puzzles. This research mostly concerned itself with answering the following question: “How can we create a framework that is capable of analyzing and repairing plot holes by looking at the story structure in stories for visual novel type games?”. This question was answered though desk research and research through design which resulted in a proof of concept in the form of a program.

From the desk research we learned that a story is the sequence of events that lead to the end of the story. Stories contain elements that are carefully created by the author. An example of this is a narrative arc. The plot is solely the structure of a story. There are three different types of plot: the three-act story structure, the hero’s journey and an episodic story structure.

A plot consists of different sections, which are called plot points. The characters of a story and the author can both have different goals. When the goals of the character and the goals of the author are balanced, the plot is logical and no plot holes occur.

A plot hole is an inconsistency that does not align with the flow of logic that the author has established in the story. This inconsistency can be caused by different story elements and can therefore be the source of a plot hole. There are different types of plot holes that contradict different parts of said logic. Stories in visual novels can be made interactive through the use of puzzles, twines or by asking the user questions about previous events.

Using research through design it was found that plot holes can be detected in a non-natural language. This can be done through the use of C# and ANTLR. By creating a lexer and a parser ANTLR can generate code in C# that will be capable of creating a story tree. This story tree can be used to create a story model from which plot holes can be detected. To do that the story model must first be checked on completeness, which can be done through the use of a resolver and the user input must be checked on mistakes which can be done through the use of a checker. The system created is currently capable of detecting four types of plot holes.
Acknowledgements

For making this research possible I would like to thank a few people. First of all: Anders Bouwer and Riemer van Rozen for supporting this research and providing me with a lot of support and information needed for my research. The research group at lectoraat Play & Civic media and the people at CWI for letting me use their workspace and all the encouragement they have given me. Robbin Schepers and Riemer van Rozen for continuously helping me with the programming aspect of this thesis. My friends and family for believing in me. With special note to my boyfriend Dennis van den Noort, my parents and my brother for their continuous support. And lastly, you for taking the time to read my thesis.
Index

1. Introduction 7
   1.1 Motivation for the research 7
      1.1.1 Visual novels & Interactive fiction 7
   1.2 Research questions 9
   1.3 Method 10

2 Related work 12
   2.1 Visual novels 12
   2.2 Structuring stories in writing 13
      2.2.1 Story 13
      2.2.2 Plot and plot points 13
      2.2.3 Plot types 14
         2.2.3.1 The three-act story structure 14
         2.2.3.2 The hero’s journey 14
         2.2.3.3 The episodic story structure 15
      2.2.4 MacGuffins 15
   2.3 Making stories interactive 16
      2.3.1 Interactivity in visual novels 16
      2.3.2 Twines 16
   2.4 Generating storylines 18
   2.5 Defining a story and creating a language 19
      2.5.1 A graph of existents or story graphs 19
      2.5.2 Defining a story through tools 19
      2.5.3 Lexers and parsers 19

3. Plot holes 21
   3.1 Identifying plot holes 21
      3.1.1 Defining different types of plot holes 21
   3.2 Extension to defining different plot holes 23
   3.3 Plot holes represented in Scribbl 26
      3.3.1 Faulty time traveled 1 26
      3.3.2 Faulty time traveled 2 and Unexplained: Object displacement 28
      3.3.3 Trait changes 28

4. Domain analysis 29
   4.1 Parts 29
      4.1.1 General story structure 29
      4.1.2 Plot elements 30
4.1.3 Characters
  4.1.3.1 Character Physiology
  4.1.3.2 Character Sociology
  4.1.3.3 Character Psychology
  4.1.3.4 Character Archetypes

4.1.4 Plot types

4.2 Plot hole types

5. The inner workings of Scribbl
  5.1 Quality requirements for the system
  5.2 Choosing the environment
    5.2.1 Choosing the IDE
      5.2.1.1 ANTLR
  5.3 System Overview
    5.3.1 Process diagram
    5.3.2 T-Model/Tombstone diagram
    5.3.3 The component model
  5.4 Implementation details
    5.4.1 Defining the Scribbl.g4 file for Scribbl
    5.4.2 Executing the Scribbl.g4 file
    5.4.3 Creating the builder/visitor override
    5.4.4 Creating the tree walker
    5.4.5 Creating the logic error finder
    5.4.6 Creating the plot hole finder
    5.4.7 Other resources used in this project
  5.5 Using the System
    5.5.1 Input
      5.5.1.1. Defining input
    5.5.2 Running the program
    5.5.3 Output
    5.5.4 Testing the system
      5.5.4.1 The first phase
      5.5.4.2 The second phase
      5.5.4.3 The third phase
      5.5.4.4 The fourth phase
    5.4.5 Changes made after testing the system
6. Contributions and future work 63
   6.1 Advantages and limitations of the system 63
      6.1.1 Theoretical contributions 63
      6.1.2 Practical contributions 63
   6.2 Tool limitations 64
   6.3 Advice for future work 65

7. References 66

Appendix A .scribb example 69
Appendix B output example 79
Appendix C Second test phase results 89
1. Introduction

In this chapter the reason why this research was done will be discussed, this includes a short introduction to visual novels and interactive fiction. The research questions that were used during this research and the way this research was executed can also be found in this chapter.

1.1 Motivation for the research

Almost all games have a narrative (Uchikoshi, 2013). This is because most games (apart from a few abstract games like tetris) gain benefit by using story elements (Johnson, 2013). In some games there are short backstories, serving to set up the action. (Cheong, O. Riedl, Bae & Nelson, 2016). Especially visual novels have a great focus on the story (Van der Geest, 2015).

In stories plot holes can occur. A way for plot holes to occur is when within a story a conflict between the goals of the characters and the goals of the author is covered up or ignored Ryan (2009).

According to Ryan (2009), good plots come from the characters in them, their memories, the events that already happened in the story and the way they react to the events happening.

The aim of this research is to create insight on preventing plot holes to help game designers write their story for an interactive fiction or visual novel. There are two main reasons for this research. The first reason is that when writing a story there are a lot of elements to keep track of and a lot of factors to keep in mind. If this is not done properly plot holes will appear and plot holes can be perceived as annoying by the reader.

The second reason is that the many elements make writing stories relatively difficult and it might therefore be helpful for game designers to have a tool that creates insight in plot holes to help them write such stories.

1.1.1 Visual novels & Interactive fiction

Following Paul (2018) and Aarseth (2009) an interactive fiction is defined here as a type of (computer) game that usually allows players to use a set of text commands to control the main character of the game (Paul, 2018). Aarseth (2009) mentions that interactive fiction is another name for text-only adventure games. Following Valens (2016) and van der Geest (2015) visual novels will be defined here as a type of interactive fiction (Valens, 2016), which usually does not contain many gameplay elements and has a great focus on the storyline. The gameplay elements that visual novels do contain, if they have them, are usually in the form of puzzles (van der Geest, 2015).

An example of an interactive fiction is Zork, mentioned by both Montfort, Marcus and Prince (2007) and Murray, J. H., & Murray, J. H. (2017). Zork was made by Tim Anderson, Marc Blank, Bruce Daniels, and Dave Lebling (Wikipedia, 2019). An example of a visual novel is Virtue’s last
reward made by Spike Chunsoft (Uchikoshi, 2013) or Long live the queen made by Hanako games (Hanako games, 2013)
1.2 Research questions

The primary focus of this research is to answer the following question:

“How can we create a framework that is capable of finding plot holes by looking at the story structure in stories for visual novel type games?”

To answer this question in this research a few subquestions will be studied. These questions can be answered through literature research or through research through design. Questions one to three will be answered purely through literature research and the questions four and five will be answered through a combination of both research methods.

The subquestions:

1. How are stories structured?
2. What are plot holes and how do they relate to story structure?
3. Which tools currently exist that can be used to support story designers?
4. How can plot holes be detected?
5. Which types of plot holes exist?
1.3 Method

The goal of this research is to find out how a framework can be created to find plot holes in the story structure. This research is meant to be an explorative research in which both the theoretical and the practical aspects of finding plot holes in visual novels come to light. On the theoretical side of this research we hope to provide insight into what plot holes are exactly and provide a way to solve them through the practical aspect of this research. Where we hope to deliver a proof of concept which demonstrates that plot holes can be found through the use of code.

To conduct this research successfully we first did research on how stories are structured. This was done with the purpose of gaining a better understanding on the general principle of storylines, story structure, visual novels and on what plot holes are. We also did this with the purpose of finding out if plot holes have a relation to story structure. Then a research was done to see if there were any tools that did something similar to what we were hoping to achieve and/or could be used to modify our needs.

The desk research was done through reading articles and book about the subject. These articles were found by searching terms such as “Visual novels”, “storyline in games”, “plot holes”, “plot holes in visual novels”, “visual novel creation”, “plot”, “lexers and parsers” and similar terms through google (scholar) or through recommendations from Mr. Bouwer and Mr. van Rozen. The books that were utilized were mostly found and borrowed through the research group.

To make this research more visual and to figure out what would be required to identify a plot hole I created a domain model, which can be found in chapter 4. During the research it also became apparent that the categorization of plot holes as it currently existed was not sufficient to successfully be able to identify plot holes through code. Therefore I created an expanded categorization based on the categories that already existed. This categorization can be found in chapter 3.2. After this research was done several brainstorm sessions took place between myself, Mr van Rozen and Mr. Bouwer to determine the requirements for the design and the functionality of the system.

We decided we wanted to create a tool that would analyze a script that the user uses as input and find plot holes based on this script (scribbl language). We wanted the user to be able to receive output that states which plot holes were found. If possible, we wanted the system to accompany this with a graph that represented the plot hole.

When building the tool we decided on using ANTLR in combination with C#. This allowed us to create our own programming language of some sorts that the user uses to represent his or her story. We later called this programming language scribbl language. The tool then translates this language into a parse tree which can be walked through and information can be retrieved. The
bit of code that does this is called the treewalker. Then the builder was made. When those bits of code were running the next step was to build a checker to check if the input was correct. Then the tree to identify the plot holes was made. Lastly the classes were made that would generate the output of the system. The code used in this project was made with some help from Mr. van Rozen.
2 Related work

In this chapter visual novels will be described in more detail. Then storylines and story structure will be discussed. Then we will move on to discussing the interactivity in storylines and the tools to create storylines.

2.1 Visual novels

“The essence of a visual novel is a video game with an emphasis on narrative” - (Uchikoshi, 2013).

There are a lot of different types of visual novels. On vndb.org there are currently 25267 different visual novels registered (vndb, n.d). Visual novels have mostly evolved in Japan where the genre is very popular (Uchikoshi, 2013) As stated before, visual novels can be seen as a genre of interactive fiction (Valens, 2016).

Uchikoshi (2013) says that some people do not view visual novels as games. He disagrees with this statement because he sees a game as something with selectivity. Something can be qualified as a game when, within a given set of rules, the history or the outcome of events is changed through the selection made by the players decision. He says that a visual novel is a video game that specializes in selectivity.

Van der Geest (2015) describes a visual novel set-up as a character displayed on a screen with a box of text somewhere in the area around the character which holds the character’s dialogue and the player’s dialogue. The color of the text and character displayed may vary depending on who is speaking at that time. The player is usually not displayed because the world is seen through the eyes of the protagonist.

Uchikoshi (2013) says that the biggest strength of a visual novel is that players are able to interact with the story at will. Johnson (2013) calls interactive fiction (and therefore visual novels too) one of the most tantalizing aspects of mixing video games and narrative. Although he feels that the potential of interactive fiction is unrealized because the amount of branches is limited.
2.2 Structuring stories in writing

A story is generally an important aspect of a game. The story can even be the distinguishing factor of a game in cases where the gameplay is identical. Aarseth (2009) says that in some cases, such as in DEAN FOR IOWA and KABOOM!, the gameplay of these two games is identical but the story is what makes the game two completely different entities. In a game the story is often intertwined with the gameplay. The role of a story varies depending on the genre of the game and its mechanics (Cheong et al., 2016).

2.2.1 Story

Stories (also referred to as storylines) contain elements that are carefully crafted by the author, such as a narrative arc, dramatic tension, a certain pace etc. (Cheong et al., 2016). In a world, the story is a sequence of events which eventually lead to the ending of the story (Alez, 2017 and Cheong et al., 2016). Alez (2017) says that an event can be defined as something that happens to a character during the story that has a meaningful impact on a character’s life. These events help to transform the character’s world.

A story always has a backstory even though it is not always visible (Krawczyk and Novak, 2006). Merriam-Webster (n.d.) states that a backstory is a story that gives the viewer information as to what led up to the main story or plot. Krawczyk and Novak (2006) say that games utilize backstory in the same way that movies, novels, or plays do. They say that a backstory is useful because it orients the players and it helps the players to understand what they can and cannot do. It also helps the reader to understand their enemies within the game.

2.2.2 Plot and plot points

The difference between a plot and a story is that the plot is the backbone of the story. While the plot does not contain the elements such as dramatic tension that makes the story interesting to read by itself, the plot is very important to make the story overall coherent and interesting (Krawczyk and Novak, 2006). A plot is the sequence of all the plot points that the plot is made out of. Plot points are the events that impact the story (Krawczyk and Novak, 2006 and NowNovels, n.d.). Hull (n.d.) does support this definition but describes plot points as big events rather than small ones (as done by Krawczyk and Novak). He believes that a plot consists of only four different plot points: the inciting incident, the first act turn, the midpoint and the second-act turn. Following Krawczyk and Novak (2006) in this paper the term plot point is defined here as the small events which jointly make up the plot of the story.

Ryan (2009) says that a plot exists on two levels: the plotting of the characters and the plotting of the author. Ryan makes this distinction because if plotting is done well, the characters in the story will start wanting different things which might not completely fit in the story that the author has planned out. This causes a clash between what the characters themselves want to do and
what the author wants to do. When a plot is bad the character’s personal goals are ignored and the plot is completely driven through external circumstances. In good plots the needs of the character are portrayed in the logical reasoning that they use. In great plotting the events arise from actions that are probable within the circumstances and the characters use their memory and inner disposition to steer their actions (Ryan, 2009).

2.2.3 Plot types

There are different ways to structure a story. Krawczyk and Novak (2006) give three different ways to achieve a structured story through the use of plot types. They mention three different types of plot: The three-act story structure, the hero’s journey and an episodic story structure.

2.2.3.1 The three-act story structure

Out of the three types of plot mentioned previously, the three-act story structure is the most straightforward one. It consists of a beginning to capture the viewer’s attention and introduce the problem. The middle to build tension and present obstacles. Finally, the end to provide closure and resolve the problem (Krawczyk and Novak, 2006).

2.2.3.2 The hero’s journey

The hero’s journey is a more complex story structure with more components. It contains twelve parts in total:

1. It starts off with the ordinary world in which we can see the hero in his normal life and his usual surroundings.
2. Then he or she is asked to go on a journey or a quest. Here the hero is introduced to an alternate world.
3. At first the hero refuses because of the sacrifice that would have to be made by leaving the comfortable surroundings he or she is in.
4. The hero meets with a mentor and receives a piece of information that is relevant to the quest and need to go.
5. The first threshold needs to be crossed and here the hero abandons the initial refusal (because of the information received in stage 4) and starts the journey into the special world.
6. The sixth stage is Tests, allies and enemies this is where, as the title suggests, the hero meets allies or enemies. The hero’s resolve will be tested in this stage.
7. Then the hero comes into the phase where he or she approaches the innermost cave. In this stage there will be more tests and a period of supreme terror and/or wonder. In this phase preparations are made for the ordeal, which is phase eight.
8. The ordeal. This is the biggest challenge that the hero faces thus far and in this phase the hero must defeat the main villain of the story.
9. After defeating the main villain, the hero receives a reward which feels like the end of the story but that usually is not the case.
10. In the tenth phase the ordeal is finally over. This leaves the hero with the choice to stay in the special world or to return to the ordinary world.

11. The eleventh phase: Resurrection. Here the hero faces death one more time in another ordeal. This is also known as the climax. The hero demonstrates that he or she has changed by the journey and is therefore resurrected in a more complete and fully realized person.

12. Return with the elixir. This is the final stage. In this stage the hero returns to his or her original situation but is forever changed because of the experience. Sometimes the hero returns with an elixir to help the people originally left behind (Krawczyk and Novak, 2006 and Reedsy, 2018).

2.2.3.3 The episodic story structure

Krawczyk and Novak (2006) describe an episodic story structure as one that is generally used in TV shows where each of the episodes have their own little story and there is one story that connects these episodes together.

2.2.4 MacGuffins

According to Whistler (2017), a MacGuffins is “an object, event or person that the characters in a story value greatly - so much so that nearly the whole plot revolves around it”. MacGuffins are, according to Miyamoto (2018), motivating elements that exist because they drive the plot. The MacGuffin is usually an object so desirable that they motivate the protagonist (and a lot of times, the antagonist) to pursue them. The MacGuffin is usually the cause of the conflict that the main character, but sometimes also other characters, are dealing with within the story.

Whistler (2017) said that the term (first used by Alfred Hitchcock) is usually something that the characters care a lot about but what it is itself does not matter that much for the story. The fact that it is meaningful to the characters is the only thing its existence adds to the story.
2.3 Making stories interactive

A common way of making the story interactive is through the use of a quest where the game gives the player something to do in the world. What they have to do specifically is often motivated by the storyline and when the quest is completed it will usually help the story to advance in some way. Sometimes the level progression of a game has very little to do with its storyline but other times a story provides a narrative arc in which the game takes place and therefore the progression is structured by the storyline of the game (Cheong et al., 2016).

2.3.1 Interactivity in visual novels

Van der Geest (2015) says that the setting and the narrative of a visual novel are introduced through both narrative play and visuals. Visuals are used to introduce the location and the characters. Narrative play is used to display the story. According to van der Geest (2015) within a visual novel one can add gameplay by asking the player questions about either events that have already passed in the story or about what is going to happen in the next section of the story. Therefore, players get influence over the story through interaction. The answer that the player gives is influenced by two things, one of which is the way the choices are emotionally brought to the attention of the players and the other being the psyche of the player.

2.3.2 Twines

When the game asks questions about what will happen in the next section of the story sometimes the storyline changes because of the answer. When this happens, twines are used. Twines happen when the player can make choices to influence the storyline of the game (van der Geest, 2015). For example in Bandersnatch (Netflix, 2018) the player can choose between accepting the offer for a dream job or declining it. This decision plays a big role in how the movie ends and how the story plays out. The choices do not always have to present themselves in a straightforward way. This is shown in the game Orwell by Osmotic Studios (2016). In the game Orwell the player needs to collect information chunks about people and the game does not always let you know if uploading that specific chunk will impact the storyline. Van der Geest (2015) says that twines can be used to analyse choices in games. Some twines just have fail states and no success states. Sometimes the game allows the player to revisit his or her choice. According to Goodwin (2014) these choices can have different types of motivations. One type of motivation is that players can decide on what they ethically would feel like is the right choice to make. Another type of motivation would be that the player can choose the type of consequence he or she wants from their decision.

Van der Geest (2015) says that games like VLR intend for the player to visit every single twine and find all the different endings. Goodwin (2014) mentions the game Papers please by Lucas Pope (2013) which has the same intention for the player. This is shown through the fact that on one of the first screens it shows a couple of dots that will light up for each ending found. This is a simple way to let the player know to go back and find all the different endings. This
phenomenon is also mentioned by Goodwin (2014) and he describes a problem that comes with it. This problem occurs when the players are trying to achieve different endings in a story. The story will have them walk through certain parts of the story that stay the same as other times that they have played them. This will make the player feel like their decisions have less impact. He calls this phenomenon the “rethread problem”, which means that players will not feel like decisions that they make have real consequences. One of the ways to avoid this is by creating a structure that is so dense that players do not feel the need to explore it as much. However, this method is very expensive and difficult to make. Johnson (2013) says that this is usually not a viable option because when production costs rise the developers simply cannot afford to create sections of which there is no guarantee that the players will experience them.

A less labor intensive alternative is to have checkpoints instead of being able to save the game. This ensures that the player is not able to go back in the storyline without starting a new game. Checkpoints make the length between playing the same part of the story longer and therefore the player will not notice it as much (van der Geest, 2015). Uchikoshi (2013) mentions that this is a tactic that is used in the visual novels Kamaitachi and VLR.
2.4 Generating storylines

Procedurally generated game stories are especially useful if used in a game world that is also procedurally generated. This is because if there is a generated world in which stories do not get generated the game world might lack context, motivation, and meaning for the player. (Cheong et al., 2016)

Storylines can be generated through planning algorithms. Algorithms that search in the plan space and in the state space are both useful in generating storylines. One of the ways one can look at procedurally generated storylines is to view them as a planning problem and therefore, stories can be generated through a simulation of what happens when the characters move around the story world and then summing up the events that happened as the characters moved around (Cheong et al., 2016). In TALE-SPIN (Meehan, 1977) this principle is used. TALE-SPIN works through providing the characters in the story with a setting, some basic rules for the world, personalities, relationships between characters and a goal. The program then attempts to solve each character’s problems according to these factors. This method does lack certain features that are usually made through careful planning by the author and therefore the story usually does not feel complete or high-quality (Cheong et al., 2016). A story can also be defined using story graphs according to Montfort, Marcus and Prince (2007), which shall be discussed in the next chapter.
2.5 Defining a story and creating a language

Defining a story could be done through a graph of the things that exist within a story world (existents). This is a common practise in the world of interactive fiction according to Montfort, Marcus and Prince (2007). They mention that this is also how Inform7, made by Nelson (2000) defines stories.

2.5.1 A graph of existents or story graphs

All of the existents of a world are the objects that are represented in the story world. A graph of existents can be made by giving each existent a parent to represent their place in the world. The highest parent in the hierarchy is the COSMOS. So for example if you have a world it could be a child of the COSMOS. A continent could be a child of that world which, in of itself has countries. These countries may have cities as their children. If this is done with every existent of the world, a tree forms to represent all the objects in the story world (Montfort, Marcus and Prince; 2007).

2.5.2 Defining a story through tools

To represent dialogue in a game there are two techniques that can be used. One is based on story nodes and the other is based on scripts. One can also decide to use a hybrid approach. (Manning, 2019).

In a node based representation of a story the story is represented in a directed graph. In this representations the vertices represent the lines of dialogue and the edges are the choices. If there is more than one edge attached to a vertice it represents a choice that the player can make. This type of representation is usually created using a visual tool and it is easy to read and visualize. The downside is that, because you need to create a new node for each line it is quite fiddly to construct. Examples of this type of representation include: Chatmapper and Articy:draft (Manning, 2019).

When using a script based representation the dialogue is represented through lines of code and a special syntax is used to represent choices. (Manning, 2019).

2.5.3 Lexers and parsers

Lexers and parsers can be used to define a grammar. This grammar represents the different elements out of which the input consists. So for example if one wants to make a definition for “cheese” he or she would first need to tell the parser that cheese is a string. But the parser does not know what a string is so this is when the lexer comes into play. The lexer tells the parser what a string is and this makes the parser able to parse cheese and gives us the ability to define it. This grammar of language can be used to analyze something. Lexers and parsers are two different things. A parser is a program that takes a piece of text and transforms it into an organized structure. To do this the parser uses a lexer because a lexer is capable of processing
the characters and then transforms these characters into tokens that the parser can use to create the organized structure. It is possible to create a parser by hand but it is also possible to use a program to help generate your parser such as ANTLR (Tomassetti, 2017). Montfort, Marcus and Prince (2007) also use lexers and parsers to generate storylines.

It would also be possible to, instead of a parser, use regular expressions. Tomassetti (2017) argues that this is not the best method to create your language. This is because there is a lack of recursion unless it is done by hand. Another argument against it is that using regular expressions is rather difficult (Tomassetti, 2017).
3. Plot holes

This section is about identifying and defining different (types of) plot holes and cheap plot tricks. Plot holes can exist in traditional stories and are frequent in films (Ryan, 2009), but they can also exist in visual novels (Rinne, 2019).

3.1 Identifying plot holes

Pettersen De Lucena and Ribeiro da Mota (2017) mention that the visual novels Heavy rain and Miyazaki’s Bloodborne contain plot holes. Shadowdeku (2017) mentions a plot hole in the zero-escape series where he describes a logic plot hole (as described by Miyamoto, 2018) that has to do with time shifting within the visual novel. The game long live the queen (Hanako games, 2013) also contains a plot hole where a magic crystal cannot attune with someone if the last holder was an enemy. However, in a certain path later in the game it is possible to have Charlotte (the main character’s lovely niece) attune with the crystal that belonged to Togami (the foreign king that attacked Elodie’s country). Since the latter is an enemy and the former is not, this could be considered a plot hole.

3.1.1 Defining different types of plot holes

According to Kieffer (2018) a plot hole is: “A gap or inconsistency in a narrative that specifically contradicts the flow of logic established in the story.”. Wikipedia (2018) and the Urban Dictionary (2013) support this definition. Ryan (2009) adds to this by saying that it can happen in both the motivational and the logical texture of the story.

Movieplotholes (n.d.) specifies three different types of plot holes: The minor plot hole, the major plot hole and the super plot hole. The minor plot hole is a plot hole in a small detail in a movie and therefore does not have a great impact on the story, for example if a character is said to have blue eyes but then later on, appears to have brown eyes. The major plot hole is when an inconsistency in a character’s behaviour occurs, for example when a good character turns evil with no plausible cause. This is also a plot hole named by Miyamoto (2018), although he calls this a character plot hole. A super plot hole just destroys the logic of the entire story, for example if in a world magic does not exist while at the end magic saves the day. Miyamoto (2018) also reports the phenomenon that Movieplotholes (n.d.) calls a super plot hole but he defines it as a logic plot hole, which according to him, occurs when the writers break their own set of rules. There are also other plot holes, namely, the narrative plot hole which is an inconsistency or a gap in the storyline itself or the Macguffin plot hole, which is an inconsistency that relate to the MacGuffin of the story (Miyamoto, 2018). Out of the five types of plot holes described by Miyamoto (2018) the MacGuffin type has the least impact on the cinematic story.
Miyamoto (2018) also mentions the deus ex machina as a plot hole. However, Ryan (2009) sorts the Deus ex machina under cheap plot tricks rather than a plot hole. In the remainder of this thesis the Deus ex machina will be considered a cheap plot trick rather than a plot hole.

According to Ryan (2009) cheap plot tricks (CPTs), are go-to devices that the writer uses to get him or herself out of a certain scenario created by the plot. CPTs usually make the world seem smaller to the reader. Most of these devices involve extreme forms of coincidence.

It was found that automatic detection of plot holes is not something that has been done to our knowledge, therefore an extended model regarding plot holes is needed to successfully complete this task, which will be discussed in the next chapter.
3.2 Extension to defining different plot holes

In table 2 the different kinds of plot holes will be subdivided into smaller categories and explained. This model was made with the purpose of defining the plot hole through code since code requires the definitions to be as specific as possible. In the ‘type’ column the type of plot hole is defined according to the definitions of Ryan (2009) and Miyamoto (2018). As to not make this table too large, abbreviations will be used in table 2. These abbreviations will be displayed in table 1.

*Abbreviations used*

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>Character</td>
</tr>
<tr>
<td>Ca</td>
<td>Character appearance</td>
</tr>
<tr>
<td>D</td>
<td>Distance</td>
</tr>
<tr>
<td>E</td>
<td>Event</td>
</tr>
<tr>
<td>L</td>
<td>Location</td>
</tr>
<tr>
<td>M</td>
<td>Motivation</td>
</tr>
<tr>
<td>O</td>
<td>Object</td>
</tr>
<tr>
<td>P</td>
<td>Personality</td>
</tr>
<tr>
<td>Pr</td>
<td>Property/ properties</td>
</tr>
<tr>
<td>R</td>
<td>Rule</td>
</tr>
<tr>
<td>S</td>
<td>Skill</td>
</tr>
<tr>
<td>Sp</td>
<td>Speed</td>
</tr>
<tr>
<td>Te</td>
<td>Time event</td>
</tr>
<tr>
<td>Tp</td>
<td>Time predicted</td>
</tr>
<tr>
<td>Tpt</td>
<td>Time past</td>
</tr>
<tr>
<td>Tr</td>
<td>Trait</td>
</tr>
</tbody>
</table>

*Table 1: Abbreviations used in this chapter*
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
<th>Plot Points/Participants/objects</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>MacGuffin unexplained</td>
<td>MacGuffin plot hole</td>
<td>When the MacGuffin is not fully explained or changes properties.</td>
<td>Oa has properties Pr. Oa has properties other than Pr.</td>
<td>Eating the fruit would make someone capable of bending steel. Robert eats the fruit. Thanks to the fruit he can now bend water.</td>
</tr>
<tr>
<td>Unexplained: Object displacement</td>
<td>Logic plot hole</td>
<td>When in the world an object appears in location L1 and L2 but it is not explained in the story.</td>
<td>Oa is seen in L1 and then Oa is seen in L2 without it being moved.</td>
<td>For example: Suzie left her bracelet in the cave. So Suzie and Rick went to retrieve it. They find Suzie's bracelet in the river next to the town.</td>
</tr>
<tr>
<td>Rules of universe ignored</td>
<td>Logic plot hole</td>
<td>When in a world a certain rule, which has to do with what is possible in the world, is defined but later on the rule is broken.</td>
<td>E1 is possible. E2 is not possible. After some time either E1 is no longer possible or E2 now is possible.</td>
<td>In The Chilling adventures of Sabrina, they explain that when you use Astral projection you can see everything that is happening on the other side but can't touch anyone or anything. It is also described as something you should not be doing. Later on they use astral projection rather freely and touch people while doing so.</td>
</tr>
<tr>
<td>Object transmutation</td>
<td>Narrative plot hole</td>
<td>When in the world a desired object is changed from one thing to another without a logical explanation</td>
<td>Mc desires only O1. Mc does not obtain O1. Later on. Mc desires only O2.</td>
<td>Everyone was after the ring that could make the wearer all powerful. When they found the necklace it was white but it still made the person wearing it all powerful.</td>
</tr>
<tr>
<td>Faulty past 1</td>
<td>Narrative plot hole</td>
<td>When in the story one character says that something happened in the past but this event does not fall in the timeline correctly.</td>
<td>C says E1 happened at Tpt. But E1 actually happened at Te which is not equal to Tpt.</td>
<td>The history teacher said that in 1982 Rome was very peaceful. All the Romans were happy. In 1982 a civil war happened in Rome.</td>
</tr>
<tr>
<td>Faulty past 2</td>
<td>Narrative plot hole</td>
<td>When in the story one character says that something happened in the past but this event is disregarded in the rest of the story.</td>
<td>C says E1 happened but it is as if the event has not happened at all.</td>
<td>Jan and Christie go on an adventure. Christie is sad because her house burned down. They go and visit the castle. Jan tries to cheer up Christie. Afterwards they both happily go home.</td>
</tr>
<tr>
<td>Faulty future prediction</td>
<td>Narrative plot hole</td>
<td>When in the story one character says that in a certain time something will happen but when that time comes it does not actually happen. Without giving an explanation</td>
<td>The C says in Tp E1 will happen, but E1 happens in Te, which is not equal to Tpt. The C says that in Tp E1 will happen but in Tp E2 happens. While E1 does not happen at all.</td>
<td>In Doctor Elise the main character says that she will be meeting the emperor that day but the day passes and she does not meet the emperor.</td>
</tr>
<tr>
<td>Faulty time traveled 1</td>
<td>Narrative plot hole</td>
<td>When a character in a story travels from one place to another.</td>
<td>L3 is D km from L1. C travels from L1 to</td>
<td>John travels to a foreign land a thousand kilometers away. He is traveling by foot</td>
</tr>
<tr>
<td>Faulty time traveled 2</td>
<td>Narrative plot hole</td>
<td>When the same character is in two different locations at the same time</td>
<td>C is at L1 at T1. C is also at L2 at T1.</td>
<td>Anna is talking to charles in London. At the same time charles is talking to Bert in Amsterdam.</td>
</tr>
<tr>
<td>-----------------------</td>
<td>---------------------</td>
<td>-------------------------------------------------</td>
<td>------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Rule breaking</td>
<td>Narrative plot hole</td>
<td>When the characters of a story break a rule and are applauded/ not reprimanded for it.</td>
<td>R1 is established C breaks R1 C does not face negative consequences</td>
<td>In Karate Kid the Kid wins the final match with a kick to the head while it is previously stated that such a kick is not allowed as mentioned by Miyamoto (2018).</td>
</tr>
<tr>
<td>Unknown skills</td>
<td>Character plot hole</td>
<td>When the character of a story unexpectedly has a skill which he did not possess earlier in the story.</td>
<td>C does not have skill S. C uses S1.</td>
<td>Anna does not know karate. She enters in a professional karate contest anyway. She wins the contest.</td>
</tr>
<tr>
<td>Unused skills</td>
<td>Character plot hole</td>
<td>When the character of a story uses a skill that is not used again in the story while it would be logical to.</td>
<td>C learns that he or she has/obtains skill S1. C uses S1. S1 is never used again.</td>
<td>Superman learns that he can spin around the earth really fast to go back in time, yet he only uses this once.</td>
</tr>
<tr>
<td>Motivational change</td>
<td>Character plot hole</td>
<td>When a character of the story changes their motivation without any logical reason.</td>
<td>C has M1 After X amount of time C has M2</td>
<td>Peter wants to take over the world. He has an evil plan. He built a shrink ray. “This will surely get me a girlfriend!” Peter said.</td>
</tr>
<tr>
<td>Uncharacteristic event</td>
<td>Character plot hole</td>
<td>When a character of a story acts in a way that does not fit his or her personality</td>
<td>C has P1 but does something that is not characteristic for P1 but for P2 without explanation.</td>
<td>Tommy is extremely greedy. When his pirate group finds a treasure chest. He immediately offers to give his share to the other pirates.</td>
</tr>
<tr>
<td>Trait changes</td>
<td>Character plot hole</td>
<td>When a character in a story has a physical trait that is inconsistent with the rest of the story.</td>
<td>If the Ca does contain Tr1 and later it does not contain Tr1 or if Ca doesn’t contain Tr1 but later it does.</td>
<td>According to Boredpanda (n.d.) in the Harry Potter series Lily is seen with brown eyes but everyone describes Lily as having the same color eyes as Harry which is blue.</td>
</tr>
</tbody>
</table>
3.3 Plot holes represented in Scribbl

The plot holes that are represented in scribbl, according to the table above are: faulty time traveled 1, faulty time traveled 2, Unexplained: Object displacement and Trait changes. These specific plot holes were chosen because they were the most concrete and therefore are the easiest to program.

3.3.1 Faulty time traveled 1

Faulty time traveled is when a character cannot realistically reach location B from location A with their speed and in the timeframe given. A graph to represent this phenomenon would look as shown in figure 1. The ideal situation would be a case where the character arrives at Lb in exactly the time given. This however, is most of the time not the case as the character speed can vary. The character moves with by using the following formula: 

$$D = S \times t$$

Where D is the distance from La to Lb, S represents the character speed and t represents the time between two plot points.

If the character reaches Lb before the plot point where he or she is described at Lb, he or she is in the green area of the graph. If not, in the red. When the character is in the red area of the graph the faulty time traveled 1 plot hole occurs.
Figure 1: Faulty time traveled 1 graph
3.3.2 Faulty time traveled 2 and Unexplained: Object displacement

These two plot holes are relatively easy to program. They are in the same chapter because they use the almost exact same logic to determine if this plot hole occurs. To determine if this plot hole occurs, the program checks if the same object or character (determined by their object or character ID) are in a different place in two plot point that overlap. If this is the case a plot hole occurs. Figure 2 represents a graph where this is the case.

![Figure 2: Faulty time traveled 2 and Unexplained: Object displacement graph](image)

3.3.3 Trait changes

To check for a character that trait changes the user attributes a trait type to each trait. Then the user determines name of the trait and its definition. So for example: trait: eye color = “blue eyes”. The program will then check if all the types of traits have the same name for this character. If this is not the case a plot hole has occurred.
4. Domain analysis

According to Cheong et al. (2016) a domain model encodes knowledge in a particular domain, which in this specific case is made to represent our knowledge regarding plots and story structure. For this specific research a domain model was created to give an overview of the different parts that can be used to structure a story. This was made with the purpose of categorizing each component in order for the system to be able to find plot holes.

4.1 Parts

4.1.1 General story structure

In this section of the model essential story elements are shown. Without these elements the story does not exist. All of these elements were explained in chapter 2.2. and will therefore not be expanded upon further in this chapter.
4.1.2 Plot elements

As explained in 2.2.2 plot points are the events that impact the story. However for the purpose of this research we need to define a plot point through smaller elements that can be programmed into the system. A plot point is an event that occurs within a story. A plot point can consist of different plot point elements.

Plot point elements are divided in five categories, which are all types of plot point elements: Actions, characters, time, story objects and locations.

**Actions**

One of these elements are the actions that characters can take. Within a plot point multiple actions can be present. A character can perform multiple actions within a plot point and different characters can also perform different actions during a plot point.

**Characters**

Characters are all the characters present in a plot point. Each character can only be in a plot point once but can appear in multiple plot points, although not at the same time.

**Time**

An event should logically always have a start time and an end time. This is divided up into a start time, end time, start date and end date, of which each plot point can only possess one.

**Story objects**

Story objects are all the objects that appear in the story and are relevant. The MacGuffins fall under this section but also more general objects. For example, if a character throws a chair the chair becomes relevant for that plot point but the chair does not have to be a MacGuffin. A plot point may contain multiple story objects or none at all.

**Location**

The location is the place where the plot point takes place. Every plot point has to have exactly one location. A location may also have one location name.
4.1.3 Characters

Characters themselves are pretty complex structures. Each character has a physiology, sociology and psychology. A character also has a character archetype which refers to what type of character it is.

4.1.3.1 Character Physiology

A character always has a physiology which is the sum of all the elements that make up the character’s appearance.
Figure 6: Domain model - Character physiology
4.1.3.2 Character Sociology

A character also always has a sociology which consists of all the elements which make up the character’s social relationships as described by (Krawczyk and Novak, 2006)

*Figure 7: Domain model - Character sociology*
4.1.3.3 Character Psychology

The character’s psychology is made up of all the elements that describe a character’s psychology as described by Krawczyk and Novak (2006)

Figure 8: Domain model - Character Psychology
4.1.3.4 Character Archetypes

For this model all types of characters have been collected under the term “Character” because when modeling the system, the differences between the types are not relevant for outlining the system, but because there is a distinct difference when writing about them the different character types will be explained here briefly.

The protagonist is the hero of the story, it is the character that the story usually follows or the story is seen from this character’s point of view. It is possible for a story to have more than one protagonist (also known as a co-protagonist). The anti-hero is a type of protagonist but is not the typical hero. The anti-hero does not live a moral life and he often makes poor choices (Krawczyk and Novak, 2006).

Ancillary characters are the supporting characters. They can be either good, evil or something in between (Krawczyk and Novak, 2006).

The antagonist is the character who opposes the protagonist. An antagonist is not always evil, but does cause conflict. There are different types of antagonists. The exaggerated antagonist is a one-dimensional character and is (usually) a representation of evil in a cartoon-like fashion while the realistic antagonist is a more realistic representation where his/her intentions are not always clear. The mistaken antagonist is a character who appears to be the antagonist but will later on reveal itself as innocent and the transformational antagonist will develop itself throughout the story to become a different character type. (Krawczyk and Novak, 2006)

![Diagram of Character Archetypes](image)

*Figure 9: Domain model - Character archetypes*
4.1.4 Plot types

Plot types in this model have been displayed in the same way as they were explained in chapter 2.2.3. There are three types of plot types represented in this model although in reality there are probably more types. The three plot types that are represented are the Hero’s Journey, the episodic story structure and the three-act story structure.

Figure 10: Domain model - Plot types general

Figure 11: Domain model - Hero’s journey part 1

Figure 12: Domain model - Hero’s journey part 2
4.2 Plot hole types

The four types of plot holes used in this research, as shown in figure 13, are based on the plot holes listed by Miyamoto (2018):

- The MacGuffin plot hole
- The Logic plot hole
- The Character plot hole
- The Narrative plot hole

The plot holes represented here will be used as a base for chapter 4 where subcategories will be made for each of these plot holes. For the scope of this model cheap plot tricks will be excluded and because Ryan (2009) qualifies the Deus ex Machina as a cheap plot trick rather than a plot hole the Deus ex Machina will be excluded from this model as well.

*Figure 13: Four different types of plot holes*
5. The inner workings of Scribbl

Scribbl was made as a proof of concept to see if it is possible to create a tool that can detect plot holes in non natural language. It was decided to create a script-based language as described by Manning (2019) and to use parsers as described by Montfort, Marcus and Prince (2007).

5.1 Quality requirements for the system

There were different requirements that were kept in mind while creating the system. One of these requirements is that it can be expanded upon/modified by programmers who wish to adapt the system according to what they need. The system also needs to be capable of being installed by the user and they should be able to figure out relatively easy if they made a mistake while using the program. The desired target group should also be able to use this system effectively.

5.2 Choosing the environment

For the development of the proof of concept a few engines were considered. The most relevant engines that were considered were: Unity; Inform7; a few engines to make visual novels with such as Yarn or Ren’py; or just simply C#. The decision was made based on the needs of the project, which includes language processing and parsing, and the languages/engines I was most proficient with at the start of this project. These languages were mostly Java or Unity in combination with C#. The language played a major factor in choosing the engine because of the time constraints.

5.2.1 Choosing the IDE

At first Unity seemed like the preferred language, which partly had to do with that UI design would be relatively easy and that I was already relatively proficient in Unity. Even though I also had a little bit of knowledge of C# due to me generally using it in combination with Unity it was not my first choice right away because at first I was under the impression that C# would require me to create my own parsers as well. Since later the possibility of combining C# with ANTLR was found it was decided to make the switch to C#.

5.2.1.1 ANTLR

ANTLR is a tool created by Parr (n.d.). ANTLR stands for ANother Tool for Language Recognition. ANTLR generates a visitor and a parser. The parser can build parse trees and the visitor is used to look at each branch of the tree separately. ANTLR is created for the purpose of reading, processing, executing or translating structured text or binary files. The tool can be used to build languages, tools, or frameworks (Parr, n.d.).
ANTLR was chosen because it allows for quick prototyping and it does not need a language workbench. Another reason for choosing ANTLR was that there is no functional programming involved and that it is C# compatible.
5.3 System Overview

The whole tool, in short uses the .scribb language to define a story which it translates to a tree. This tree is based on the story structure described in chapter 2 and contains elements of the domain model in chapter 4. The program takes this tree and builds a story model from it. In this story model it checks if the user has given it correct input, if the story model was built correctly and also if the story model contains any plot holes. If it finds a fault in any of these aspects it will display this information in a pdf that is generated by the system.

5.3.1 Process diagram

In figure 14 the process that the program goes through is displayed. In this model all the square boxes represent input or output, the ovals represent modules or processes and the triangles are the generated objects.

The process starts off with the user. The user creates a .scribb file, which he or she inputs into the system. Then the user hits the start button and this is when the process starts. The .scribb file is parsed by the program and from that a tree is created. This tree is built and a story model is made. After the story model is made the system resolves by checking if there are no errors in building the story model. The system does this by giving back the input by getting it from the story tree. In the analyzing process the system will start analyzing and check if it sees any errors in the input. Then it will identify which kind of plot hole was found by analyzing the story model one more time. From this information it will output what kind of plot hole was found, an explanation of the location of that plot hole and an overview of the story to accompany it in the form of a pdf document.

![Process diagram](image)

*Figure 14: Process diagram, showing the different steps of the program*
At the same time if a meta user wants to expand upon the program he or she needs to add to the Scribbl.g4 document. In this document the basic logic of the system is defined which ANTLR uses to create its parse tree. Then the C# scanner and parser is generated from this parse tree. In this model the scanner and parser is displayed as a circle because this is the section of the program that actually executes the parse step in figure 16. After that the meta user can just use C# code to modify the program.

![Figure 16: Process diagram - meta user](image)

**Figure 16: Process diagram - meta user**
5.3.2 T-Model/Tombstone diagram

In figure 16, all the translations are visualized. The first translation is done by ANTLR which works on Java. Here the program (Scribbl.g4) is translated from a .g4 document to C# code. In the next step the Scribbl parser, which is now written in C# is translated to Scribbl MSIL (Microsoft intermediate language) by the C# compiler which runs on .NET.

The other part of the model is the interpreter, shown in figure 17. This describes that a .Scribbl document uses Scribbl code which runs on a .NET base.
5.3.3 The component model

The component model is another representation to describe the program Scribbl. In this model all the components that scribbl has and uses to function are displayed.

Scribbl uses ANTLR as a base. ANTLR generates a piece of C# code by using a .g4 file. The ANTLR.dll is a standard piece of code that is used by ANTLR as to not generate every piece of code. The C# code represents a tree. The order in which that tree is walked over is determined through a tree walker. A story model is used to contain all the story elements that were found when walking over said tree. Then the analyzer uses the information from the story model to scan for plot holes. This information is displayed back to the user in the form of text and a graph.
5.4 Implementation details

5.4.1 Defining the Scribbl.g4 file for Scribbl

For this project a Scribbl.g4 file was made. In this file all lexers and parsers that ANTLR uses are defined. It therefore contains all the rules for the language itself. In figure 19 some of the parse rules are shown that make up the language. The Scribbl.g4 file is the base of the program onto which all the other elements are built. The file consists of a combination of lexers and parsers.

```
grammar Scribbl;

plot
  : 'plot' plotname=STRING
    plotPoint* ;

plotPoint
  : 'plotPoint' plotPointName=STRING
    L CURLY
    plotPointElement* plotPointRelation*
    R CURLY ;

plotPointElement
  :
    IMAGE img=STRING // Display image (max 1)
    NAME name=STRING //plotPointElementName
    TEXT txt=STRING //plotPointElementText // Narrative
    CHARACTER
      LPAREN
      char=sourceID
      (COMMA name=STRING)?
      (COMMA img=STRING)?
      (COMMA speed=CHARSPEED)?
      (COMMA characterElement)*
      RPAREN
    //plotPointElementCharacter// Character participates at location of plot point
    LOCATION
      LPAREN
      loc=sourceID
      (COMMA locPlace=placeID)?
      RPAREN
    //plotPointElementLocation // Sets location of plot points
    STARTDATE
      d=DATE SPACE
t=TIME //plotPointElementStartDate// Plot point starts at a time of day
    ENDDATE
      d=DATE SPACE
t=TIME //plotPointElementEndDate // Plot point starts at a time of day
    UNIQUE?
    STORYOBJECT
      LPAREN
      obj=sourceID
      RPAREN
    //plotPointElementStoryObject// Object is used at location of plot point
    ;
```

Figure 19: Example of .g4 file (parser)
As stated in the chapter 2.5.3, lexers are capable of processing characters into tokens that the parser can use to transform the text into a tree. In figure 20, a snippet of the lexicals used in Scribbl.g4 can be seen. They define the variables that will later be used in both the parser and in the program itself.
// Defining the lexicals
// Basics
sourceID   : ID;
placeID    : ID;
LPAREN     : '(';
RPAREN     : ')';
COMMA      : ',';
LCURLY     : '{';
RCURLY     : '}';
DOT        : '.';
SPACE      : ' ';
EQUALS     : '=';

// Plot element names
LOCATION   : 'loc';
CHARACTER  : 'char';
STORYOBJECT: 'obj';

// Plot action names
GOTO       : 'goto';
ACHIEVE    : 'achieve';
SETGOAL    : 'setgoal';
CARRIES    : 'carries';
STARTTIME  : 'starttime';
ENDTIME    : 'endtime';
STARTDATE  : 'startdate';
ENDDATE    : 'enddate';
UNIQUE     : 'unique';
NAME       : 'name';

// Other essentials
TIME       : DIGIT DIGIT ':' DIGIT DIGIT (':' DIGIT DIGIT)?;
DATE       : DIGIT DIGIT '-' DIGIT DIGIT '-' DIGIT DIGIT DIGIT DIGIT;
ID         : LETTER (LETTER | DIGIT)*;
PLACEID    : LETTER (LETTER | DIGIT)*;
CHARSPEED  : DIGIT (DIGIT)? (DIGIT)? (DIGIT)? (DIGIT)? (DIGIT)? (DIGIT)?;
IMAGE      : 'image';
TEXT       : 'text';
LETTER     : ('a' .. 'z') | ('A' .. 'Z');
DIGIT      : ('0' .. '9');
WS         : [\t\r\n]+ \s+ -> skip;

STRING : "" (ESC | ~["\"\n])\s+ "";

fragment ESC : '\\' (["\"\bfnrt] | UNICODE);
fragment UNICODE : 'u' HEX HEX HEX HEX;
fragment HEX : [0-9a-fA-F];

Figure 20: Example of .g4 file (lexer)
As stated in the chapter 2.5.3 a parser can take a piece of text that has been through a lexer. The parse rules can also be seen in image 19. The parse rules determine what the input of the program should look like for the input to be understood. An example of a parse tree can be found in figure 21.
Figure 21: Story tree
5.4.2 Executing the Scribbl.g4 file

The Scribbl.g4 file gets turned into C# code by running it through the antlr4-4.7.2.jar program. This is done by a .bat file which runs by executing the code shown in figure 22.

```bash
@echo off
for %i in (C:\Users\ Scribbl.g4) do echo %i | findstr *.g4 | goto :loop
echo Compiling parser
echo done
echo done
```

*Figure 22: .bat script*

The code is constructed in such a way that when editing the Scribbl.g4 file the .bat can just be left running on the computer. When the Scribbl.g4 file is saved it automatically gets turned into C# code. While compiling it gives the message “Compiling parser”. It also shows you when it is finished compiling as shown in figure 23.
Figure 23: .bat running
5.4.3 Creating the builder/visitor override

ANTLR generates a visitor, however for the purposes of the program, it was decided that most of the classes in the visitor had to be overwritten in order to make to fully be able to exploit the benefits that we needed to use from ANTLR. The parser decides what the tree looks like but the visitor override actually builds the tree in the program. It consists of classes that check if there is already one instance with that ID made (if there is an ID present) and, if necessary, adds another object of that kind. It also checks which information is given by the author and makes an object based on that. An example of this is the character class shown in figure 24.

```java
public override object VisitPlanPointElementCharacter([NotNull] ScriptParser.PlanPointElementCharacterContext context)
{
    int createNew = true;


    foreach(var tempCharacter in characters)
    {
        if (context.character.ID().ToString() == tempCharacter.characterID().ToString())
        {
            createNew = false;
        }
    }

    if (createNew == true)
    {
        if (context.name == null)
        {
            character = new Character(sourceID, context.character.ID().GetText());
        }
        else if (context.name != null)
        {
            character = new Character(sourceID, context.character.ID().GetText(), context.name.Text);
        }
        else if (context.speed == null)
        {
            character = new Character(sourceID, context.character.ID().GetText(), context.name.Text, context.speed.Text);
        }
        else
        {
            character = new Character(sourceID, context.character.ID().GetText(), context.name.Text, context.speed.Text, context.character.Text);
        }
        characters.Add(character);
    }

    foreach(var characterElement in context.characterElement())
    {
        character.characterElements.Add(characterElement.Visit(characterElement));
    }

    planPointElementCollection.Add(character);
    return character;
}
```

*Figure 24: Example code of the Builder*
5.4.4 Creating the tree walker

The tree walker was made to check if the branches of the parse tree that the program returns are correct. To make this tree walker first a new visitor was made, as shown in figure 25. The tree walker visits every branch and is designed to return the information that the user has given the system. The purpose of this is to check if the system actually saves all the information that is given to it and can successfully access that information. In our system the class that functions as a tree walker is called the PrettyPrinter. After this class was finished and tested the location error finder and the plot hole finder were both based on this class.

5.4.5 Creating the logic error finder

The logic error finder checks if the input that the user has given them is correct and sufficient. This class looks at the input and checks if all the values it needs are present. If some values are not present it will present the user with a warning. If there is an object present twice and if an object is not present at all an error will appear. The system will present the errors in the form of blue text as shown in figure 26.

The following InputErrors were found:
ERROR there is no start date present in plot point: "InputErrorFinderTest"
ERROR there is no end date present in plot point: "InputErrorFinderTest"
ERROR there is no location present in plot point: "InputErrorFinderTest"
ERROR there is no character present in plot point: "InputErrorFinderTest"

Figure 26: example of input errors
5.4.6 Creating the plot hole finder

The plot hole finder checks if the input that is given by the user contains any plot holes. It can only do so accurately if the system does not contain any logic errors. The plot hole finder is constructed in the same way as the PrettyPrinter class and makes use of the same visitor class. To check for plot holes it uses the information in the tree to check if certain variables are true.

The system is currently able to recognize the following four plot holes, as described in chapter 3: Unexplained: Object displacement, Faulty time traveled 1, Faulty time traveled 2 and Trait changes.

If a plot hole is present the system will present a warning. This is because a plot hole is always allowed to be a conscious decision made by the author. The system will present the errors in the form of purple text as shown in figure 27.

WARNING: Plot hole found! The same character is found in two plot points at the same time!

Warning: character Princess has to travel from E3 to C1 within 00:10:00 time. the distance between E3 and C1 is 2,82842712474619km. Character Princess is able to travel with a speed of 2 km/h this means that character Princess can travel 0,333333333333333 km within that timeframe which is not be fast enough to reach C1 from E3 this error was found in line: 210

WARNING: this character has the characteristic race which is inconsistent in the rest of the story. This error occured in line:110

Figure 27: example of plot hole errors

If the system finds a Faulty time traveled 1 plot hole it will also generate a graph. This graph is represented in figure 28.
Figure 28: the Faulty time traveled 1 graph generated by the system
5.4.7 Other resources used in this project

For this project a few other resources were used. One of these resources includes ANTLR (Parr, n.d.). This includes the following two NuGet packages: The Antlr4.Runtime.Standard package and the Syncfusion.Pdf.WinForms package. The background images used for testing purposes were made by Cospigeon (2016, November). The sprite packs used in the test cases are from Paradisuu (2016) and Paradisuu (2015). One of the test cases also contains sprites and (shortened) text from the game Long Live the Queen by Hanako Games (2013).
5.5 Using the System

5.5.1 Input

Input can be done by generating a file in notepad or any other text generator that allows you to change the file extension to .scribbi. This file has to have a minimum of at least one plot, one plot point, one character, one location and end with the text “the end” to indicate the end of the input. The user can add his or her input by dragging a .scribbi file in the “Tests” folder. Through automated testing the program will automatically run through all of the .scribbi files in this folder and test them for plot holes when the play button is pressed.

5.5.1.1. Defining input

In this chapter the input for a .scribbi file will be described. An example of a complete .scribbi file can be found in Appendix A. If it is desired to check multiple files at the same time it is recommended for a .scribbi file to follow the following naming convention: “name” + alphabet letter + number. For example the first file would be “TestA1”, the next “TestA2” and nine files later the name would be “TestB1”. This is so that the program processes the files in the right order and the files do not get stuck in between.

Creating a new plot

The .scribbi file always starts off with the word plot followed by a space and the name between quotation marks. The plot ends off by the words “the end”. Each .Scribbi file may only contain one plot.

Creating a plot point

In a plot there is always a plot point present. The plot point is defined underneath the plot. Through plotPoint followed by a space and the name of the plot point between quotation marks. Underneath this the beginning and ending of the plot point is defined through a ‘{’ sign for the beginning and a ‘}’ for the ending of a plot point. A plot may contain multiple plot points. A plot point contains plot point elements.

Plot point elements

A plot point can have multiple different plot elements. It may contain one or more characters, locations or storyobjects. A plot point also always has one start date and one end date.

Characters

A character is a type of plot point element. If the user wants to define a character it can be done through typing “char” followed by a ‘{’ to represent the beginning of a character and a ‘}’ to represent the ending of a character. A character can have multiple character elements which include a character ID, a character name, an image, their speed and other character elements in that order. These elements are each separated by a comma.
The character ID can be a combination of letters and numbers and needs to be unique for each separate character. It may be the name of the character but this is not seen as the best practise in case of characters with multiple names. The name and the image are always represented within quotation marks. If the user wants to add more character attributes he or she can make a new one by starting with “ID=” and then directly after the new character attribute within quotation marks.

An example of a character would be as follows:

```
char(Princess,"Princess","1.png",ID="beautiful",ID="blue eyes")
```

**Actions**

An action may be defined through typing action followed by a ‘('. Then the first characterID involved with the action is given followed by a ‘,’. Then a string with the action taken. This is followed by another comma and the ID of the other character or story object involved with the action. Then the action gets closed off by adding a ‘)’.

An example of an action would be as follows:

```
action(Queen,"Promises hand in marriage of",Princess)
```

**Locations**

Locations in the system are a type of plot point element. A location always starts with the word “loc” followed by a ‘(’ to signify the elements that the location contains and a ‘)’ at the end to signify that there will be no more location elements after this. The location elements always start off with the location ID followed by a comma and a placeID. This placeID is represented in a map in the output and is used to calculate the distance between two locations. This is to signify distance from each space. Each square on the grid represents 1 km.

An example of a location would be as follows:

```
loc(Beach,A3)
```

**Story objects**

A story object is another plot point element. A story object always begins with the word “obj” followed by a ‘(’ to represent the beginning of the elements that the story object contains and a ‘)’ at the end to signify that there will be no more story objects after this. The only element a story object can currently contain is a sourceID.

An example of a story object would be as follows:

```
obj(TrinketsGold)
```

**Story text**

Here the user can represent what is actually happening in the scene. A story text is made by typing the word “story” followed by a ‘('. After this the user can type a string of text. To close the story the user has to type a ‘)’.
An example of a story text would be as follows:

```
storytext("There once lived an old queen whose husband had been dead for many years, and
she had a beautiful daughter. When the princess grew up she was promised in marriage to a
prince who lived far away.")
```

**Start and end dates**
The start and the end dates are also a type of plot point element. A start or end date always
begins with the word “startdate” or “enddate” respectively. After that comes a space and then
the date which is represented as follows: dd-mm-yyyy. Then a space and the time which is
represented through hours:minutes:seconds.

An example of a start date and and date would be as follows:

```
startdate 09-05-2019 13:20:00
enddate 10-02-2020 20:20:00
```

5.5.2 Running the program

To run the program, place the .scribbl file in the folder named “Tests”. Start up the Scribbl
application and run it. Then the output should appear in the “Generated PDF files” folder.

5.5.3 Output

The output of the system will consist of a pdf document with the same name as the related
.scribbl document and, as stated in 5.8.2 will appear in the “Generated PDF files” folder of the
program.

In this pdf the following line should be present: “This is the input that you gave me: ” followed by
the author’s input in the program.

Then there should be a line represented in blue that states: “The following InputErrors were
found: ”. If there are no input errors found underneath this line it will state “No input errors were
found, good job!”.

Underneath the input errors the program displays the following line: “The following plot holes
were found: “. If there were no plot holes found the program will simply state underneath this
line “No plot holes were found, good job!”.

Then on the next page is a grid which represents all the locations that were given to the system.
An example of such a map generated by the system is shown in figure 29.
Figure 29: A map generated by the system
5.5.4 Testing the system

The system was tested by two experts in the field. At the same time, these people were also our desired end users. The participants of this test were: Gabriele Ferri and Walter Piccolo. These participants will henceforth be referred to by their name when it is necessary to make a distinction between the two. The purpose of this test was to get the participant’s input on how to make the system more user friendly.

The test consisted of four phases:
- In the first phase they were asked how they would describe a plot hole themselves.
- In the second phase they were asked to sort the different aspects that currently exist in the program based on their order of importance.
- In the third phase the participants were guided through the system and were shown what the system could do. The purpose of this phase was to collect input from them as to what they would still need for the program to function and what they personally would like to see improved.
- In the fourth phase they were shown a paper prototype of a proposition for the next phase of the program. In this phase input was asked based on their expectations for the system.

5.5.4.1 The first phase

The first phase was asking the participant of the following question: “What is your definition of a plot hole?”.

During this phase we found that the participants were already aware of the definition of a plot hole. Gabriele Ferri described it as “A moment in the development of a story where the sequence of events is broken in a way that is not part of the artistic vision.”. From this we conclude that the system does not have to explicitly explain what a plot hole is to the user. More testing would have to be done to affirm this.

5.5.4.2 The second phase

In the second phase the participants were asked to sort the following system functionalities on their order of importance:
- I can view my input.
- I have an overview of tips on how to solve different types of plot holes.
- I have an overview of the different types of plot holes.
- I can see which types of plot holes are present in my story.
- I can see where my plot holes are located within my story.
- I can see how many plot holes are present.
- I can see a preview of what my story would look like.
- I can scroll through my different plot points.

Here it became apparent that the users found the scrolling through different plot points the most important and they found the preview of what their story would look like the second most important feature. Both found viewing their input one of the least important features. In Appendix C a picture of how they each sorted the functionalities can be found.

5.5.4.3 The third phase

In this phase we guided the participants through the system and asked for input on what they would still like to see from the system and what they would expect the software to be able to do to help the user detect plot holes.

Here it became apparent that they found the .pdf file very useful, however, they would like to see it be more organized. They liked the fact that there were warnings presented instead of errors. They found seeing that their own input was less important.

5.5.4.4 The fourth phase

In the fourth phase we asked the participant questions based on the paper prototype. These questions were mostly based on the buttons shown, what the participant thought the button would do if you clicked on it and if he/she was missing something that he/she thinks the system would be able to do.

Here we found that the users expected a step-by-step tutorial, graphs and a customizable interface.

5.4.5 Changes made after testing the system

In the second phase of testing we found that the testers found scrolling through the different plot points the most important. To enable this we have made the system create a .html document for each plot point. This html document enables the user to scroll through their plot points and get a visual representation of what their visual novel would look like. The findings in the fourth phase were not implemented because of time constraints.
“Elodia, Crown Princess: My father says I can’t have the crystal until I’m Queen, so we’ll just have to wait. Julimna, Duchess of Urel: There may yet be time to wait. Julimna, Duchess of Urel: This world is filled with dangers, and you will need the power of a Dragon to fight them. You can’t begin training until you have bonded with your crystal. Elodia, Crown Princess: Well, what else am I supposed to do? Julimna, Duchess of Urel: The crystal is probably held under guard in the Royal Treasury. You are the prince; you should be able to find a way inside. Julimna, Duchess of Urel: Otherwise we might have to take drastic action. Elodia, Crown Princess: All right. I will try to get into the treasury.”

Figure 30: An example of a generated .html file
6. Contributions and future work

6.1 Advantages and limitations of the system

6.1.1 Theoretical contributions

The valuable elements of this thesis consist of the domain analysis, the categorization of different types of plot holes and the formalisation of a textual language for visual novels which can analyze four different types of plot holes (Scribbl). This thesis expands on the work of Krawczyk and Novak (2006), Ryan (2009) and van der Geest (2015).

Another reason why this research is valuable to the research group is because for this specific research group this is the first step into plot holes and plot hole detection. This research is also closely connected to the work of other people within the research group. Therefore, if they want to expand upon this research subject, my desk research and the problems I have encountered would help them in their future studies of the subject. One of the goals of this research group is to use the research and apply it to the field through software. This is where Scribbl is a useful addition to the knowledge of the research group.

6.1.2 Practical contributions

The first, and most obvious, advantage of this system is that it can currently successfully detect the following four plot holes as described in chapter 4: Unexplained: Object displacement, Faulty time traveled 1, Faulty time traveled 2 and Trait changes. It can generate a graph for the first type of plot hole and it can generate a preview of a visual novel.

Another advantage is that the tool itself was written keeping future work in mind and was therefore made to be very modular and can easily be expanded upon. It could also be very easily be expanded, not only to detect more plot holes but also to represent different types of games. In this research our primary focus was visual novels because of their great focus on storyline but since plot holes could be present in any game with a story the technique to detect plot holes used in this thesis could also easily be applicable to other types of games such as adventure role playing games. For example, in a story based lock and key mechanism.
6.2 Tool limitations

The tool currently has a few limitations. One of these limitations is that the tool does currently not support the use of natural languages and it requires the user to specify their story in a very specific manner.

The tool also does not cover the many types of plot holes specified in chapter 4 yet since different plot holes are very specific and work on different parts of the story some are very difficult to program. It also does not support teleportation or transport by other means than the basic character speed since it will flag this as a plot hole and the tool currently does not support storylines with twines yet.
6.3 Advice for future work

As for future work my advice would be to expand upon this tool and do more research upon detecting plot holes. There is much that remains unknown about the subject and therefore could be expanded upon.

Within the tool there are several features that could be expanded upon. One of those features is the addition of more types of plot holes. Another feature that could be expanded upon is adding the use of branching storylines to the tool. Making the user capable of using natural language as input rather than the .scribbl language would greatly improve the usability of the tool for the users. The usability could also be improved by creating a better user interface where all the elements are not separated in folders, are more easily accessible to the user and with the elements described in 5.5.4.4.

In the future this tool might be integrated into a system which can generate storylines in, for example, a dungeon generator where it would be useful to not have to check for plot holes by hand or a visual novel with a lot of generated twines. This would make it possible to detect generated storylines on plot holes. It could also be used in checking storylines of a large scale or large complexity, which might make visual novels with a lot of twines more viable. In the future the tool might also be capable of repairing plot holes.
7. References


https://dl.acm.org/citation.cfm?id=1624452


https://screencraft.org/2018/03/09/do-you-know-the-five-different-types-of-plot-holes/


Appendix A .scribbl example

In this appendix an example of what a .scribbl file can contain. This example was made based on the game long live the queen made by Hanako games (2013). In this file four plot holes can be found. The output of this file can be found in appendix B.

```
plot "Long live the queen example"
  plotPoint "Week 0"
  {
    plotPointImage("background.jpg")
    plotPointName(name "Week 1")

    storytext("Joslyn, King Dowager, Duke of Caloris: We're almost home. Your room is just the way you left it.
    Elodie, Crown Princess: ...
    Joslyn, King Dowager, Duke of Caloris: I know it's hard to leave your school and all your friends, but I've arranged the best possible tutors for every subject.
    Joslyn, King Dowager, Duke of Caloris: You'll have to work hard this year to prepare yourself before your fifteenth birthday, but I know you can do it.
    Joslyn, King Dowager, Duke of Caloris: You'll learn quickly and you'll make a wonderful queen. It's what your mother would have wanted.
    Elodie, Crown Princess: This is not what Mother would have wanted!
    Elodie, Crown Princess: She wouldn't have wanted to die and leave me.
    Joslyn, King Dowager, Duke of Caloris: No. She wouldn't. But sometimes bad things happen. We have to pick up and carry on.
    Joslyn, King Dowager, Duke of Caloris: All of Nova depends on us. On you.
    Elodie, Crown Princess: "sigh"
    Joslyn, King Dowager, Duke of Caloris: I will be here to guide you until your coronation, but the decisions you make are ultimately up to you.
    Joslyn, King Dowager, Duke of Caloris: Come. Your maids are waiting.")

    loc(Castle,E2)

    char(Princess,"Elodie","Elodie.png", speed : 2,trait: status = "crown princess",trait: eyes = "blue eyes")
    char(King,"Joslyn","Joslyn.png", speed : 2,trait: occupation = "King",trait: personality = "kind")
    action(Princess," talks to ", King)

    startdate 20-02-2020 12:20:00
    enddate 27-02-2020 12:20:00
  }
```

storytext("Elodie, Crown Princess: What's that commotion outside? Father? What's going on? Julianna, Duchess of Ursul: I have come to pay my respects in honor of your mother. Joslyn, King Dowager, Duke of Caloris: You have no respect or honor. It's because of you and your powers that my wife is dead! Elodie, Crown Princess: What?!? Julianna, Duchess of Ursul: I am a Lumen, as was she, and like every Lumen I am willing to give my life if necessary to protect our domain. Joslyn, King Dowager, Duke of Caloris: Your line has been traitors for two hundred years. I won't have you corrupting my daughter. Julianna, Duchess of Ursul: That is for the Crown Princess to decide. Julianna, Duchess of Ursul: Elodie, your mother trusted me, and I have much to teach you. Elodie, Crown Princess: I want to hear what she has to say. Elodie, Crown Princess: She stays. Joslyn, King Dowager, Duke of Caloris: As you wish. Elodie, Crown Princess: What is it you wanted to teach me? Julianna, Duchess of Ursul: First, I need to know if you have your mother's crystal. Elodie, Crown Princess: What crystal? Julianna, Duchess of Ursul: The source of power for any Lumen is a particular magical crystal. Once you've bonded with it, it becomes a part of you as long as you live. Julianna, Duchess of Ursul: The Royal Crystal has been passed down from ruler to ruler for generations. It belongs to you, now - except that the King may be keeping it from you. Elodie, Crown Princess: Do you have a crystal? Julianna, Duchess of Ursul: Of course. She holds out her hands. Light flickers above her bosom, and then, with a shower of sparkles, something takes shape. You reach out to touch it, and it dissolves away into nothing. Elodie, Crown Princess: Oh. Julianna, Duchess of Ursul: It belongs to me. No one can take it from me until I die. Julianna, Duchess of Ursul: You need to find the crystal that belongs to you.")

loc(Castle,E2)

char(Princess,"Elodie","Elodie.png",speed : 2,trait: status = "crown princess",trait: eyes = "blue eyes")
char(DuchessofUrsul,"Julianna","Julianna.png",speed : 2,trait: title = "Duchess",trait: race = "lumen")
char(King,"Joslyn","Joslyn.png",speed : 2,trait: occupation = "King",trait: personality = "kind")
action(Princess," talks to ", DuchessofUrsul)

startdate 06-03-2020 13:20:00
enddate 13-03-2020 13:20:00
}
plotPoint "Week 3"
{
  plotPointImage("background.jpg")
  plotPointName(name "Week 3")
}

storytext("Elodie, Crown Princess: Dad, what happened to Mom's Lumen crystal?  
Joslyn, King Dowager, Duke of Caloris: It's in a safe place.  
Elodie, Crown Princess: Can I have it?  
Joslyn, King Dowager, Duke of Caloris: No.  
Elodie, Crown Princess: But it's mine!  
Joslyn, King Dowager, Duke of Caloris: Meddling with magic killed your mother. I don't want that to happen to you.  
Joslyn, King Dowager, Duke of Caloris: Once you're crowned Queen, I can't stop you from doing anything you want, but for now you are still a child.
")

loc(Castle,E2)

char(Princess,"Elodie","Elodie.png",speed : 2,trait: status = "crown princess",trait: eyes = "blue eyes")
char(King,"Joslyn","Joslyn.png",speed : 2,trait: occupation = "King",trait: personality = "kind")

action(Princess," talks to ", King)

startdate 13-03-2020 13:20:00
enddate 20-03-2020 13:20:00
}

plotPoint "Week 5"
{
  plotPointImage("background.jpg")
  plotPointName(name "Week 5")
}

storytext("Elodie, Crown Princess: My father says I can't have the crystal until I'm Queen, so we'll just have to wait.  
Julianna, Duchess of Ursul: There may not be time to wait.  
Julianna, Duchess of Ursul: This world is filled with dangers, and you will need the powers of a Lumen to fight them. You can't begin training until you have bonded with your crystal.
")
Elodie, Crown Princess: Well, what else am I supposed to do about it?
Julianna, Duchess of Ursul: The crystal is probably held under guard in the Royal Treasury. You are the princess; you should be able to find a way inside.
Julianna, Duchess of Ursul: Otherwise we might have to take drastic action.
Elodie, Crown Princess: All right. I will try to get into the treasury.

loc(Castle,E2)

char(Princess,"Elodie","Elodie.png",speed : 2,trait: status = "crown princess",trait: eyes = "blue eyes")
char(DuchessofUrsul,"Julianna","Julianna.png",speed : 2,trait: title = "Duchess",trait: race = "human")

action(Princess," talks to ", DuchessofUrsul)

startdate 27-03-2020 13:20:00
enddate 03-04-2020 13:20:00

plotPoint "Week 6"
{
    plotPointImage("background.jpg")
    plotPointName(name "Week 6")
}

storytext("You overwhelm the guards with facts and figures, explaining that you need treasury access to adjust the royal budget, which may affect their pay packets as well as the taxes assessed on their friends and family. They stand aside.
The glitter is dazzling, but some sixth sense draws you towards one little locked box in particular. This is what you need. This is the crystal that will make you a Lumen. You tuck the box into your pocket. You can show it to your mentor later and get instructions on what to do next.")

loc(Castle,E2)

char(Princess,"Elodie","Elodie.png",speed : 2,trait: status = "crown princess",trait: eyes = "blue eyes")

action(Princess," talks to ", Guards)

startdate 03-04-2020 13:20:00
enddate 10-04-2020 13:20:00
"}
Elodie, Crown Princess: I got it!
Elodie, Crown Princess: What do I do now?
Julianna, Duchess of Ursul: Pick up the crystal with your bare hands. Press it against your chest, over your heart, and say "Illuminate!"
Elodie, Crown Princess: Okay... Here we go... Illuminate!
You feel dizzy, and both larger and lighter than you were before. The crystal has disappeared from view, but you know where it is.
Julianna, Duchess of Ursul: The priestess here is a friend of mine. I will come to your lessons in the faith and train you to use your powers.

Joslyn, King Dowager, Duke of Caloris: Are you ready for the grand ball?
Joslyn, King Dowager, Duke of Caloris: All the nobles in the domain are here to see you - to see their queen.
Elodie, Crown Princess: (gulp!)
You finish dressing and descend the stairs to make a grand entrance.
All around, the rich and powerful pause in their activities to gaze upon you, the ruler of them all. Seeing yourself reflected in so many eyes makes you want to run away and hide.
After a moment, you force yourself to carry on, but you're sure they've all seen you hesitate.
Your father waits for you at the bottom of the stairs and offers you his arm.
Joslyn, King Dowager, Duke of Caloris: The first dance is for us.
He guides you gently around the dance floor, never rushing you. It's fun to dance with your
teacher, but the look in his eyes is so sad.
Everyone applauds politely.
As the gala continues, you take the opportunity to observe nobles that you rarely see.
Strange that there's no sign of your cousins, though. Shouldn't they be here?
Your aunt and uncle are here, of course. It would be scandalous if they hadn't come, Merva
being so close by.
There's something itching at your senses, something familiar.
Then it hits you. Aunt Lucille is a Lumen!
Elodie, Crown Princess: Aunt Lucille, may I speak with you privately?
Lucille, Countess of Nix: Of course, dear.
You withdraw to a small side room.
Elodie, Crown Princess: I can sense your powers. You're a Lumen.
Lucille, Countess of Nix: What?!
Elodie, Crown Princess: I am, too. That's how I know. Can't you sense that?
Lucille, Countess of Nix: No. No, I can't.
She looks down, shaking her head.
Lucille, Countess of Nix: I... I do have a Lumen crystal. I found it, years ago. It called to me. But
I don't know anything about magic.
Lucille, Countess of Nix: I can make the crystal appear and disappear, and I can block out
sound with it. That's all.
Elodie, Crown Princess: (I don't think that's true at all.)
Elodie, Crown Princess: (There are traces of old magic around her, too faint to make sense of,
but strong.)
Lucille, Countess of Nix: Magic is dangerous, everyone knows that. People might think I was an
evil witch. They wouldn't trust me.
Lucille, Countess of Nix: Look at how much trouble the Duchess of Ursul has. Please, Your
Highness, don't tell anyone.
Elodie, Crown Princess: What I think I should tell everyone is that you are my new Lumen
Minister.
Lucille, Countess of Nix: What?
Elodie, Crown Princess: Lumens have power. If you train with me, you'll become even stronger.
Elodie, Crown Princess: I want that strength here to defend the capital.
Elodie, Crown Princess: You'll have status and recognition and no one will dare to call you a
witch. What do you say?
Lucille, Countess of Nix: I accept.
Elodie, Crown Princess: Good!
It's nice to be able to enjoy time with friends and family, isn't it?"
char(Princess,"Elodie","Elodie.png",speed : 2,trait: status = "crown princess",trait: eyes = "blue eyes")
char(King,"Joslyn","Joslyn.png",speed : 2,trait: occupation = "King",trait: personality = "kind")
char(CountessofNix,"Lucille","Lucille.png",speed : 2,trait: general = "secretly evil")

action(Princess," talks to ", CountessofNix)
action(Princess,"dances with", King)

startdate 26-06-2020 13:20:00
enddate 10-07-2020 13:20:00

}  
plotPoint "Week 20"
{
  plotPointImage("background.jpg")
  plotPointName(name "Week 20")

storytext("Joslyn, King Dowager, Duke of Caloris: You are requested to stand in judgement. A man has been convicted of the murder-by-strangling of his wife.
Joslyn, King Dowager, Duke of Caloris: He does not deny the act, but requests a pardon that he might be set free.
Elodie, Crown Princess: If you admit you killed your wife, why do you think I'm going to set you free?
Condemned Man: Save me, Your Majesty! Wasn't my fault. Demons made me do it.
Condemned Man: Everyone knows the power of magical beasties! They used me, they twisted my fingers into chains...
Condemned Man: My wife found me screaming, she tried to shake me, and the chains wrapped around her...
Condemned Man: I need the priestesses to bless me and make me clean again!
Elodie, Crown Princess: (Demons? This man is a Lumen! Where did he get a crystal?)
Elodie, Crown Princess: It was a demon's magic that killed your wife? Not your own magic?
Condemned Man: ... So you know what I am.
He throws out his hands, and his fingertips elongate into floating strings of gold.
Condemned Man: DIE!
Golden chains slash at you like whips.
With so much warning, of course, it is easy for you to resist. His attack bounces harmlessly off your shields, the chains flying back in the direction from which they came... where they find an easier target.
Before your eyes, the renegade Lumen is choked to death by his own powers.
After a moment, an orange crystal materialises next to his body.
Elodie, Crown Princess: I guess I'll have to take that for safe keeping.

75
You creep through the castle dungeons, catching glimpses here and there of the fate that awaits troublemakers in this kingdom. Even nobles can end up here, sometimes. It's a good reminder to stay on the straight and narrow.

loc(Garden,C1)

char(Princess,"Elodie","Elodie.png",speed : 2,trait: status = "crown princess",trait: eyes = "blue eyes")
char(King,"Joslyn","Joslyn.png",speed : 2,trait: occupation = "King",trait: personality = "kind")
char(CondemnedMan,"Condemned man",speed : 2,trait: hostilityLevel = "dangerous")

action(King, " talks to ", Princess)
action(CondemnedMan, " talks to ", Princess)

startdate 10-07-2020 13:30:00
denddate 17-07-2020 13:20:00
}

plotPoint "Week 21"
{
  plotPointImage("background.jpg")
  plotPointName(name "Week 21")
}

storytext("Elodie, Crown Princess: If you attack an enemy Lumen, and you win, and they leave behind a crystal...
Elodie, Crown Princess: Is it possible to combine that crystal with mine to make it stronger?
Julianna, Duchess of Ursul: Each Lumen can hold only one crystal.
Julianna, Duchess of Ursul: The only way to combine their magic is if a living wielder chooses to join its power to another.
Elodie, Crown Princess: Oh.
Elodie, Crown Princess: So I should destroy other crystals, to stop anyone from using them against me.
Julianna, Duchess of Ursul: That is the traditional policy - but it is a lie. Lumen crystals cannot be destroyed.
Julianna, Duchess of Ursul: Rulers claim otherwise to discourage thieves from seeking dangerous treasures.
Elodie, Crown Princess: What am I supposed to do with a crystal, then?
Julianna, Duchess of Ursul: Guard it, or hide it, or give it into other hands to wield.
Julianna, Duchess of Ursul: But if the last holder of the crystal was an enemy, the resonance of the crystal will work against you.
Julianna, Duchess of Ursul: It will be attuned to a personality as foolish or violent or scheming as the Lumen you defeated.
Elodie, Crown Princess: So anyone taking the crystal would probably become a new enemy. That's useless!
Julianna, Duchess of Ursul: A weak-willed candidate might be acceptable to the crystal and yet still controllable."

76
Julianna, Duchess of Ursul: You could then steer that Lumen away from dangerous paths, making an ally and changing the crystal's resonance. Julianna, Duchess of Ursul: It is, however, a dangerous endeavor and one that requires your complete attention. Elodie, Crown Princess: (Not really a project I can work on while I'm so busy studying for my coronation.) Elodie, Crown Princess: (Well, it's only one crystal. That's not too much trouble.)

loc(Castle,E2)

cchar(Princess,"Elodie","Elodie.png",speed : 2,trait: status = "crown princess",trait: eyes = "blue eyes")
cchar(DuchessofUrsul,"Julianna","Julianna.png",speed : 2,trait: title = "Duchess",trait: race = "lumen")

taction(Princess," talks to ", DuchessofUrsul)

startdate 17-07-2020 13:20:00
deenddate 24-07-2020 13:20:00
}

plotPoint "Week 40"
{
plotPointImage("background.jpg")
plotPointName(name "Week 40")

storytext("At this time last year, you were celebrating your fourteenth birthday. You were in the school gardens, surrounded by your friends. A teacher brought you tea and cakes, while a wealthy merchant's son wove a crown of flowers for your head. It didn't matter so much that you were a princess then. Your title was something for the future. Many of your peers would be Duchesses or Earls or the like, someday, but not then. You were children. Your parents could not attend on the actual day, but they did send wonderful gifts, some for you, and some for you to share. And a week later, they came for a visit, and your mother took you with her through the countryside in a splendid carriage. It was the last time you would ever see her. You wonder if, wherever she is, she can see you now.

You are fifteen years old, a legal adult. You have worked and studied and suffered and prepared, and now the time has come. You kneel before the priestess, barely hearing her words as she recites the blessings. She calls upon the gods to deliver peace, wisdom, and prosperity to you, and through you, to all of Nova. And then she calls upon you, for your oath of rulership. Priestess of the First Circle: Will you guide and govern and protect your people to the best of your ability, according to law and custom? Elodie, Crown Princess: I will. Priestess of the First Circle: Will you to the best of your power uphold the ideals of love, honor, justice, and mercy? Elodie, Crown Princess: I will. Priestess of the First Circle: Lords and Ladies assembled: I present to you your undoubted Queen, who has sworn you her loyalty. Priestess of the First Circle:

77
Circle: You who have come to give homage, will you do the same? One at a time, the head of each duchy approaches your throne and kneels to swear his or her service to you and your heirs. Priestess of the First Circle: People of Nova, I give you Elodie, daughter of Fidelia, your true sovereign. What say you all? People: LONG LIVE THE QUEEN! LONG LIVE THE QUEEN! LONG LIVE THE QUEEN! Elodie, Queen of Nova: (Thank you! I will!!) Once his daughter was secure on her throne, Joslyn returned to his birthplace to focus on his duties as Duke of Caloris. As Lumen Minister, Lucille requisitioned the orange crystal that Elodie had taken from the dead criminal and offered it to her daughter. However, Charlotte was not able to activate the crystal's powers. The young Queen pointed out that a crazed murderer would obviously have a very different resonance than her favorite cousin. Charlotte would have to wait to inherit her mother's magic. Elodie's interest in trade and economics led her to treat country management as something of a game, and a game she was very good at. Why waste lives conquering when you can dominate by natural superiority? Her manipulations increased Nova's ties with its neighbors and raised the standard of living for all of its citizens. And thus Queen Elodie's legacy stretched into the future..."

loc(Castle,E2)

char(Princess,"Elodie","Elodie.png",speed : 2,trait: status = "crown princess",trait: eyes = "blue eyes")
char(King,"Joslyn","Joslyn.png",speed : 2,trait: occupation = "King",trait: personality = "kind")
char(CountessofNix,"Lucille","Lucille.png",speed : 2,trait: general = "secretly evil")

startdate 09-10-2020 13:20:00
enddate 16-10-2020 13:20:00
}
Appendix B output example

This is the pdf output of the .scribbl file represented in appendix A.

```plaintext
This is the input that you gave me:
1. plotPoint "Week 0"
2. {
3.   plotPointImage("background.jpg")
4.   plotPointName(name "Week 1")
5. }
6.   storytext(Justyn, King Dowager, Duke of Caloris: We're almost home. Your room is just the way you left it.
7.   Elodie, Crown Princess: ...
8.   Justyn, King Dowager, Duke of Caloris: I know it's hard to leave your school and all your friends, but I've arranged the best possible tutors for every subject.
9.   Justyn, King Dowager, Duke of Caloris: You'll have to work hard this year to prepare yourself before your fifteenth birthday, but I know you can do it.
10.  Justyn, King Dowager, Duke of Caloris: You'll learn quickly and you'll make a wonderful queen. It's what your mother would have wanted.
11.   Elodie, Crown Princess: This is not what Mother would have wanted!
12.   Elodie, Crown Princess: She wouldn't have wanted to die and leave me.
13.   Justyn, King Dowager, Duke of Caloris: No. She wouldn't. But sometimes bad things happen. We have to pick up and carry on.
15.   Elodie, Crown Princess: "sigh"
16.   Justyn, King Dowager, Duke of Caloris: I will be here to guide you until your coronation, but the decisions you make are ultimately up to you.
18. }
19. loc(Castle,E2)
20. char(Princess,"Elodie","Elodie.png", speed : 2, trait: status = "crown princess", trait: eyes = "blue eyes")
21. char(King,"Justyn","Joslyn.png", speed : 2, trait: occupation = "King", trait: personality = "kind")
22. action(Princess:"talks to", King)
23. {
24.   startdate 20-02-2020 12:20:00
25.   enddate 27-02-2020 12:20:00
26. }
27. plotPoint "Week 2"
28. {
29.   plotPointImage("background.jpg")
30.   plotPointName(name "Week 2")
31. }
33.   Justyna, Duchess of Ursul: I have come to pay my respects in honor of your mother.
34.   Justyn, King Dowager, Duke of Caloris: You have no respect or honor. It's because of you and your power that my wife is dead!
36.   Justyna, Duchess of Ursul: I am a Lumen, as was she, and like every Lumen I am willing to give my life if necessary to protect our domain.
37.   Justyn, King Dowager, Duke of Caloris: Your line has been traitors for two hundred years. I won't have you corrupting my daughter.
38.   Justyna, Duchess of Ursul: That is for the Crown Princess to decide.
39.   Justyna, Duchess of Ursul: Elodie, your mother trusted me, and I have much to teach you.
```
Elodie, Crown Princess: I want to hear what she has to say.
Elodie, Crown Princess: She says.
Joslyn, King Dowager, Duke of Caloris: As you wish.
Elodie, Crown Princess: What is it you wanted to teach me?
Juliana, Duchess of Ursul: First, I need to know if you have your mother's crystal.
Elodie, Crown Princess: What crystal?
Juliana, Duchess of Ursul: The source of power for any Lumen is a particular magical crystal. Once you've bonded with it, it becomes a part of you as long as you live.
Juliana, Duchess of Ursul: The Royal Crystal has been passed down from ruler to ruler for generations. It belongs to you, now - except that the King may be keeping it from you.
Elodie, Crown Princess: Do you have a crystal?
Juliana, Duchess of Ursul: Of course.
She holds out her hand. Light flickers above her bosom, and then, with a shower of sparkles, something takes shape.
You reach out to touch it, and it dissolves away into nothing.
Elodie, Crown Prince: Oh.
Juliana, Duchess of Ursul: It belongs to me. No one can take it from me until I die.
Juliana, Duchess of Ursul: You need to find the crystal that belongs to you.

```
loc(Castle,E2)
char(Princess,"Elodie","Elodie.png",speed : 2,trait: status = "crown princess",trait: eyes = "blue eyes")
char(DuchessofUrsul,"Juliana","Juliana.png",speed : 2,trait: title = "Duchess",trait: race = "lumen")
char(King,"Joslyn","Joslyn.png",speed : 2,trait: occupation = "King",trait: personality = "kind")
action(Princess," talks to ",DuchessofUrsul)
startdate 06-03-2020 13:20:00
denddate 13-03-2020 13:20:00
```
plotPoint "Week 3"
plotPointImage("background.jpg")
plotPointName(name "Week 3")
storytext("Elodie, Crown Princess: Dad, what happened to Mom's Lumen crystal?
Joslyn, King Dowager, Duke of Caloris: It's in a safe place.
Elodie, Crown Princess: Can I have it?
Joslyn, King Dowager, Duke of Caloris: No.
Elodie, Crown Princess: But it's mine!
Joslyn, King Dowager, Duke of Caloris: Meddling with magic killed your mother. I don't want that to happen to you.
Joslyn, King Dowager, Duke of Caloris: Once you're crowned Queen, I can't stop you from doing anything you want, but for now you are still a child.
")
loc(Castle,E2)
char(Princess,"Elodie","Elodie.png",speed : 2,trait: status = "crown princess",trait:}
eyes = "blue eyes")
38. char(King,"Jaslyn"."Jaslyn.png".speed : 2.trait: occupation = "King".trait: personality = "kind")
39. 
40. action(Princess," talks to ", King)
41. 
42. startdate 13-03-2020 13:20:00
43. enddate 20-03-2020 13:20:00
44. }
45. 
46. plotPoint "Week 5"
47. {
48. plotPointImage("background.jpg")
49. plotPointName(name = "Week 5")
50. 
51. storytext("Elodie, Crown Princess: My father says I can't have the crystal until I'm Queen, so we'll just have to wait.
52. Julianna, Duchess of Ursul: There may not be time to wait.
53. Julianna, Duchess of Ursul: This world is filled with dangers, and you will need the powers of a Lumen to fight them. You can't begin training until you have bonded with your crystal.
54. Elodie, Crown Princess: Well, what else am I supposed to do about it?
55. Julianna, Duchess of Ursul: The crystal is probably held under guard in the Royal Treasury. You are the princess; you should be able to find a way inside.
56. Julianna, Duchess of Ursul: Otherwise we might have to take drastic action.
57. Elodie, Crown Princess: All right. I will try to get into the treasury.")
58. 
59. loc(Castle,E2)
60. 
61. char(Princess,"Elodie"."Elodie.png".speed : 2.trait: status = "crown princess".trait: eyes = "blue eyes")
62. char(DuchessOfUrsul,"Julianna"."Julianna.png".speed : 2.trait: title = "Duchess".trait: race = "human")
63. 
64. action(Princess," talks to ", DuchessOfUrsul)
65. 
66. startdate 27-03-2020 13:20:00
67. enddate 03-04-2020 13:20:00
68. }
69. 
70. plotPoint "Week 6"
71. {
72. plotPointImage("background.jpg")
73. plotPointName(name = "Week 6")
74. 
75. storytext("You overwhelm the guards with facts and figures, explaining that you need treasury access to adjust the royal budget, which may affect their pay packets as well as the taxes assessed on their friends and family. They stand aside.
76. The glitter is dazzling, but some sixth sense draws you towards one little locked box in particular. This is what you need. This is the crystal that will make you a Lumen.
77. You tuck the box into your pocket. You can show it to your mentor later and get instructions on what to do next.
78. ")
79. 
80. 
81
loc(Castle.E2)

car(Princess."Elodie","Elodie.png",speed : 2,trait: status = "crown princess",trait: eyes = "blue eyes")

action(Princess," talks to ", Guards)

startdate 03-04-2020 13:20:00
enddate 10-04-2020 13:20:00
}

plotPoint "Week 7"
{
plotPointImage("background.jpg")
plotPointName(name = "Week 7")

storytext("Week 7
Elodie, Crown Princess: I got it!
Elodie, Crown Princess: What do I do now?
Juliana, Duchess of Ursul: Pick up the crystal with your bare hands. Press it against your chest, over your heart, and say 'Illuminate!'
Elodie, Crown Princess: Okay... Here we go... Illuminate!
You feel dizzy, and both larger and lighter than you were before. The crystal has disappeared from view, but you know where it is.
Juliana, Duchess of Ursul: The priestess here is a friend of mine. I will come to your lessons in the faith and train you to use your powers."

loc(Castle.E2)


char(Princess."Elodie","Elodie.png",speed : 2,trait: status = "crown princess",trait: eyes = "blue eyes")

char(DuchessofUrsul,"Juliana","Juliana.png",speed : 2,trait: title = "Duchess",trait: race = "lumen")

action(Princess," talks to ", DuchessofUrsul)

startdate 00-04-2020 13:20:00
enddate 17-04-2020 13:20:00
}

plotPoint "Week 17"
{
plotPointImage("background.jpg")
plotPointName(name = "Week 17")

storytext("Joslyn, King Dowager, Duke of Calors: Are you ready for the grand ball?
Joslyn, King Dowager, Duke of Calors: All the nobles in the domain are here to see you - to see their queen.
Elodie, Crown Princess: (gulp!) You finish dressing and descend the stairs to make a grand entrance.
All around, the rich and powerful pause in their activities to gaze upon you, the ruler of them all.
Seeing yourself reflected in so many eyes makes you want to run away and hide.

82
174. After a moment, you force yourself to carry on, but you’re sure they’ve all seen you hesitate.
175. Your father waits for you at the bottom of the stairs and offers you his arm.
176. Joslyn, King Dowager. Duke of Caloria: The first dance is for us.
177. He guides you gently around the dance floor, never rushing you. It’s fun to dance with your father, but the look in his eyes is so sad.
178. Everyone applauds politely.
179. As the gala continues, you take the opportunity to observe nobles that you rarely see.
180. Strange that there’s no sign of your cousins, though. Shouldn’t they be here?
181. Your aunt and uncle are here, of course. It would be scandalous if they hadn’t come, Merva being so close by.
182. There’s something itching at your senses, something familiar.
183. Then it hits you. Aunt Lucille is a Lumen!
184. Elodie, Crown Princess: Aunt Lucille, may I speak with you privately?
185. Lucille, Countess of Nix: Of course, dear.
186. You withdraw to a small side room.
188. Lucille, Countess of Nix: What?!?
189. Elodie, Crown Princess: I am, too. That’s how I know. Can’t you sense that?
190. Lucille, Countess of Nix: No. No, I can’t.
191. She looks down, shaking her head.
192. Lucille, Countess of Nix: I... I do have a Lumen crystal. I found it, years ago. It called to me. But I don’t know anything about magic.
193. Lucille, Countess of Nix: I can make the crystal appear and disappear, and I can block out sound with it. That’s all.
194. Elodie, Crown Princess: (I don’t think that’s true at all.)
195. Elodie, Crown Princess: (There are traces of old magic around her, too faint to make sense of, but strong.)
197. Lucille, Countess of Nix: Magic is dangerous. Everyone knows that. People might think I was an evil witch. They wouldn’t trust me.
198. Lucille, Countess of Nix: Look at how much trouble the Duchess of Ursul has.
199. Please, Your Highness, don’t tell anyone.
200. Elodie, Crown Princess: What I think I should tell everyone is that you are my new Lumen Minister.
201. Lucille, Countess of Nix: What?
202. Elodie, Crown Princess: Lumen’s have power. If you train with me, you’ll become even stronger.
203. Elodie, Crown Princess: I want that strength here to defend the capital.
204. Elodie, Crown Princess: You’ll have status and recognition and no one will dare to call you a witch. What do you say?
205. Lucille, Countess of Nix: I accept.
206. Elodie, Crown Princess: Good!
207. It’s nice to be able to enjoy time with friends and family, isn’t it?”
208. loc(Ballroom,E3)
209. char(Princess,"Elodie","Elodie.png",speed : 2,trait: status = "crown princess",trait: eyes = "blue eyes")
210. char(King,"Joslyn","Joslyn.png",speed : 2,trait: occupation = "King" trait: personality = "kind")
211. char(CountessOfNix,"Lucille","Lucille.png",speed : 2,trait: general = "secretly evil")
212. char(CountessOfNix,"Lucille","Lucille.png",speed : 2,trait: general = "secretly evil")
214. action(Princess," talks to ", CountessofNix)
215. action(Princess,"dances with", King)
216. 
217. startdate 26-06-2020 13:20:00
218. enddate 10-07-2020 13:20:00
219. 
220. 
221. plotPoint: "Week 20"
222. { 
223. plotPointImage("background.jpg")
224. plotPointName(name "Week 20")
225. 
226. storytext("Joslyn, King Dowager, Duke of Catoris: You are requested to stand in judgement. A man has been convicted of the murder-by-strangling of his wife.
227. Joslyn, King Dowager, Duke of Catoris: He does not deny the act, but requests a pardon that he might be set free.
228. Elodie, Crown Princess: If you admit you killed your wife, why do you think I'm going to set you free?
229. Condemned Man: Save me, Your Majesty! Wasn't my fault. Demons made me do it.
230. Condemned Man: Everyone knows the power of magical beasts! They used me, they twisted my fingers into chains...
231. Condemned Man: My wife found me screaming, she tried to shake me, and the chains wrapped around her...
232. Condemned Man: I need the priestesses to bless me and make me clean again!
233. Elodie, Crown Princess: (Demons? This man is a Lumen! Where did he get a crystal?)
234. Elodie, Crown Princess: It was a demon's magic that killed your wife? Not your own magic?
235. Condemned Man: ... So you know what I am.
236. He throws out his hands, and his fingertips elongate into floating stings of gold.
237. Condemned Man: DIE!
238. Golden chains slash at you like whips.
239. With so much warning, of course, it is easy for you to resist. His attack bounces harmlessly off your shields, the chains flying back in the direction from which they came... where they find an easier target.
240. Before your eyes, the renegade Lumen is choked to death by his own powers.
241. After a moment, an orange crystal materialises next to his body.
242. Elodie, Crown Princess: I guess I'll have to take that for safe keeping.
243. You creep through the castle dungeons, catching glimpses here and there of the fate that awaits troublemakers in this kingdom. Even nobles can end up here, sometimes. It's a good reminder to stay on the straight and narrow.")
244. 
245. loc(Garden,C1)
246. 
247. char(Princess,"Elodie","Elodie.png",speed : 2,trait: status = "crown princess",trait: eyes = "blue eyes")
248. char(King,"Joslyn","Joslyn.png",speed : 2,trait: occupation = "King",trait: personality = "kind")
249. char(CondemnedMan,"Condemned man",speed : 2,trait: hostilityLevel = "dangerous")
250. 
251. action(King, " talks to ", Princess)
252. action(CondemnedMan, " talks to ", Princess)
254. startdate 10-07-2020 13:30:00
255. enddate 17-07-2020 13:20:00
256. }
257. )
258. plotPoint "Week 21"
259. {
260. plotPointImage("background.png")
261. plotPointName(name "Week 21")
262. 
263. storytext("Elodie, Crown Princess: If you attack an enemy Lumen, and you win, and they leave behind a crystal...
264. Elodie, Crown Princess: Is it possible to combine that crystal with mine to make it stronger?
265. Julianna, Duchess of Ursul: Each Lumen can hold only one crystal.
266. Julianna, Duchess of Ursul: The only way to combine their magic is if a living wielder chooses to join its power to another.
268. Elodie, Crown Princess: So I should destroy other crystals, to stop anyone from using them against me.
269. Julianna, Duchess of Ursul: That is the traditional policy - but it is a lie. Lumen crystals cannot be destroyed.
270. Julianna, Duchess of Ursul: Rulers claim otherwise to discourage thieves from seeking dangerous treasures.
271. Elodie, Crown Princess: What am I supposed to do with a crystal, then?
272. Julianna, Duchess of Ursul: Guard it, or hide it, or give it into other hands to wield.
273. Julianna, Duchess of Ursul: But if the last holder of the crystal was an enemy, the resonance of the crystal will work against you.
274. Julianna, Duchess of Ursul: It will be attuned to a personality as foolish or violent or scheming as the Lumen you defeated.
275. Elodie, Crown Princess: So anyone taking the crystal would probably become a new enemy. That's useless!
276. Julianna, Duchess of Ursul: A weak-willed candidate might be acceptable to the crystal and yet still controllable.
277. Julianna, Duchess of Ursul: You could then steer that Lumen away from dangerous paths, making an ally and changing the crystal's resonance.
278. Julianna, Duchess of Ursul: It is, however, a dangerous endeavor and one that requires your complete attention.
279. Elodie, Crown Princess: (Not really a project I can work on while I'm so busy studying for my coronation.)
280. Elodie, Crown Princess: (Well, it's only one crystal. That's not too much trouble.)"
281. }
282. loc(Castle,E2)
283. 
284. char(Princess,"Elodie","Elodie.png",speed : 2,trait: status = "crown princess",trait: eyes = "blue eyes")
285. char(DuchessofUrsul,"Julianna","Julianna.png",speed : 2,trait: title = "Duchess",trait: race = "lumen")
286. 
287. action(Princess," talks to ", DuchessofUrsul)
288. 
289. startdate 17-07-2020 13:20:00
290. enddate 24-07-2020 13:20:00
291. }
292. }
plotPoint "Week 40"
{
    plotPointImage("background.jpg")
    plotPointName(name "Week 40")
    storytext("At this time last year, you were celebrating your fourteenth birthday. You were in the school gardens, surrounded by your friends. A teacher brought you tea and cakes, while a wealthy merchant's son wove a crown of flowers for your head.

    It didn't matter so much that you were a princess then. Your title was something for the future. Many of your peers would be Duchesses or Earls or the like, someday, but not then. You were children.

    Your parents could not attend on the actual day, but they did send wonderful gifts, some for you, and some for you to share. And a week later, they came for a visit, and your mother took you with her through the countryside in a splendid carriage.

    It was the last time you would ever see her.

    You wonder if, wherever she is, she can see you now.

    You are fifteen years old, a legal adult. You have worked and studied and suffered and prepared, and now the time has come.

    You kneel before the priestess, barely hearing her words as she recites the blessings.

    She calls upon the gods to deliver peace, wisdom, and prosperity to you, and through you, to all of Nova. And then she calls upon you, for your oath of rulership.

    Priestess of the First Circle: Will you guide and govern and protect your people to the best of your ability, according to law and custom?

    Elodie, Crown Princess: I will.

    Priestess of the First Circle: Will you to the best of your power uphold the ideals of love, honor, justice, and mercy?

    Elodie, Crown Princess: I will.

    Priestess of the First Circle: Lords and Ladies assembled: I present to you your undoubted Queen, who has sworn you her loyalty.

    Priestess of the First Circle: You who have come to give homage, will you do the same?

    One at a time, the head of each duchy approaches your throne and kneels to swear his or her service to you and your heirs.

    Priestess of the First Circle: People of Nova, I give you Elodie, daughter of Fidelia, your true sovereign. What say you all?

    People: LONG LIVE THE QUEEN! LONG LIVE THE QUEEN! LONG LIVE THE QUEEN!

    Elodie, Queen of Nova: (Thank you! I will!)

    Once his daughter was secure on her throne, Joslyn returned to his birthplace to focus on his duties as Duke of Caloris.

    As Lumen Minister, Lucille requisitioned the orange crystal that Elodie had taken from the dead criminal and offered it to her daughter. However, Charlotte was not able to activate the crystal's powers. The young Queen pointed out that a crazed murderer would obviously have a very different resonance than her favorite cousin. Charlotte would have to wait to inherit her mother's magic.

    Elodie's interest in trade and economics led her to treat country management as something of a game, and a game she was very good at. Why waste lives conquering when you can dominate by natural superiority? Her manipulations increased Nova's ties with its neighbors and raised the standard of living for all of its citizens.

    And thus Queen Elodie's legacy stretched into the future...")

loc(Castle,E2)
}

char(Princess,"Elodie","Elodie.png",speed : 2,trait: status = "crown princess",trait: eyes = "blue eyes")
char(King,"Joslyn","Joslyn.png",speed : 2,trait: occupation = "King",trait: personality = "kind")
char(CountessofNix,"Lucille","Lucille.png",speed : 2,trait: general = "secretly evil")

startdate 09-10-2020 13:20:00
enddate 16-10-2020 13:20:00
}

the end
The following InputErrors were found:
There were no input errors found, good job!

The following plot holes were found:

WARNING: this character has the characteristic race which is inconsistent in the rest of the story. This error was found in line: 110

WARNING: The character Princess is found in two plot points at the same time.

WARNING: character Princess has to travel from E3 to C1 within 00:10:00 time, the distance between E3 and C1 is 2,82842712474619km. Character Princess is able to travel with a speed of 2 km/h this means that character Princess can travel 0,3333333333333333 km within that timeframe which is not be fast enough to reach C1 from E3. This error was found in line: 210

WARNING: character King has to travel from E3 to C1 within 00:10:00 time, the distance between E3 and C1 is 2,82842712474619km. Character King is able to travel with a speed of 2 km/h this means that character King can travel 0,3333333333333333 km within that timeframe which is not be fast enough to reach C1 from E3. This error was found in line: 211

The generated graph and map of the .scribb file represented in appendix A.

Figure 31: The graph generated by the system
Figure 31: The map generated by the system
Appendix C Second test phase results

Figure 32: the second phase as sorted by Walter Piccolo.
Figure 33: The second phase as sorted by Gabriele Ferri