Graduate Theses, Dissertations, and Problem Reports

2018

Generating Narrative in an Interactive Fiction Game

John L. Jacko

Follow this and additional works at: https://researchrepository.wvu.edu/etd

Generating Narrative in an Interactive Fiction Game

John L. Jacko

Thesis submitted
to the Benjamin M. Statler College of Engineering and Mineral Resources
at West Virginia University

in partial fulfillment of the requirements for the degree of

Master of Science in
Computer Science

Frances Van Scoy, Ph.D., Chair
Saiph Savage, Ph.D.
Jeffrey Moser, MFA

Lane Department of Computer Science and Electrical Engineering

Morgantown, West Virginia
2018

Abstract

Generating Narrative in an Interactive Fiction Game
John L. Jacko

This thesis explores a niche field of Computer Science called Interactive Fiction, a field that utilizes the conventions of a regular story to offer multiple variations on how the story plays out. Our goal is to explore the possibility of developing a game that can generate a story file during game play that not only reads like a short story but reflects the events that transpire during a given game play. During development, we have determined that keeping track of various "states", we can simulate a narrative based on actions that transpire in the game.

We developed the game using a language called Inform 7. Inform 7 is a language developed for Interactive Fiction. It contains classes with functionality similar to real-life objects from a narrative stand-point and provides a system of rules that can be edited to simulate real-life actions and events. The language also bases its syntax on English and is thus easy to read and understand.

Acknowledgements

I would like to thank Dr. Van Scoy for serving as my committee chair and for introducing me to the field of interactive fiction, a field I hope to explore later in life, and for agreeing to meet with me almost every week during the semesters in which I carried out this project. I would also like to thank Dr. Savage and Mr. Moser for participating as members of the committee and their willingness to meet with me as the day of the defense grew closer.

I would also like to thank Dr. Adjeroh and Dr. Woerner for signing the various forms needed to get this project approved as well as the Lane Department of Computer Science and Engineering for accepting me into their program and enabling me to carry out this research.

Table of Contents

Chapter 1: Introduction

Generating Narrative from Gameplay

The primary purpose of this project is to determine whether it is possible to develop a literary narrative that could be written using an interactive fiction game. For instance, one round of gameplay could produce one original short story (or novella) and a second round of gameplay could produce a similar, but different original story. Such a project has the potential to revolutionize how game development is handled and how the stories within games are developed.

This project demonstrates a proof-of-concept where the player is able to travel inside a game board where conversations and events occur based off of the player's current position in the game and the set of circumstances surrounding it. Such a game could lead to a new genre of games where players could read the novel they generated after gameplay and replay the game to observe how the story changes.

Interactive Fiction

Interactive Fiction is a genre of gameplay on a computer, chiefly using a text-based interface, where the game describes rooms or situations to a player (Alexander, 2014). However, they likely had their origin outside of the computing world in the form of *Create Your Own Adventure* books (CYOA) that appeared in the 1940's and were popularized the following decades (Siebert, 2014). The origin of the term *Interactive Fiction* was likely due to the use of the word *interactive* to imply some technological advancement (Montfort, 2003) although the phrase has been used outside the scope of Interactive Fiction such as distinguishing an interactive process from a batch process.

As technology became more popular, there were increased efforts to implement Interactive Fiction to electronic platforms. For example, the creator of the CYOA series, Edward Packard, has been porting his works to iOS – using HTML and iOS's Notes application to do so (Stuart, 2011). However, Interactive Fiction itself has never truly penetrated the mainstream consciousness and has even had to compete with an alternative form of fiction called Hypertext Fiction (Montfort, 2003), a type of electronic fiction that uses hyperlinks in HTML to generate complex narratives (Dougas, 2001).

However, Interactive Fiction as used in this project can be traced back to the creation of the *Z-Machine* – a virtual machine not unlike the Java Virtual Machine that was written to run *Story Files* and has its own *assembly language* (Nelson, 2001). Just as C was created to simplify software development on regular hardware (particularly in developing the Unix Operating System), languages targeting the Z-Machine were developed to simplify the process of writing Interactive Fiction. Just as traditional assembly languages were difficult to use when programming on hardware, the assembly language for the Z-Machine was rather tedious to write an entire program in as well (Nelson, 2001).

Arguably the first high-level language targeting the Z-Machine is known as the Zork Implementation Language (ZIL) and was used by a company called Infocom from 1979 (when the Z-Machine was Introduced) to 1989 (Nelson, 2001). Other languages include Inform, TADS, and Hugo, all of which use C's procedural paradigm to simply IF programming (Granade, 2010).

Inform and Interactive Fiction

There are two major versions of Inform: Inform 6 and Inform 7. Inform 6 was released in 1996 and was subject to incremental changes for a decade until Inform 7 was released (Granade, 2010). Inform 6 was described as a C-Like Language by Granade and while the example code found in Graham Nelson's *The Inform Designer's Manual* isn't exactly C, Inform 6 does appear to resemble C-Like Languages.

Inform 7 deviates away from traditional languages by basing itself on English as well as using logical programming over Object-Oriented Programming (Granade, 2010). However, Inform 7 compiles to Inform 6 prior to being compiled into a story file, which contains code related to a Z-Machine's *assembly language* (Nelson, 2001), which should not be confused with traditional assembly languages such as x86 assembly or ARM Assembly. Inform 7 uses a series of rules, many of which behave like functions when certain events occur.

Just as C/C++ and Rust allow programmers to embed Assembly code into their routines, Inform 7 (which compiles to Inform 6) allows interactive fiction writers to

embed Inform 6 into Inform 7 code using "(-" and "-)" around the Inform 6 code to partition between Inform 7 and Inform 6 (Granade, 2010).

Chapter 2: Previous Works of Interactive Fiction

There have been many video and computer games developed since computers were invented. The developers of a few of those games have looked into various aspects of the story their game may tell and innovate ways in which stories could become flexible and descriptive. Many approaches have been explored, from in-depth transcripts to multiple social cues.

The introduction of stories within games and social interactions on computers have made great strides in the ability to generate stories. However, many game generated stories are either tightly controlled by the developers where players only influence the details but not the actual plot or players control the plot, but the details are scarce (McCoy, et al., 2013). This chapter discusses some of the games that have explored the art of flexible, interactive storytelling as well as the innovations of turning control of the plot over to the player.

Façade

Façade is a computer game that features drawings to depict a couple whose relationship is unstable. It consists of an interactive GUI where you, the player, can see the room and the characters you interact with. The Game also manages their interactions with you as you type in commands that could result in them remaining together or splitting up. They invite you, a friend, over for your advice on an issue.

Façade is seen as an advancement of procedural authorship – the ability to create procedures in order to showcase a story or narrative (Mateas & Stern, 2010). It was praised for having characters that are "people rather than dolls". For one thing, the things you say have a major impact on the direction *Façade* goes in and "no two plays are identical" (Rauch, 2006). It is likely because the creators of *Façade* wanted to give players *global agency* over the entire story rather than *local agency* that give players control over what happens in a given moment (Mateas & Stern, 2010).

Façade introduced the concept of unique dialog and how the players words have a genuine impact on the direction of the narrative. In fact, many times the player character is compelled and/or pressured to participate in the conversation between the couple. However, Façade is not without its flaws. Rauch discovered that the characters did not fully grasp the concept of pregnancy when he attempted to claim that he (playing as a woman) was pregnant. The characters also do not know what milk is.[1]

Alabaster

Alabaster is a game written in Inform 7 that offers a unique take on the story of Snow White. The player is the huntsman, who is caught in the struggle between the Queen and Snow White, with each having her own ulterior motives. The game is said to have up to eighteen possible endings (Carter, et al., 2009).

It was developed by a team of interactive fiction writers and documented in a blog by one of the developers – Emily Short – who also worked on collaborating with people who made contributions to the project made through crowdsourcing. To that end, Emily reached out to potential developers and received information regarding bugs, patches, as well as general feedback on user experience (Short, 2008).

Prom Week

The game known as *Prom Week* was an attempt at creating a game that not only allowed rich interactions but allowed them to occur at any point in the game. Players in the game *Prom Week* are given a set of tasks or goals to complete within a week, which they do so by initiating *exchanges* that could modify the *social state* (McCoy, et al., 2013). Exchanges, of course, are conversations between characters and social states are as series of variables used to deduce relationships.

Game development traditionally involves a balance between gameplay (or player interaction) and narrative/storytelling (McCoy, et al., 2013). Many games attempt to do one of two things. They tell a rich story, but the overall narrative is still tightly controlled by the developers (McCoy, et al., 2014). They can also allow interactions to occur at

---

[1] I discovered the lack of familiarity with milk. When my character was offered a drink, I suggested milk and the game characters genuinely did not know what I was talking about.

any point in the game, but the narrative is described as limited and possessing few-consequences (McCoy, et al., 2013).

To bypass these narrative barriers, *Prom Week* utilizes a *social engine* called *Comme il Faut* (CiF) designed to manage the presentation of social interaction amongst characters within a game (McCoy, et al., 2014). CiF is described as an AI system that takes into account context surrounding a conversation (McCoy, et al., 2013). In fact, there are over 5000 rules related to social interaction and a more dynamic relationship than *friends/enemies* dynamic that impact the turns a conversation may take (Samuel, et al., 2016). The exchanges and social states used by *Prom Week* can be managed by CiF, which contains procedures that manage how they impact the social state and reports actions a given character makes – which *Prom Week* interprets for the user (McCoy, et al., 2014). Because goals within the game could be attempted at anytime (McCoy, et al., 2013), *Prom Week* was able to bridge the gap between deep narrative and player flexibility.

Chapter 3: Narrative and Contribution to knowledge

Our proof-of-concept for generating *Game Narrative* is an interactive fiction game using Inform 7. The game is for middle school aged girls designed to provide them with a story that can be read by a friend with no knowledge of the game. Other potential goals with the game is to help the player discover that learning is fun and provide the player with a review of what was learned.

Inform 7 provides a means through which narrative could be generated during gameplay, by means of declaring a file and writing to it during various events. However, the challenge comes in the form of making the text flow as it would in an ordinary novel. Simply declaring that every turn, the short story will describe which characters the protagonist encountered could look messy. The project had to be modified to avoid narrative such as when the player enters a series of rooms, the game does not generate text such as "She saw Walrus. She Saw Walrus. She Saw Walrus." Walrus is a character in the game and this narrative was generated when the player moved to the same rooms Walrus was moving to.

One way of knowing which text to append to the story is by checking and updating the *state* of various objects and environment relevant to the current events. Such states come in various forms throughout the game but are instrumental in determining the story line and its expression. There are two ways to describe the type of state: the data type used and the story type. In regard to data types, such states are typically either numbers or indirect booleans.[2]

In terms of a state's contribution to the story itself, they typically come in the form of *awareness*, *objects*, *progress*, and *attributes*. States that contribute to awareness are typically used to answer questions about what either the player knows or what a non-player character knows. The most prominent example of the awareness state is whether or not the player knows how to win the game using a boolean attached to the player character. This attribute is further discussed in the *Win Condition* section of Chapter 4.

Another form of the narrative state of the game is what a player is carrying. Inform 7 provides an inbuilt mechanism of managing whether an object is being carried or not, whether such an object is a *container* that can carry other objects, as well as whether an object is something that can be worn. This mechanism is used by the game to require the player to hold a water jug before watering plants and how Tweedle-dee and Tweedledum, during their fight, will acquire a helmet and/or body armor. However, when it comes to generic objects of an unspecified quantity, it is easier to declare a number and have that number represent the quantity of goods a character is either possessing or storing in an *inventory*. The clearest example of this technique in the game are the global variables used to represent the Inventory of Walrus and Carpenter's restaurant (which holds salmon, corn, carrots, and peas) as well has how many of these goods either the Walrus is carrying or the player is carrying back to the restaurant.

In this game, *progress* is determined by the four numbers that influence how many coins of a certain type out of four that the player possesses. The player can earn

---

[2] The phrase *indirect boolean* is used as they are never referred to in Inform 7 as bool or boolean and don't use the traditional true/false values. Instead, Inform will have an object described as either one adjective or a different adjective, such as a room being built or not built, for example.

certain coins by performing certain tasks and each coin type brings the player closer to being able to enter a specific castle. One of these numbers can also influence the *awareness* of the player. If the player knows about the win conditions but possesses no yellow coins, then the player believes that the yellow coins don't exist as that is what the characters who tell her about the win condition believes. On the other hand, characters who *do* know about the yellow coins and elect to inform her about the win condition will provide her with one.

The final category, *attitude*, can work in two ways: the player's attitude towards a given non-playable character and a non-playable character's attitude towards the player (though non-playable character's attitudes towards each other could be a third way, that way is never handled by the game). The clearest example of a non-player's attitude towards the player is a plants possessive willingness to allow the player to pick them up. If a plant is willing, then the player is allowed to pick up the plant. Otherwise, attempts to pick up the plant will earn the player a strike in the Garden. This leads to another attitude attribute: the willow tree's attitude towards the player. Each strike increase boost's the Flower Garden's Willow Tree's animosity towards the player before the Tree bans the player from the Flower Garden for a variable amount of time. The best example of the player having an attitude towards another character is whether the player has purchased vegetables at the Vegetable Garden before. If she has not, then she'll find the vendor somewhat intimidating. But after purchasing vegetables for the first time, the errand becomes routine and the narrative never mentions her trepidation towards the vendor again – as she has gotten somewhat familiar with him.

The strongest mechanism for documenting history is to append it to the narrative as it happens. For instance, the moment enters a long conversation with Tweedle-dee/Tweedledum, the Lion, Humpty-Dumpty, the Red King, or the Latin teacher at the Great Lake, it makes sense to jot down what is being said. This is especially true if the conversations themselves are static and don't branch out in multiple possible directions. In other phases of the narrative where what is printed is heavily dependent on certain states, great care has to be taken in not only accurately describing the situation but describing it in a way that sounds natural to human readers.

Chapter 4: Project

Overview

The project was written in Inform 7, the interactive fiction language that more closely resembles natural languages than other languages. The first aspect of the game developed was the environment in which this game would take place. The specification provided by Dr. Van Scoy described a gameboard-like setup with eight-by-eight tiles, each of which had the potential to add their own content to the story.

Previous approaches to implementing the game *Through the Looking Glass* focused on developing one or two rooms and implementing in great detail the storyline specific to that location. Specifically, the rooms related to the Flower Garden and Vegetable Garden have been well-developed (VanScoy, 2016). The approach used in this project deviates away from previous approaches by setting up the entire board and filling in the details of various rooms one room at a time. It also rewards the player character with in-game coins for various actions taken, rather than badges for information learned.

```
The GameBoard is a region.
A room can be built or not built.
A room has a number called visit.

There is a room called Castle1. The visit of Castle1 is 0. Castle1 is in the GameBoard.

There is a room called Blank1. The visit of Blank1 is 0. Blank1 is in the GameBoard. Blank1 is east of
Castle1.
```

Code Listing 1: Extending Room and creating rooms.
Rooms can be within regions. They can also be extended. In this game, the room class was extended to be built or not built (a boolean value) as well as a number to indicate whether the player visited it or not. Two rooms are created in the code above (though many more are created throughout the actual code). The first room (Castle1) is placed in the game board region but is used as a reference for other rooms (like Blank1) to position themselves in the game board.

To develop the board, Inform's *room* class was used. Rooms in Inform simulate places where characters can be, and they can be connected freely or locked off. They possess features such as locks and doors such that you can depict them as actual

rooms or you can treat them as if they are simply areas in a field. The room was extended to support a *visited* attribute where rooms not visited by the player are considered unvisited.

Once the board area was set-up, characters were developed. Each character was essentially an *object* of the type character. The reason the character class was written was that while Inform provided the base class of person (further divided into man, woman, or animal), the person class did not possess all of the attributes (whether numbers, booleans, texts, or another type) needed, and while such attributes could have been appended onto person, they were added to a *character* class as the character class has its own sub-class called plants, who serve as characters capable of speech. While they had the description string, for example, they needed to have multiple strings to respond to various interactions with the player or possibly even with each other.

Walrus and Carpenter's Restaurant

A great deal of the story revolved around the restaurant set up by the Walrus and Carpenter. In the story, the game begins with the Walrus and Carpenter arriving on the

Table 1: Variables related to the restaurant run by Walrus and Carpenter and how they work. The *count* variables keep track of the supplies in the Restaurant's inventory. The *carry* variables indicate the number of supplies being brought back to the restaurant, either by the player or by the Walrus. The *table* variables are lists that contain characters who become customers at the restaurant.

CornCount	Servings of Corn Held by the Restaurant
PeaCount	Servings of Peas Held by the Restaurant
CarrotCount	Servings of Carrots Held by the Restaurant
SalmonCount	Servings of Salmon Held by the Restaurant
CornCarry	Servings of Corn being carried back to the Restaurant
PeaCarry	Servings of Peas being carried back to the Restaurant
CarrotCarry	Servings of Carrot being carried back to the Restaurant
SalmonCarry	Servings of Salmon being carried back to the Restaurant
RestaurantNeed	Used to keep track of which menu item is running low, with individual bits being used.
CarpenterState	Keeps track of the Carpenters activity if he must Serve Customers
Table1	List of Customers Seated at table 1
Table2	List of Customers seated at table 2
Table3	List of Customers seated at table 3

scene, which then causes the previously unbuilt restaurant to be built. While the two individuals start out at different locations on the game board, they navigate towards the site. Once the Walrus and the Carpenter arrive, the restaurant is marked as built and the player can now enter it.

After construction, the Carpenter remains at the restaurant while the Walrus begins a journey to retrieve supplies for the restaurant's inventory. It is at this point that the player can enter the restaurant and apply for a job. If the player does so, then it becomes the players responsibility to service orders and retrieve supplies for the restaurant (although the carpenter can service orders in the players absence).

The restaurant has a menu of four items: peas, carrots, corn, and salmon. Peas, carrots, and corn were mentioned in the game's specification as possible menu choices (VanScoy, 2016). Customers will venture to the restaurant when their hunger statistic reaches three. Their hunger statistic starts out at ten (not hungry) and will slowly decrement at random frequencies.

Once characters arrive at the restaurant, they are randomly assigned a table and have a number from 0 to 255 that represents their order. Indirect bitwise manipulation is used to partition this number into four numbers with a range of 0-3 with each smaller number representing the number of choices on the menu.

If the player is employed and present, then the player has three commands available for taking orders, one command for each of the three tables. The player can handle only one order at a time and once an order is taken, it can be reviewed. The player then has the dispatch commands (three for each table) and should run the dispatch command that aligns with the table that the order was taken from. For more information on commands, see Appendix C.

The dispatch command, under optimal conditions, will satisfy the customer, who will then be on his/her way. However, the inventory could run dry (and will if the player services characters long enough). Once this happens, it is incumbent upon the player to go out of the restaurant and refill the inventory. While most of the food can be found in the Vegetable Garden, salmon can be retrieved from the area known as Great Lake.

Flower and Vegetable Garden

The Flower Garden and Vegetable Garden are two rooms located on the east edge of the board. The Flower Garden is located north of the Vegetable Garden. Multiple Flowers are defined and positioned as *scenery*, or things that won't be picked up. There are multiple character-derived flowers positioned in the flower garden as well. The player can interact with the character-flowers but not the scenery flowers.

```
Watering is an action applying to one thing.
Understand "Water [something]" as Watering.

Before watering something:
        if the noun is not a plant:
                say "Only plants can be watered!";
                stop the action;
        if the player is not holding the Water Jug:
                say "You need to pick up the water jug in order to Water a plant";
                stop the action;

After watering a plant:
        Now the Water Level of the noun is 10;
        If the willNeed of the noun is 1:
                now the willNeed of the noun is 0;
                Now the noun is willing;
        else if the willNeed of the noun is 3:
                Now willNeed of the noun is 2;
```

Code Listing 2: The *watering* action that allows the player to water plants. There are four main components to this action (as well as most other actions declared in the game). There is the declaration of the action, the command that activates it, the *before* rule that check for conditions that make the action infeasible, and the a*fter* rule that implements the action. Successful runs of the *watering* action increase the plant's water level to 10 (they have been watered) and might make the plant willing to be picked up.

The character flowers have a thirst meter and once their thirst is satisfied, some will allow you to pick them up and carry them in a soil pot (become *willing*). One storyline not implemented in the game is to allow the player to converse with the plant[s] she takes with her and gain the flower's insight on the game world. However, many plants will not allow the player to pick them up and the player earn up to five strikes for

each attempt to pick up an unwilling plant. If the player accumulates five strikes, then the player is expelled from the Flower Garden for a varied amount of time. Every turn During the suspension, a random chance (of 1 in 10) will reduce the strike count. When the strike count reaches zero, then the player will be allowed to enter the Flower Garden again.

The player is also required to acquire potting soil before picking up a plant (failure to do so can also increase the strike count). At the start of the game, Potting Soil can be found in the shed which is located in the Vegetable Garden. The Vegetable Garden is also the location where vegetables for the restaurant can be purchased. If the player is employed, then purchasing vegetables is as simple as typing the command *Purchase Vegetables* (assuming the player is in the Vegetable Garden).

There are five sub-rooms within the Vegetable Garden: Hydroponics, Shed, Green House, Demo Area, and Product Stand. There are commands set up to enter each of these rooms as Inform's built in command to "go in" only handles 1 room within a room, not five. Consequently, these five commands were set up to handle each of the five rooms. Within each room, there is a command to get back to the Vegetable Garden.

Humpty Dumpty

Humpty-Dumpty is situated in the room known as *Humpty-Dumpty's Wall.* If the player has not chatted with Humpty-Dumpty before, there is a four-in-five chance that the player will have a conversation with the character. Essentially, should a conversation begin between the two, the player will have the opportunity to complement a feature on the character, either calling it a necktie or a belt. If Alice calls it a belt, Humpty-Dumpty will insist it is a necktie (VanScoy, 2016).

However, it is also possible for Humpty-Dumpty to fall from his wall and while the scene is not all that graphic, it does provide the player with an opportunity to aid a character. This can occur regardless of whether or not Humpty-Dumpty and the player have previously talked. This scenario occurs in the 1-in-5 chance that a conversation *doesn't* occur. If Humpty-Dumpty falls, a numeric variable is set to a certain value and this number is used to represent various stages in Humpty-Dumpty's interaction with the character.

If Humpty-Falls, the player will be able to take him to the North-West Castle where he is presumably healed from his fall. While the player may not be able to enter the castle yet, the player will earn blue coins that could be used to gain access to the castle in the future.

It can also be noted that, because of the possibility of Humpty-Dumpty falling from his wall, that Humpty-Dumpty is a somewhat tragic figure. Given that Humpty-Dumpty is a non-playable character, this arrangement makes sense as the fall can surprise players of the game[3] as some games such as a game called *Neverwinter Nights* (Sheldon, 2004).

Minor Story Lines

Most of the story line discussed so far allowed the player to commit to certain actions in order to drive the narrative. However, there are two areas where the narrative is relatively simple and straight-forward: The room called "Great Lake" and the room called "TT House" (Tweedle-dee and Tweedledum's House).

Great Lake

There are three major scenarios that can occur at "Great Lake." There is a state, labeled "LakeStat" in the code, that is used to keep track of which scenario has occurred. Consequently, each scenario will only occur once during game play.

The first two scenarios are completely straight forward and only serve up a straight sub-story line. The third scenario can branch off into whether or not the player accepts the offer of the character (in this case, a Latin Teacher) to learn how to spin yarn. If the player rejects the offer, then it can be given later down the line, although it allows the player to learn how to win the game.

The House of Tweedle-dee and Tweedledum

There are four scenarios of narrative that can occur at the house of Tweedle-dee and Tweedledum. The first three scenarios are extremely similar to each other and occur if at least one of the twins are at the home. The fourth scenario occurs when the twins are absent and if it occurs, then the Red King is brought into the narrative at this

---

[3] Assuming the player is unfamiliar with the nursery rhyme or is unaware that Humpty-Dumpty is in the game.

place and explains to the player that she can become a queen by entering one of the four castles and collecting enough coins for a given castle.

Win Condition

The Player wins the game by entering one of the four castles located in the four corners of the game board. However, entry to the castles require the accumulation of seven coins and each castle accepts only a certain type of coin. The table below showcases the types of coins available, how to earn them, and the castle they correspond to.

Coin Type	Castle	Availability
Green	North-West	Taking a Fallen Humpty-Dumpty to the North-West Castle for treatment; Summoning the Crow to break up Tweedle-dee and Tweedledum's fight
Blue	North-East	Picking up Supplies for Walrus and Carpenter's Restaurant
Red	South-West	Dispatching Orders at the Restaurant
Yellow	South-East	Given if the Lion informs the player, and if the player waters plants in the Flower Garden

Table 2: There are four-coin types and castles that they correspond to. The table also lists the ways in which each coin type can be earned.

Also of importance is keeping track of whether the player knows about the win condition. Presumably, the player begins the game not knowing how to win the game (become a queen) and thus will likely explore the game board looking for interactivity. However, there are opportunities to learn how to become a queen. The first two opportunities occur at Tweedle-dee and Tweedledum's House and at the lake.

At Tweedle-dee and Tweedledum's House, the twin's absence cause a conversation to commence between the player and the Red King that leads to a quick discussion on how Alice becomes Queen. She asked the King how and he brings up the seven coins requirement per castle and which coin type goes to which castle before departing.

In the room called "Great Lake," rejecting the Latin Teacher's offer to teach yarn weaving leads to the teacher remarking how Alice would never become a queen with an attitude like that. If Alice did not previously know that she could become a queen, then she'll ask the teacher about the matter. The teacher will provide a quick explanation regarding the win conditions.

Both the Red King and the Latin Teacher describe which Coin type goes to which Castle as well as the lack of yellow coins available. However, the absence of yellow coins is something that most characters merely believe is true. The Lion knows that they exist and under the right circumstances (i.e. the Lion informs the Player of the win conditions) will give a couple to her. The characters also omit the means through which each coin can be earned as that is something the player needs to discover for himself/herself.

It is also possible that the player does not run into the Latin Teacher at the Great Lake or the Red King at the House, meaning Alice never learns how to become a queen. To better remedy this, three characters will explain to Alice the win conditions when they encounter her, assuming she doesn't already know. They can also introduce new information that could help her on her journey during this time.

In the event the player wins the game and Alice never discovers the win conditions, then naturally, she is going to be shocked when she is crowned Queen. Because of this, her status as *knowing* or *unknowing* will be used by the game upon entry into the castle and which version of the ending to use: the regular ending or the surprise ending. If Alice did not anticipate becoming Queen, then a variation of the original narrative will be used, emphasizing her surprise at the coronation that takes place.

Chapter 5: Narrative Techniques

In this section, I will present various techniques used to implement narrative in a short story generated from gameplay. As stated before, customizing narrative appropriately to different circumstances depended on keeping track of various states in the game. However, that is a generalization and the specific mechanisms through which this is done is just as important.

```
[Prepare File support]
The File of the Looking Glass is called "story2". [Declare File]
When play begins:
        Write "<Opening Text>" to the file of the looking glass;

After <SomeEvent>:
        Append "<Event Details>" to the File of the looking glass;
```

Code Listing 3: Inform 7's mechanism for setting up a text file to write to. *Opening Text* is a place holder to the text that's written, as seen in Figure 1.

Knowing what to write presents one challenge to story development. Knowing how to write it is another challenge.

Beginning

The player starts out in a room called "Launch Point." A file called "story2.glkdata" is declared when the game begins. This file is referred to throughout the game as "The File of the Looking Glass".

Adding to the procedure that is run when the Game Begins, the file is written to for the first time. The *write* command is used when the game begins. However, subsequent writes to the file are done through the *append* command in order to not overwrite the current contents of the file.

While the opening text may be consistent regardless of gameplay, the player is placed in control shortly thereafter and the narrative can diverge from there.

"Her name was Alice! That much she knew, even as her dreams began to fade and her consciousness returned to her. She did not remember what was going on before she fell asleep, but she now felt herself waking up. As her vision came into focus, she began to process the environment in which she found herself

"She appeared to be in a light meadow, with a few flowers here and there. She could see in one direction a vast sea. Two Castles were situated on the sea shore, both being considerably far away from each other. To the opposite side of Alice stood a dark building that looked long abandoned.

"It dawned on Alice that she had no idea where she was but she felt around a pocket in her dress and felt a parchment. But then she took another look and discovered a sign was present and left Alice wondering if she should read it."

Figure 1: Opening text for each file generated in the game. While the opening text may be consistent regardless of the gameplay, the narrative can easily diverge from this point.

Character Encounters

One critical aspect of developing a story is documenting the protagonist's encounters with other characters. If the given character has not been mentioned previously, then a mention of the character should come with a description of that character. Initially, this was done with one loop going through the list of characters. For each character in the list, if the character is in the player's location, say "She saw

<character>" and if the character has not been previously encountered, say
"<character> was <description of character>".

One drawback from using such a simple approach was that while the description of the character made sense, the "She Saw <character>" ended up being written persistently and repeatedly. Such text naturally reduces the quality of the narrative so steps were taken to mitigate this.

One step that was taken was that a counter was added to each character such that they are mentioned as spotted by the player after every five turns. Another step taken was to check to see if a given turn resulted from the player moving or performing

```
Every turn:
        if the player is unmoved:
                stop the action;
        let CharacterList be the list of characters;
        let playLoc be the location of the player;
        Append "After moving to the [playLoc], Alice saw";
        Repeat with chara running through CharacterList:
                if the Location of chara is playLoc:
                        Append "[chara], " to the file of the looking glass;
        Append "." to the file of the looking glass;
        Repeat with chara running through CharacterList:
                if the narrativeCounter of chara is greater than 0:
                        decrease the narrativeCounter of chara by 1;
                let charLoc be the location of chara;
                if playLoc is charLoc and the narrativeCounter of chara is 0:
                        if chara is unencountered:
                                Append "[chara] was [description of chara].";
                        now chara is encountered;
                        now the narrativeCounter of chara is 5;
        Append "[line break]";
```

Code Listing 4: Code snippet of the game that handles character sightings. This version was active as of March 16, 2018. The player can be *moved* or *unmoved* and is "moved" when the character enters a different room. This status is checked before the characters in the room are mentioned. After that, the characters not previously encountered are described.

Note: Append statements have the file to be appended omitted in this particular code snippet.

some other task. The logic behind this move was that the protagonist is way more likely to notice someone else while on the move. If a character enters the same location as a stationary player, it is possible that the player won't even notice.

However, such changes proved to be insufficient as if there was nobody present, then the story file still contained text such as "After moving to <location 1>, Alice saw . After moving to <location 2>, Alice saw ." As with repetitive "She saw <character>" statements, such text is unsuitable for a given narrative and thus additional restrictions had to be put in place. A separate character list was created alongside "CharacterList" called "LocalCharacter". Any character in the first list that's in the same location as the player was placed in the second list.

```
let CharacterList be the list of characters;
[Using "the" puts all existing characters in "CharacterList"]
Let localCharacter be a list of Characters;
[Using "a" caused "localCharacter" to start as an empty list]

let xSize be the number of entries in localCharacter;
let x be 0;
if localCharacter is not Empty:
        Append "After moving to the [playLoc], Alice saw ";
        Repeat with chara running through LocalCharacter:
                if xSize is 1 or x is xSize:
                        Append "[chara]. ";
                else if x is xSize - 1:
                        Append "[chara], and ";
                else:
                        Append "[chara], ";
        [say "[x] [xSize][line break]";]
        Increase x by 1;
```

Code Listing 5: The game's way of knowing how to format a list properly in narrative text. Two list of characters are created: one list with all characters in the game and one list to hold which ever characters are currently in the room. By keeping track of the number of characters and keeping a counter, the game can deduce how to list the characters and determine if/when to put the word *and*.

This second list was then filtered out from individuals whose inclusion would distort the narrative. Humpty-Dumpty is typically removed as he has his own narrative introduction, as is any character the player is currently *carrying*. A final change made was to determine how the list should be printed to the file in order to properly determine when to use a comma, period and the word *and*, as showcased by Code Listing 5.

Narrative During an Event

Many times, in an inform game, the console will print out text that describes the situation the player is in. This occurs in *Alabaster* and this can also occur in this game. However, the description printed to the console describing what was going on also acts as a narrative read by the player during gameplay. Consequently, one approach in generating narrative is to write to the file text similar to what appears on screen.

For example, if you encounter Tweedle-dee and Tweedledum fighting each other, then you have the option of entering the command *Interrupt the Twins*. The following table shows the text that will appear on screen next to the text that is written down on the file.

Console	Story File
The twins stop for a moment. 'Why are you fighting?' you ask 'He's trying to take my throne' one replied 'Because it IS MY throne!' the other snapped back. 'What throne?' you asked, puzzled. The two instantly snap to attention. 'We are the sons of the red King and Queen,' they said. 'I'm Tweedle-dee', said the twin with the word 'DEE' on him. 'I'm Tweedledum, and when I become King, I will get my name changed' said the twin with 'DUM' attached. 'But how is fighting going to help resolve this dispute?' 'One of us will get tired and submit,' Tweedle-dee said. 'Or you could have the crow decide,' Tweedledum replied. 'He can solve everything'. The twins resume fighting and you realize only the Crow could intervene properly. (If you encounter the Crow, type 'Summon Crow' to resolve the situation).	The twins stopped for a moment. 'Why are you fighting?' Alice asked. 'He's trying to take my throne' one replied 'Because it IS MY throne!' the other snapped back. 'What throne?' Alice asked, puzzled. The two instantly snap to attention. 'We are the sons of the red King and Queen,' they said. 'I'm Tweedle-dee', said the twin with the word 'DEE' on him. 'I'm Tweedledum, and when I become King, I will get my name changed' said the twin with 'DUM' attached. 'But how is fighting going to help resolve this dispute?' 'One of us will get tired and submit,' Tweedle-dee said. 'Or you could have the crow decide,' Tweedledum replied. 'He can solve everything'. The twins resume fighting and Alice realized only the Crow could intervene properly.
Figure 2: Comparison between the console text and the story file text generated after a successful execution of the "Interrupt the Twins" command.	

Traditionally, the player is given narrative from the second-person point of view, whereas the story book is told from the third-person point of view. Also, the console might offer hints as to new commands available, information that would not be used by someone reading a book after the fact. Other than that, the narrative generated is the same.

Knowing Text (case where player/Alice is informed of the win condition)
Alice cautiously approached the Castle Gates. She was approached by a knight in used-silver armor. 'How many Yellow Coins do you have?' the knight asked. Alice cautiously counted her coins. [Yellow Key] Yellow Coins, YES! [Yellow Key - 7] more than she needed. She also had [Blue Key] Blue Coins, [Green Key] green keys, and [red key] red keys, but they didn't matter for this castle. She showed the knight the Yellow Coins she had collected. 'Come with me!' the knight said, to which Alice obliged. Alice followed the knight to a gathering of knights gathered in a throne room, one of whom held a crown. Once Alice was a few feet away from the knight, she instinctively kneeled. The knight placed the crown on Alice's head and said 'Rise, Queen Alison.' She rose and turned to the crowd of knight, who began kneeling before her.
Surprised Text (case where player/Alice are never informed of win condition)
Alice cautiously approached the Castle Gates. She was approached by a knight in used-silver armor. 'How many Yellow Coins do you have?' the knight asked. Alice cautiously counted her coins. [Yellow Key] Yellow Coins! Would that be enough? How many would the knight demand? She also had [Blue Key] Blue Coins, [Green Key] green keys, and [red key] red keys, but they didn't seem to matter for this castle - no that the knight was letting on, anyway. She showed the knight the Yellow Coins she had collected. 'Come with me!' the knight said, to which Alice obliged, hoping nothing bad was about to happen. Alice followed the knight to a gathering of knights gathered in a throne room, one of whom held a crown. But the knight wasn't wearing the crown but rather was holding it in his gloved hand. Once Alice was a few feet away from the knight, she stood there, not knowing what was going on. 'Alice! You have successfully return all the coins this Castle requires!' The knight with the crown said. 'As the prophesy requires, you are now anointed to become a queen!' 'A queen?' Alice asked, shocked and wondering if what the knight said was what she thought he said. 'If you kneel before me now, you will become Queen Alice.' Alice decided to kneel, if only to follow the knight's order, still thinking this might be a joke. The knight placed the crown on Alice's head and said 'Rise, Queen Alison.' She rose and turned to the crowd of knight, who began kneeling before her. She still needed to grow accustomed to what was happening.

After going to Castle4:
      say "<**Congratulatory text to player**>";
      if the player is knowing:
            Append "<**knowing Text**>" to the file of the looking glass;
      else:
            Append "<**surprised Text**>" to the file of the looking glass;

Figure 3: Comparison of text between case where Alice knows what is going on verses the case where she is in the dark about the situation. Red text indicates a deviance away from the other narrative. Also featured is the code snippet that handles which text to append to the story file, with the text being provided condensed down for size as indicated by the **bold** font. This example focuses on Castle4, the South-East Castle that requires yellow coins, but with the exception of the coins mention in the narrative, the narrative for entering the other three castles is the Same.

Conditional Narrative

Sometimes, it is necessary to provide alternative narrative due to the presence of certain conditions. For instance, the player can be informed of how he/she wins the game or can go through the game without ever receiving this information. If she wins the game without knowing the win condition, she will be considerably shocked. If she has been told beforehand, then Alice will be better prepared for what happens when she wins. The preceding figure shows the ending of the narrative and details both scenarios.

Sometimes the narrative can deviate with one simple insertion of text under the right circumstances. For instance, it was stated earlier that most characters believe the yellow coins to be a myth. However, if the Crow or the Fawn (two characters that inform Alice of the win conditions) inform her of how she can become a queen, they will express skepticism that the yellow coins even exist.

However, if Alice has previously collected yellow coins, she can quickly mention that she does in fact have yellow coins. such a narrative can quickly be inserted into the text. For instance, the following text showcases the encounter with the crow, with the text added when Alice has yellow coins in red.

'Ahh, Miss Alice!' said a screechy voice. It was the crow. 'Do you know how to become Queen?' he asked.
'I wasn't aware that was possible,' Alice said.
'Ahh, but it is! If you collect seven coins of a certain type, you can enter a specific Castle and become it's queen. Collect Green Coins and you can enter the North-West Castle. Blue gets you access to the North-East, Red the South West and Yellow I'm told the South-East, though I don't believe that Yellow Coins even exist.'
'Ahh, but the Yellow Coins DO exist!' she said, holding the coins out.
'Ohh, learn something new every day,' the crow remarked, shocked at the sight.
'So I can become a queen just like that?' Alice asked, still skeptical.
'Yes!' the Crow said, starting to fly away.

Figure 4: Text generated if the player encounters the crow without knowing the win condition. The text in red is present if the player has already earned yellow coins (having watered plants) and is omitted if no yellow coins have been earned.

Preventing Repeats

One crucial aspect of narrative generation is the avoidance of repeats. Many games permit a repeat of certain events, often involving a character whose intended

contribution to the game has been used (Sheldon, 2004). While this may be acceptable in a regular game, it is less acceptable in a game that generates text during gameplay.

We have already seen one clear example of restrictions on repeat text in the form of character descriptions. When Alice encounters a character not seen before, a description of the character is added to the story file. However, if she encounters the character again, then the description is omitted – and should only be permitted if a description of the character changes since their last encounter[4].

Of course, it's not enough to simply restrict a description being given. We also must avoid allowing the same events and conversations to occur. Many examples of such restrictions include states that not only indicate the progress of a storyline but are also used to prevent the storyline from occurring a second time. Such restrictions have been added to the storyline of Humpty-Dumpty, the Teacher in the Great Lake, and Tweedle-dee and Tweedledum. More information on these states can be found in appendix D.

Transitional Narrative

Arguably the most difficult aspect of interactive narrative development is the transition between the various events that can occur and being able to provide a smooth transition from one short story to another. The reason this is the most difficult step in narrative generation is because not only does a given story line have to be considered but its variation based off of the storyline that came before it. Consequently, generating narrative that flows is more precise than previous aspects of the project.

Consider Alice's entry into one of the Castles. If Alice has entered the Castle shortly after earning enough coins to do so, then it is safe to assume she rushed to the castle and was eager to become a Queen. However, if she enters into a conversation with someone or does some task between the time she earns enough coins and the

---

[4] The Specification of the game gives a greater role to the character known as "The Frog" whose real name in Henry. In it, Henry slowly acquires more clothing as Alice gets closer to becoming a queen and the clothing type reflects the region in which Alice does become queen. However, this storyline has not been implemented in the game.

time she enters the castle, then she clearly didn't rush to the castle and the narrative should not be written as if she did.

To deduce the proper narrative transition, a numeric variable called "Narrative State" was used. In the current build of the game, there are a total of 17 possible

0	The Player just started
1	Alice was just at the W&C Restaurant
2	Alice was sent to collect supplies for the restaurant
3	Alice Finished dropping them off
4	Alice finished a chat with Humpty-Dumpty
5	Alice Finished helping Humpty-Dumpty
6	Alice Finished talking to the Teacher in the Lake
7	Alice Finished chatting with Tweedle-dee and Tweedledum
8	Alice just finished purchasing supplies
9	Alice just watered a plant
10	Alice just completed a coin collection
11	Alice completed a coin collection and she is coming from the Flower Garden
12	Alice completed a coin collection and she is coming from the Restaurant
13	Alice just earned coins from the Crow
14	Alice just earned coins from the Crow and it completed her coin collection
15	Alice just got acquainted with the flowers
16	Alice just got introduced to the Vegetable Garden

Table 3: The Narrative States implemented in the game. When jumping from scene to scene, it is important to provide some context regarding how the transition occurred. The narrative state is used by each scene to check what the previous scene was and generate the appropriate narrative. Many states offer insight regarding Alice's previous location and thus it is possible to deduce whether she traveled from the east or traveled south to get to her current location.

values, each of which indicate where the last major activity of the player was relative to the narrative. The following table describes the possible states that this value could be along with a description.

Given that there are only seventeen states and 71 rooms, it makes sense that not every room is covered by this state. However, given that most of these states are room specific, it is easy to deduce where Alice came from when introducing new narrative. For instance, given that the Flower Garden is to the east of the large lake, if the *Narrative State* is 6 (the Teacher at the lake) when Alice enters the Flower Garden,

"That makes no sense!" Alice exclaimed!
"But it makes PERFECT sense. I'm a Latin teacher, and in Latin the word "iam" or "jam" means "now" but it can only be used in the past or future tense. To say "now" in the present tense in Latin you have to say "nunc."" she said.
After saying good bye to the Woman, she walked around the lake and proceeded to go slightly further to the north. Alice found herself walking towards (or next to) a wall and to her surprise, she heard a voice from the top of the wall! "State your name!" it said.
"My name is Alice!" she replied, stopping.
"What does Alice mean?" said the voice.
 "I don't know!" Alice responded.

Figure 5: Presentation of the transitional narrative. This example showcases a chat Alice with the Teacher in the Great Lake before proceeding to travel north where she ran into Humpty Dumpty. The green text is text from the lake narrative and the blue text is from the Humpty-Dumpty narrative. The red text is the newly added transition narrative between the two.

then Alice had entered the garden by walking east. If she was coming from further north, then the narrative says that she was coming from the north-west.

Chapter 6: Survey

Much discussion has been made about how the game generates narrative. However, determining our degree of success is another aspect of this project. While it is possible to simply declare that we have generated narrative, such a declaration would be subjective and thus not suited for professional researchers.

While subjectivity cannot be entirely excluded from assessing the degree of success, a survey of potential users can bypass self-bias and offer an assessment closer to objectivity as the individuals assessing the narrative have no stake in declaring the project a success or a failure.

One such survey could take a group of three individuals. One group of participants would be given an older narrative dated March 18th, 2018, before the transition narrative was added and the character encounters were smoothed out. The second group would be given a more recent narrative dated March 31st, 2018, after the transition narrative was added. The third group would be given the chance to play the game themselves and assess the narrative they themselves have generated.

The Survey

The survey in its current form consists of six questions. The first two are a rating from 1 to 5 regarding the quality of the narrative. The next three questions are short answer questions that assess areas of the narrative that readers believe were well developed and areas where the narrative could be improved. The final question asks which file the individual read so their answers could be grouped in the right category.

Question	Question Type
How much did you feel the story was written by a Human Author vs a machine?	1 to 5
How detailed was the story?	1 to 5
What areas of the story flow Well?	Long Answer
What areas of the story are too mechanical?	Long Answer
Where do you feel the Story or Game can be improved?	Long Answer
What is the name of the file you read?	Short Answer

Prediction

The March 18th file is expected to get lower scores than the other two categories, especially in the first question. The older file is also expected to get along response to the Long Answer questions that ask for areas where the story can be improved (Questions 4 and 5).

The individuals who play the game themselves are likely to give the narrative a slightly better score than the March 31st file. The reason this could be the case is that they will be able to better see their own hand in the story they read and thus give the story higher marks than the March 31st file that is generic.

Time Constraints

Unfortunately, time constraints prevented a successful survey from being carried out. For such a survey to be carried out and the results published, approval would have had to be procured from the Research Integrity Committee and the idea for a survey and the need for committee approval came late in the development of the project. Consequently, such a survey would have to be carried out as part of a future project.

Chapter 7: Reflections

Future Directions

There were many aspects of the game that have not been developed but are specified in the TTLG Document on which the game was based. While many elements from the document were included, many others were omitted or hinted at. For instance, there was a common story element regarding the Lion and the Unicorn, yet the only hint in the game regarding their relationship is a remark made by the Lion while talking with Alice.

Other elements from the document include the Bread and Butterfly, the Dark Shop (and the sheep who serves as its shopkeeper), a conversation between Alice and a gnat, the owl and the observatory, amongst other things. All of these are elements of the story that are not addressed in the game.

Should production of the game continue, adding more of these elements would be a worthy pursuit. I myself would likely start with the Dark Shop. The narrative state should also be expanded upon, even with only the elements that currently exist. Adding more granularity would enhance the narrative and make it easier to produce a story file that reads more naturally to human readers. Other work could focus on typos in the current narrative.

Applicability of Inform 7

Inform 7 proved to be a great tool for developing the game. Having a series of rules and dividing them up between *before, after,* and *instead* rules proved beneficial when it came to describing game specific actions the player could take in the game-board. With that said, it would be nice if Inform 7 offered functions that certain rules could use. There are three actions regarding the Restaurant of Walrus and the Carpenter that apply to different tables but essentially do the same thing. Inform 7 could be improved if it was possible to write a function that refactors the common functionality of those actions and have their rules call that function and specify objects or values specific to the individual action.

Although I myself am more accustomed to using languages like C++, C#, or even Java, implementing this game in any of those languages would have been difficult as I

would have had to define classes such as the character class, the room class as well as implement the system of rules and shortcuts that Inform 7 provides. With that said, it would have been easier to implement a GUI to make game play somewhat easier. Many people are not used to using the command-line (which Inform 7 uses) and it is difficult to develop a command-line based user experience satisfying to those unfamiliar with the command line.

Before developing a flexible system of narrative for the game, I chose to lay the foundations of the game down when development began. My familiarity with Inform 7 when I began this project was no-where near what it is today and thus, approached this project similarly to how I would have done it in a mainstream object-oriented language. I anticipated needing extra attributes for the room class and thus created the *board area* and added attributes to that, not knowing that I could add those attributes directly to the room class.

Approach

While working on specific sections of the game, I was focused on developing multiple story-lines that I felt could easily be featured in the game. Earlier story-lines, like the Walrus and the Carpenter's Restaurant as well as the Vegetable/Flower Garden, were fleshed out heavily. The third story line developed, Humpty-Dumpty, has fewer details than the first two but still has a significant amount of action.

There was also an evolution on how I implemented certain details. For instance, when I was developing the layout of the Vegetable Garden, the document called for five major sections: Demo Area, shed, Product Stand, Green-house, and Hydroponics. I created five rooms to represent each of them as well as implemented five actions to allow access to each of them. Later, when implementing three tables as well as the inventory of the Restaurant, I simply created three lists of characters to represent customers seated at the three tables. I also simply declared four numbers to represent the quantity of each menu item.

Work on the narrative truly started a couple months prior to the defense. My thinking was that the actions and events had to be fleshed out before narrative could be applied to it. However, as the deadline drew nearer, I realized that narrative could be

applied to the story elements already well developed while work on newer story elements was underway. This approached lasted until late February when I applied narrative to story-lines while developing them. It was also that time I began to contemplate how to glue chunks of narrative together in a way that flowed somewhat naturally.

Appendix A: Rooms

The game provides two major sets of rooms: game board rooms and sub-rooms. Game board rooms are rooms placed in a certain location within the 8-by-8 game board. Sub-rooms are rooms that are placed on some game board room. The following table provides an overview of the game board rooms and their location on the game board.

Room Name	Coordinate	Room Name	Coordinate
Castle1	(0,0)	Upper South Shore	(0,4)
NP Meadows Far West	(1,0)	Railroad8	(1,4)
NP Meadows West	(2,0)	LPS	(2,4)
NP Meadows Midwest	(3,0)	Railroad9	(3,4)
NP Meadows Mideast	(4,0)	TT House	(4,4)
NP Meadows East	(5,0)	Railroad10	(5,4)
NP Meadows Far East	(6,0)	Railroad11	(6,4)
Castle2	(7,0)	Vegetable Garden	(7,4)
Upper North Shore	(0,1)	Mid South Shore	(0,5)
Messenger's House	(1,1)	Railroad12	(1,5)
NP Meadows low West	(2,1)	Railroad13	(2,5)
NP Meadows low Midwest	(3,1)	Railroad14	(3,5)
NP Meadows low Mideast	(4,1)	Wood of Forgetfulness	(4,5)
NP Meadows Low east	(5,1)	Railroad15	(5,5)
NP Meadows low Far east	(6,1)	Railroad16	(6,5)
NP Meadows low Castle	(7,1)	Blank21	(7,5)
Lower North Shore	(0,2)	Low South Shore	(0,6)
Railroad1	(1,2)	Haigha's House	(1,6)
Railroad2	(2,2)	Southern Meadow Upper East	(2,6)
Railroad3	(3,2)	Southern Meadow Upper Mideast	(3,6)
Humpty Dumpty's Wall	(4,2)	Southern Meadow Upper Midwest	(4,6)
Railroad4	(5,2)	Southern Meadow Upper West	(5,6)
Railroad5	(6,2)	Hatta's House	(6,6)
NP Meadows S	(7,2)	Lower Woods	(7,6)
Walrus_and_Carpenter's_Restaurant	(0,3)	Castle3	(0,7)
Railroad6	(1,3)	Southern Meadow Far West	(1,7)
Launch Point	(2,3)	Southern Meadow West	(2,7)
Dark Shop	(3,3)	Southern Meadow Midwest	(3,7)
Great Lake	(4,3)	Southern Meadow Mideast	(4,7)
Small Meadow	(5,3)	Southern Meadow East	(5,7)
Railroad7	(6,3)	Southern Meadow Far East	(6,7)
Flower_Garden	(7,3)	Castle4	(7,7)

Table 4: Full listing of the rooms in the game board. The rooms are sorted according to their location on the game board.

Many of these rooms have content within them. Some of the content is as simple as having a sign set up while some content is as elaborate as the Walrus and Carpenter Restaurant. Other rooms simply have a few storylines but little variety while four rooms have the means to travel to different locations.

Room	Content Type	Room	Content Type
Castle1	EP	Humpty Dumpty's Wall	C
Castle2	E	Walrus_and_Carpenter's_Restaurant	RPC
Castle3	E	Launch Point	S
Castle4	E	Great Lake	CP
Railroad1	T	Small Meadow	S
Railroad5	T	Flower_Garden	RCP
Railroad12	T	TT House	C
Railroad16	T	Lower Woods	S
Vegetable Garden	RP	Southern Meadow Upper Mideast	S
		NP Meadows Midwest	S

Table 5: Room and Content type

The following letters indicate the content type available in a given room:

E: Endpoint – Rooms upon which entering means you win the game. These rooms require you to earn enough points under a certain category before entering.

P: Points – Rooms marked with a P mean they can contribute points to the player's score. Most point opportunities come from the Walrus and Carpenter Restaurant. However, since they player can purchase vegetables (Vegetable Garden) or fish salmon (Great Lake) in order to earn coins, these rooms are marked. Castle1 and Humpty Dumpty's Wall also are involved in earning coins and are marked with a P as well.

T: Transport – Rooms marked with a T provide commands that let you travel to other rooms also marked with a T.

R: Rooms – Rooms marked with an R have rooms within them.

C: Conversation – Rooms marked with a C support conversations with certain characters that are unique to that room

S: Sign – Rooms marked with a sign have signed through which the player can read for directions to various points of interest.

Initially, many rooms were simply considered *Blankx* where x was a number between 1 and 33. Below is an image of the map with the blank rooms marked with a dark yellow color.

Figure 6: Early Map Layout. All of the blanks have been highlighted.
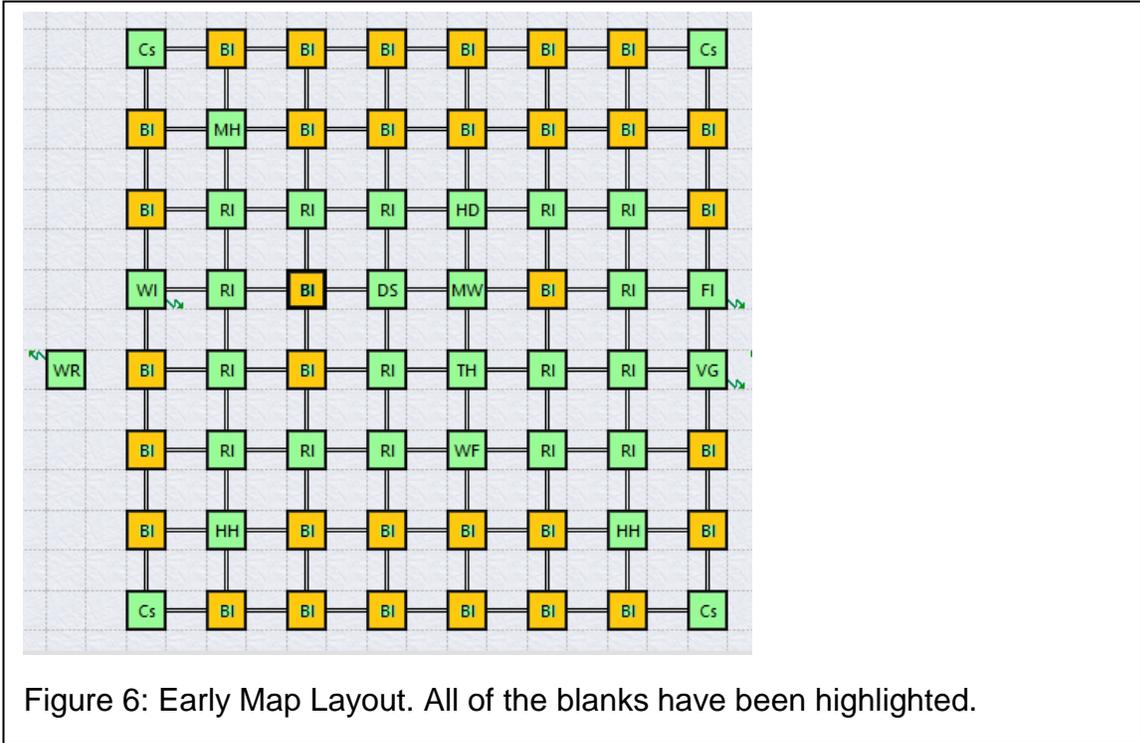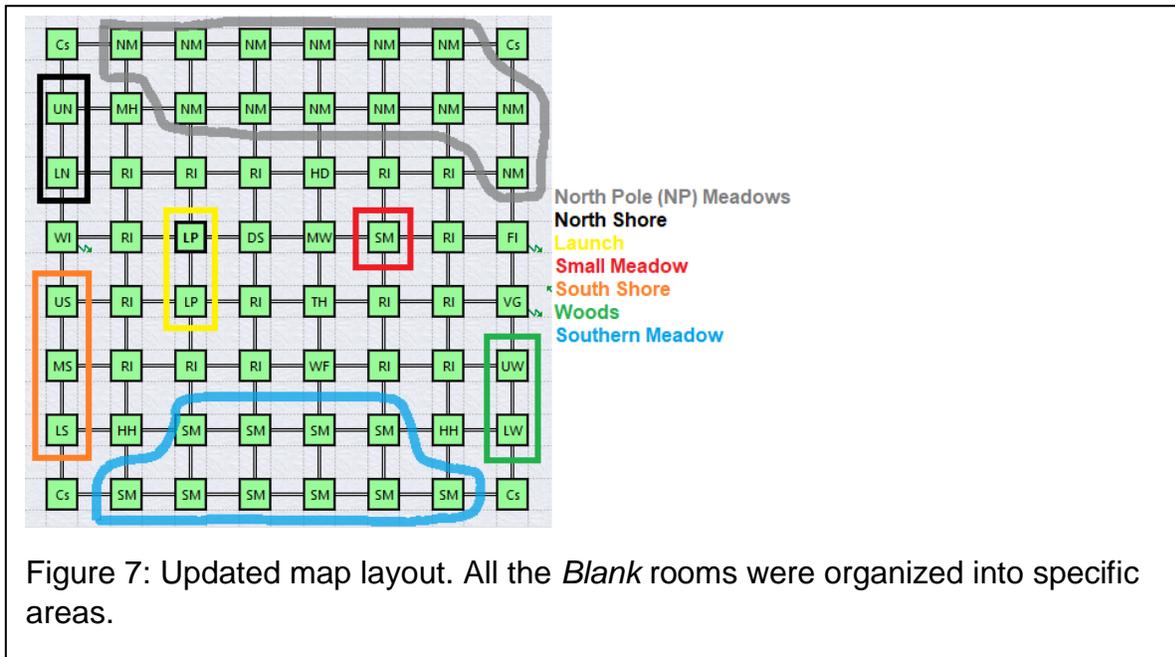
However, the game under this set-up was deemed unplayable and so the blank rooms were partitioned into different sections and made to be a part of various environments – usually a meadow. The following image showcases the partitions.

Figure 7: Updated map layout. All the *Blank* rooms were organized into specific areas.

As seen from the previous table, five rooms contain signs. The following table showcases the sign and their associated text.

Room (coordinates)	Sign text
Launch Point (2,3)	This is the starting point in the game.
	The Restaurant of Walrus and Carpenter is two blocks west
	Messenger's House is one block west and two blocks north.
	The Sign in the North Pole Meadows is three blocks north and 1 block east.
	The Dark Shop is one block east.
	The Lake is two blocks east.
	Tweedle-dee and Tweedledum's house is two blocks east and one block south.
	The Wood of Forgetfulness is 2 blocks east and 2 blocks south.
	The Sign in the Small Meadow is three blocks east.
	the Sign in the Southern Meadows is three blocks down and one block east
Small Meadow (5,3)	Welcome to the Small Meadow.
	The Launch Point is three blocks west.
	The Dark Shop is two blocks west.
	The Lake is one block west.
	Humpty Dumpty's wall is one block north and one block west.
	The Flower Garden is two blocks east.
	The Vegetable Garden is two blocks east and one block south.
	The Sign in the Lower Woods is two blocks east and three blocks south.
Lower Woods (7,6)	Welcome to the southern Woods.
	The South east castle is one block south.
	Hatta's House is one block west.
	The Sign in the Southern Meadows is four blocks west.
	The Vegetable Garden is two blocks north.
	The Flower garden is three blocks north.
	The Sign in the Small Meadow is three blocks north and two blocks west.
Southern Meadow Upper Mideast (3,6)	Welcome to the Southern Meadows.
	The Launch Point is three blocks north and one block west.
	Haigha's House is two blocks west.
	The South west Castle is three blocks West and one block south.
	The Wood of Forgetfulness is one block east and one block north. Tweedle-dee and Tweedledum's house is one block east and two blocks north.
	Hatta's House is three blocks east.
	The Sign in the Lower Woods is four blocks east.
NP Meadows Midwest (3,0)	Welcome to the North Pole (NP) Meadows.
	The North West Castle is three sections to the west.
	The North East Castle is four sections east.
	Messenger's House is two blocks west and one block south.
	Humpty Dumpty's wall is one block east and two blocks south.
	The Launch Point is one block west and three blocks south.

Table 6: Rooms with their signs. The rooms listed have a sign in them and the player can opt to read the sign during gameplay. The text showcases what the player sees when running the *Read Sign* command in one of these rooms.

Most of the information presented was on rooms on the game board. However,

there are seven other rooms in the game that are on a game board room. These rooms are "W&C Restaurant", "Elevator", "Product Stand", "Shed", "Demo Area", "Green

Room	Parent Room	Entry Command	Exit Command
W&C Restaurant	Walrus_and_ Carpenter's_Restaurant	go in	go out
Elevator	Flower Garden	go in	go out
Product Stand	Vegetable Garden	Enter Product Stand	Exit to Garden
Shed		Enter Shed	
Demo Area		Enter Demo Area	
Green House		Enter Green House	
Hydroponics		Enter Hydroponics	

Table 7: List of sub-rooms not directly featured in the game board. Also listed are the parent rooms in which the sub-rooms are located as well as the command to enter/exit them.

House", and "Hydroponics". The following table showcases the sub-rooms and the board room that they are found in.

Appendix B: Characters

Many characters declared in the game were derived from a table from the game specification seen below, with their development status highlighted.

White		Red	
Tweedle-dee	Daisy	Daisy	Humpty Dumpty
Unicorn	Haigha	Messenger	Carpenter
Sheep	Oyster	Oyster	Walrus
White Queen	"Lily"	Tiger-Lily	Red Queen
White King	Fawn	Rose	Red King
Aged man	Oyster	Oyster	Crow
White Knight	Hatta	Frog	Red Knight
Tweedledum	Daisy	Daisy	Lion

Table 8: List of characters and what their classifications. Light blue shaded characters are characters that have been Developed heavily in the game and have specific relevance to the story of Alice's quest to become a queen. Characters with a light red shading are characters that inform Alice of how she can become a queen during a chance encounter (The Red King might be brought to Tweedle-dee and Tweedledum's house to inform her, but this isn't a "chance" encounter). The Crow has a violet shading as the character has earned both a light blue and light red shading.

Green shaded characters are actually plants and are located in the Flower Garden.

As mentioned before, characters are a class in the game that are used for character interactions. The following code snippet declares the character class and their attributes.

Most of a character's attributes are geared towards adapting the narrative appropriately towards the unique circumstances. These include the "narrativeCounter" as we don't want to repeatedly say "She saw <insert name here>." time and time again, the *charDo* in case they introduce themselves to the player or describe themselves, the encountered boolean to determine if the story should describe their appearance, and the spoken boolean to determine if there should be some annoyance text added to the narrative.

[Declare the Character Class]
A character is a kind of Person. A character has a number called ID. A character has a number called narrativeCounter. The narrativeCounter of a character is usually 0. A character has a number called HungerRate. A character has a number called orderVal. A character has some text called charDo.

[Give the player OrderVal so that Alice can take orders from characters]
the player has a number called orderVal.

[Now Describe the character through a series of booleans:
       1. Served (normal) or unserved (hungry)
       2. Encountered (The player has seen the character before) or nonEncountered (player has yet to come across the character)
       3. Spoken (player has spoken with the individual) or unspoken (player has yet to make contact)]
A character can be served or unserved. A character is usually served.
A character can be encountered or unencountered. A character is usually unencountered.
A character can be spoken or unspoken. A character is usually unspoken.

Code Listing 6: Declaration of the character class. Here is the list of attributes and their description.
ID – number – used to identify a white character or a red character (2,3 or 1,4 respectively) and whether the character is a messenger or a pawn (1,2 or 3,4 respectively). This attribute is a legacy attribute and is not used by the game
narrativeCounter – number – keeps track of the last time the player has encountered the character to determine it the character's sighting should be in the narrative
HungerRate – number – Determines whether the character will head to the Restaurant (when it gets below 4), 10 means the character is NOT hungry.
orderVal – number – used to determine what to order at the restaurant.
charDo – text – how the character describes themselves if the player asks about their occupation.
served/unserved – boolean – served if the character is not hungry and unserved if the character is
encountered/unencountered – boolean – determines whether the player has encountered the character and thus whether a description needs to be added to the narrative.
spoken/unspoken – boolean – whether the character has already described themselves to the player (they'll do it again but express subtle annoyance).

Appendix C: Commands

Most of this section will focus on commands relevant to the game. Commands during an inform game enable the player to perform some sort of action. Actions described in the game typically have four main components: a declaration, command, a before rule, and after rule. The table below describes these four sections.

Action Component	Description
Declaration	Declares the action and whether any objects should be specified in the command
Understand (command)	Names what the player would type to activate the command
Before Rule	Used to check for conditions where the action should not (or could not) be done, such as "Go fishing" when there is no body of water nearby.
After Rule	Used to execute the action.

Table 9: Components of an Inform 7 action. Each of these are used extensively in the game to implement the multiple commands available to the player as well as determine which command should be permitted based off of the player's location and circumstances.

Other components could be added, such as an instead rule (which depicts and alternative action to take), or a consecutive turn count. However, such sections were never added in the game. Most of the functionality for an *instead* rule fit inside a *Before* rule and the consecutive turn count was replaced by a manually coded counter (in the Flower Garden Section).

"w"; "west"	Takes the player to the room west of the current one
"e"; "east"	Takes the player to the room east of the current one
"n"; "north"	Takes the player to the room north of the current one
"s"; "south"	Takes the player to the room south of the current one
"wait"	Allows time to pass (one turn)
"List Local Characters"	Lists each character present at your location

Table 10: Core Commands. These commands can be run at almost any time, though a before-rule might decide it can't currently be run and thus block it. For instance, if you're on the western edge, the *west* command won't work, and a similar principle holds for the other edges. The directional commands also won't work in a *sub-room*. Apart from that, these commands will always run.

The first set of commands are commands that can be run almost anywhere. Granted, if you happen to be in a room on the east side of the board and there is no room to the east, then the "east" command won't work. All the following commands (except for *List Local Characters*) are defined by Inform, not the game.

Command	Description	Room where Command doesn't work
"Head to northeast station"	Takes player to the Northeast station (Railroad1)	Railroad1
"Head to northwest station"	Takes player to the Northwest station (Railroad5)	Railroad5
"Head to southeast station"	Takes player to the Southeast station (Railroad12)	Railroad12
"Head to southwest station"	Takes player to the Southwest station (Railroad16)	Railroad16

Table 11: Transportation station commands. These commands allow you to speed up transportation by theoretically taking the train. Obviously, the commands will not take you to your current location, but they can take you to the other stations.

The next four commands are relevant to the four active railroad stations. They are only active in "Railroad1", "Railroad5", "Railroad12", and "Railroad16" and in each station, one of those commands is redundant as you are already at the location the command takes you to.

Command	Rooms Relevant	Description
"go in"*	Walrus_and_Carpenter's_Restaurant; Flower_Garden	Allows the user to enter W&C Restaurant or Elevator respectively
"go out"	W&C Restaurant; Elevator	Allows User to Return to the Game Board
"exit to garden"	Demo Area; Hydroponics; shed; Product Stand; Green House	Returns the User to the Vegetable Garden
"Enter Product Stand"	Vegetable Garden	Allows User to enter Product Stand
"Enter Shed"	Vegetable Garden	Allows User to enter Shed
"Enter Demo Area"	Vegetable Garden	Allows User to enter Demo Area
"Enter Green House"	Vegetable Garden	Allows User to enter Green House
"Enter Hydroponics"	Vegetable Garden	Allows User to enter Hydroponics

* Technically, the command will also work in the Vegetable Garden, but it only takes you to the Product Stand, hence the other commands created for the Vegetable Garden.

Table 12: Commands for entering and exiting sub-rooms. Special commands were created for the rooms within the vegetable garden. Inform 7's commands (*go in* and *go out*) suffice for the rooms outside the Vegetable Garden.

The Next set of commands handle the players ability to enter or exit rooms within a room. There are three rooms in the Game Board that have rooms within them: Walrus_and_Carpenter's_Restaurant, Flower_Garden, and Vegetable Garden. In the Flower Garden, there is a room called Elevator, but it is "not discovered" and there is currently no way to discover it. However, it is still possible to enter.

The one area of the game with the most development is the Restaurant owned by Walrus and Carpenter. A couple character attributes are used by this storyline as are two of the three currently available coin types. Consequently, a fair number of

Command	Description	Room Relevant
"Apply for a job"*	Allows Player to be employed by the restaurant	W&C Restaurant
"Order from Table 1"	Allows Player to take an Order from Table 1	W&C Restaurant
"Order from Table 2"	Allows Player to take an Order from Table 2	W&C Restaurant
"Order from Table 3"	Allows Player to take an Order from Table 3	W&C Restaurant
"Dispatch Table 1"	Allows Player to dispatch the order to Table 1	W&C Restaurant
"Dispatch Table 2"	Allows Player to dispatch the order to Table 2	W&C Restaurant
"Dispatch Table 3"	Allows Player to dispatch the order to Table 3	W&C Restaurant
"Review Order"	Allows player to look over the order	W&C Restaurant
"Drop off Food"	Allows Player to deposit fish and vegetables to the Restaurant's inventory	W&C Restaurant
"Check Inventory"	Allows Player to check supply count of the Restaurant	W&C Restaurant
"Drop Order"	Allows Player to Drop the Current Order without dispatching it	W&C Restaurant
"Purchase Vegetables"	Allows player to pick up vegetables	Vegetable Garden
"Go Fishing"	Allows Player to pick up a salmon	Great Lake

Table 13: Commands related to the Restaurant. Most commands require being in the Restaurant but the two that don't are still related to the Restaurant's storyline.

commands are dedicated to this story line as well. Apart from the "Apply for a job" command (*), all these commands require the player to be employed by the restaurant.

There are a few other commands available to the user under certain circumstances like specific story-line commands. The following Table lists the extra commands. Four of these commands relate to Humpty-Dumpty while the next two relate to Tweedle-dee and Tweedledum. The last command deals with signs that appear in certain rooms in the game.

Command	Description	Relevant Location
"Respond to Help"	Allows player to reach out to Humpty-Dumpty after he fell	Humpty-Dumpty's Wall
"Take Humpty-Dumpty"	Allows Player to pick up Humpty-Dumpty	Humpty-Dumpty's Wall
"complement necktie"	Allows Player to complement Humpty-Dumpty's necktie	Humpty-Dumpty's Wall
"complement belt"	Allows Player to complement Humpty-Dumpty's "belt" (it's a necktie)	Humpty-Dumpty's Wall
"Interrupt the twins"	Allows Player to interrupt Tweedle-dee and Tweedledum, enables the "Summon Crow" command	[Location of the Twins and their fight]
"Summon Crow"	Sends the Crow to break up the Twins fight	[Location of the Crow]
"Read Sign"	Allows player to look at a sign of directions	[Any room with a sign, see Appendix A]

Table 14: Miscellaneous commands. Technically, the "Take Humpty-Dumpty" command is Inform's "Take [noun]" command. While Inform usually restricts the command so people can't be taken, the Game suspends this rule.

Appendix D: Narrative States

The game uses numbers in the form of states to keep track of either what is going on or what has already happened. There are multiple areas in the game that use this technique: carpenter servicing customers at his and Walrus's Restaurant, the interactions between the player and the Latin Teacher at the Great Lake, The state of

State	Description
0	The Restaurant has not been built
1	The Restaurant has been built but the Carpenter is not doing anything at the moment.
2	The Carpenter is taking an order from Table 1
3	The Carpenter is dispatching an order to Table 1
4	The Carpenter is taking an order from Table 2
5	The Carpenter is dispatching an order to Table 2
6	The Carpenter is taking an order from Table 3
7	The Carpenter is dispatching an order to Table 3
8	The Carpenter is letting the player do his work

Table 15: The State of the Carpenter. Since the Walrus is usually tasked with retrieving the supplies (unless the player is employed), the Carpenter remains behind to respond to any customers. This state keeps track of what he is up to.

Tweedle-dee and Tweedledum's relationship, and interactions with Humpty-Dumpty. Recall that if Walrus and Carpenter's Restaurant is serving customers and the player is not employed or is not present, then the Carpenter will service the orders himself. The number is referred to in source code as "CarpenterState".

The state used to track interactions between the Latin Teacher and the player is less straight forward as these interactions have either happened or they haven't. Therefore, the variable used here ("LakeStat") is used as a boolean table where if a certain scenario occurs, a corresponding bit is set. The following table lists the scenario and which bit is set.

The checking mechanism takes advantage of boolean properties of numbers. For instance, if someone wants to check the first bit (bit 0), then they just need to ask if the number is even. If it is, then the target bit is 0 and the scenario has not yet occurred. Otherwise, that bit is 1 and we should not repeat the scenario.

Scenario (bit)	Scenario Description	Checking mechanism
1 (0)	Latin Teacher eating Jam and introduces Alice to a Latin version of *Twinkle Little Star!*	if LakeStat is even
2 (1)	Latin Teacher is eating Jam and offers some to Alice any day but "today"	if LakeStat divided by 2 is even
3 (2)	Latin Teacher offers to teach how to spin yarn (if Alice says no, then the state actually increases by 8 instead of 4 and Alice will be allowed another opportunity at a later time)	if LakeStat divided by 4 is even

Table 16: Bit marking occurrence of a *Great Lake* story line.

If the bit of interest is later down the line, then we need to divide it by the appropriate power of 2 in order to use the even technique. For instance, the second bit (bit 1) can be divided by 2 and checked as even or odd.

The relationship between Tweedle-dee and Tweedledum is handled by a number called "TweedleSwitch". The following table showcases TweedleSwitch's possible values and their meaning.

TweedleSwitch Value	Description
0	Tweedle-dee moves randomly, Tweedledum gravitates towards Tweedle-dee
1	Tweedledum moves randomly, Tweedle-dee gravitates towards Tweedledum
2	Both Tweedle-dee and Tweedledum Gravitate towards their House (TT House)
3	Tweedle-dee and Tweedledum commence fighting
4	Tweedle-dee and Tweedledum are interrupted by the player
5	Tweedle-dee and Tweedledum resume fighting
6	The Crow has broken up their fight and carried them home

Table 17: State of Tweedle-dee and Tweedledum. Used by the game to keep track of what Tweedle-dee and Tweedledum are up to and how far the player has invested in their story.

When the game begins, TweedleSwitch cycles through the states 0-2, starting at 0 and this cycle is broken when Tweedle-dee and Tweedledum encounter each other and begin fighting. They can only be interrupted by the player once (when the state is 3). Other than that, the states of 3 and 5 are essentially the same.

State	Description
0	Nothing has happened between Alice and Humpty-Dumpty
1	Humpty-Dumpty has fallen off his wall
2	Alice and Humpty-Dumpty begin a conversation
3	Alice has responded to Humpty-Dumpty's cry for help
4	Alice and Humpty-Dumpty continue their conversation
5	Alice has taken Humpty-Dumpty
6	Alice and Humpty-Dumpty complete their conversation
7*	Alice has abandoned Humpty-Dumpty
8	Alice has Dropped Humpty-Dumpty off at his requested Castle
9	Alice and Humpty-Dumpty have finished their conversation, Alice has returned to the wall

* Mode has not been implemented in the game

Table 18: State of Humpty-Dumpty. There are two lines the story can follow: a conversation and a fall.

The state used to track the interactions between Alice and Humpty-Dumpty is relatively straight forward, although there are two lines that their interactions can take. They can either have a conversation or Humpty-Dumpty can fall off his wall, requiring Alice to take

him to the North-West Castle. The following table showcases the value/description of "HumptyStat", the variable used to keep track of the state.

Between State 0 and 7 exclusively, the states diverge between even and odd numbers with even numbers being the friendly conversation storyline and the odd numbers being the fall-storyline. As the friendly conversation story line ends at state six, this opens even number 8 to be used in the fall storyline and is subsequently used to declare that Humpty-Dumpty has gotten help and Alice has received green coins for her act.

Appendix E: Story Files from Survey

March 20, 2018

Her name was Alice! That much she knew, even as her dreams began to fade, and her consciousness returned to her. She did not remember what was going on before she fell asleep, but she now felt herself waking up. As her vision came into focus, she began to process the environment in which she found herself.

She appeared to be in a light meadow, with a few flowers here and there. She could see in one direction a vast sea. Two Castles were situated on the sea shore, both being considerably far away from each other. To the opposite side of Alice stood a dark building that looked long abandoned.

It dawned on Alice that she had no idea where she was, but she felt around a pocket in her dress and felt a parchment. But then she took another look and discovered a sign was present and left Alice wondering if she should read it.

After moving to the Great Lake, Alice saw The Frog. The Frog was a green creature the size of a mouse with long legs and a moist body, beady eyes and large neck that cloaks any chin it may have.

Alice went over to where the large lake was located. Ahead of her was a small lake with a dock. A row boat was tied to the dark. Sitting on a bench on the dock was a disheveled woman.

 She was spinning yarn using a drop spindle.

"What are you doing?" Alice asked.

"I'm spinning yarn. Would you like me to show you how?" she responded.

"Sure", Alice replied.

She sit down next to her and found out she was a Latin Teacher and quickly learned the basics of drop spinning. She invited Alice to her cabin, and Alice went with her. Inside the living room of the cabin was a spinning wheel, a loom, and a shelf with two baskets. One basket was labeled "LINUM" and one was labeled "???."

"What do you think is in the left basket?" she asked.

"Linen?" Alice guessed.

"Close," she said. "It's flax. Linum is the Latin word for flax, the plan from which we get the fibers to make linen."

Alice found herself walking towards (or next to) a wall and to her surprise, she heard a voice from the top of the wall! "State your name!" it said.

"My name is Alice!" she replied, stopping.

"What does Alice mean?" said the voice.

 "I don't know!" Alice responded.

The voice then goes on a lecture! "You don't know? Every name means SOMETHING. My name is Humpty Dumpty, and my name means the shape I am in, and a good handsome shape it is, too. With a name like yours, you might be any shape, almost!"

Alice noticed a strip of cloth tied around his midsection and decide to compliment him on it, though she wasn't sure if it was a necktie or a belt.

"That's a nice necktie," she said.

"It's a cravat, child, and a beautiful one. It is a present from the White King and Queen," Humpty replied.

"Why did they give it to you" Alice asked.

"It was an un-birthday present," Humpty replied.

"What is an Unbirthday Present!"

"A present given when it isn't your birthday, of course. How many days are there in a year?", he answered.

"365"

"And, how many birthdays do you have each year?"

"1"

"So how many un-birthdays do you have each year?"

"364"

Alice wanted to apply for a job at the W&C Restaurant. Alice cautiously filled out an employment form and found it relatively easy to fill out. She was unsure of her qualifications, but this wasn't a high-paid position, just something to help her get by in this world. The Carpenter accepted her application and she became an employee of the Restaurant. The Walrus called Alice over. He told her that the Restaurant needed new supplies."We need, 10 salmon,10 carrots,10 corn cobs,10 pea salmon,"he said, handing Alice some money meant to purchase the Vegetables before leaving her.

Alice was now in a wooded area, moving towards a clearing. In the clearing, she saw 2 tiny cottages, each a mirror image of the other. Alice heard a sound like the puffing of a large steam-engine. She left the clearing and walked into the woods. There was a man fast asleep. He was wearing a tall red night cap with a tassel and he was snoring loudly. She approached the man.

"Hello!" Alice said cautiously.

Suddenly, the man wakes up. Opening his eyes slightly, he yawned and stretched before standing up with difficulty. He pulled the night cap off of his head. Underneath the night cap he is wearing a red crown. He spoke, "What day is it?"

"At this point, I don't know," Alice said.

"It's November 4 and tomorrow is the big bonfire. You ARE here for the bonfire, aren't you? And, you look like a pawn. I assume you are trying to become a queen in time for the bonfire!"

"How do I become a queen?" Alice asked.

"By entering one of the four castles in the corners of the realm, of course! Now, I must be going!" he replied.

"But Wait!" Alice called. "Is there any requirement for entering a castle or ca I simply walk in?"

"Well, each castle requires a payment of seven coins, but there are four types of coins corresponding to each Castle," he answered. "Green coins will get you into the North-West Castle. Blue coins to the North-East Castle, Red coin o the South-West Castle, and the South East Castle will take Yellow Coins that don't even exist." With that, he was gone.

Alice found herself in the Vegetable Garden. It had the appearance of a small central park with various features in the corners and the center. In the center, there was a product stand, where vegetables were being sold. In one corner, there was a shed, where Alice suspected many gardening supplies were stored. There was a Green House on another corner. The remaining corners contained a Demo Area and a "Hydroponics" area, according to the sign.

Alice slowly approached the Product stand, where the Vegetables were. An individual she didn't recognize stood at the stand. "Excuse me sir" Alice said cautiously. "Are you selling?"

"Yes maam!" the man said. "What will it be?"

" ten carrots, ten cobs of corn ,and ten servings of peas," Alice said.

"Coming Right up" the man said, after Alice placed the money on the desk. "Here you go!" he said, placing a box on the tand that Alice took.

"Thank You sir" she said, before continuing on her way.

Alice was now in a wooded area, moving towards a clearing. In the clearing, she saw 2 tiny cottages, each a mirror image of the other. After moving to the Walrus_and_Carpenter's_Restaurant, Alice saw Messenger. Messenger was a man with a simple cloak and carrying what appeared to be a scroll.

After moving to the Dark Shop, Alice saw Haigha. Haigha was .

After moving to the Great Lake, Alice saw Red King, The Frog, and Red King was a crimson bearded man with a long red cape and a red-dome-golden crown.

Alice went over to Table 3 and took an order from Haigha.Alice looked at the current order: 1 salmon sandwiches, 0 Corn Servings, 0 pea Servings, and 0 carrots.

Haigha took the order and handed Alice a Red Coin. Looking around, you notice that there are no customers at the moment and thus you breath a sigh of relief.

Alice looked over at the third table but it was relatively unoccupied.After moving to the Great Lake, Alice saw Red King, The Frog, and

Alice saw the small lake with a dock and rowboat tied as well as the Latin Teacher she met earlier.

She was looking longingly at a jar of jam with a spoon from a jar.

"Hello there, would you like some jam?" the woman asked.

"Yes please," Alice replied.

"What day would you like it?" she asked.

The question seemed so absurd to Alice that she wondered if it was even asked. It was apparent enough that it was. "Why, today, I imagine!" she remarked.

"Sorry. You can have jam yesterday and jam tomorrow, but you can never have jam today," the woman said.

"That makes no sense!" Alice exclaimed!

"But it makes PERFECT sense. I'm a Latin teacher, and in Latin the word "iam" or "jam" means "now" but it can only be used in the past or future tense. To say "now" in the present tense in Latin you have to say "nunc."" she said.

Alice Found herself in the Flower Garden. She looked around but it appeared to be devoid of any sentient being. She could see Daisies, Larkspur, Roses, Tiger Lilies, Dogwood. There was even some weeds known as Kudzu! Alice began to look around. She began to hear giggling, but when she turned to the direction of the laughter, she couldn't see who it could have been.

"Whose there?" she asked.

"We are" said a voice.

It startled Alice to know that it came from the flowers themselves! After moving to the Flower_Garden, Alice saw Daisy1, Lily, Daisy2, Daisy3, Tiger-Lily, Rosey, Daisy4, and Daisy1 was a white flower with multiple pedals and a yellow center. Lily was a red flower with long thin pedals and long sprouts. Daisy2 was a white flower with multiple pedals and a yellow center. Daisy3 was a white flower with multiple pedals and a yellow center. Tiger-Lily was a birght orange flower with long thin pedals. Rosey was a red flower with thick pedals. Daisy4 was a white flower with multiple pedals and a yellow center.

As Alice watered the plant, the plant revealed a Golden Coin and tossed it to her. But wait! Wasn't she told that such coins don't exist? Would these work in getting her into the South-East Castle? Would it really be as simple as watering all of these plants?

"I thought these didn't exist!" Alice commented.

"No one knows about them!" said the Lily.

"Except for the Lion", said the Tiger-Lily."Ahh, the Lion" Lily said.

"But why give them to me?" Alice asked.

"We've been waiting for someone to water us" Rose said. "And it's not like we need these coins to be secret, anyway!"

Alice cautiously approached the Castle Gates. She was approached by a knight in used-silver armor. "How many Yellow Coins do you have?" the knight asked.

Alice cautiously counted her coins. 7 Yellow Coins, YES! 0 more than she needed. She also had 4 Blue Coins, 0 green keys, and 1 red keys, but they didn't matter for this castle. She showed the knight the Yellow Coins she had collected.

"Come with me!" the knight said, to which Alice obliged.

Alice followed the knight to a gathering of knights gathered in a throne room, one of whom held a crown. Once Alice was a few feet away from the knight, she instinctively kneeled. The knight placed the crown on Alice's head and said "Rise, Queen Alison." She rose and turned to the crowd of knight, who began kneeling before her.

March 31st 2018
* //BEB4BC65-B65B-4BA7-B6AE-CCCD93BA7002// story2

Her name was Alice! That much she knew, even as her dreams began to fade and her consciousness returned to her. She did not remember what was going on before she fell asleep, but she now felt herself waking up. As her vision came into focus, she began to process the environment in which she found herself.

She appeared to be in a light meadow, with a few flowers here and there. She could see in one direction a vast sea. Two Castles were situated on the sea shore, both being considerably far away from each other. To the opposite side of Alice stood a dark building that looked long abandoned.

It dawned on Alice that she had no idea where she was but she felt around a pocket in her dress and felt a parchment. But then she took another look and discovered a sign was present and left Alice wondering if she should read it.

After moving to the Great Lake, Alice saw The Frog. The Frog was a green creature the size of a mouse with long legs and a moist body, beady eyes and large neck that cloaks any chin it may have.

Alice went over to where the large lake was located. Ahead of her was a small lake with a dock. A row boat was tied to the dark. Sitting on a bench on the dock was a disheveled woman.

She was looking longingly at a jar of jam with a spoon from a jar.

"Hello there, would you like some jam?" the woman asked.

"Yes please," Alice replied.

"What day would you like it?" she asked.

The question seemed so absurd to Alice that she wondered if it was even asked. It was apparent enough that it was. "Why, today, I imagine!" she remarked.

"Sorry. You can have jam yesterday and jam tomorrow, but you can never have jam today," the woman said.

"That makes no sense!" Alice exclaimed!

"But it makes PERFECT sense. I'm a Latin teacher, and in Latin the word "iam" or "jam" means "now" but it can only be used in the past or future tense. To say "now" in the present tense in Latin you have to say "nunc."" she said.

After saying good bye to the Woman, she walked around the lake and proceeded to go slightly further to the north. Alice found herself walking towards (or next to) a wall and to her surprise, she heard a voice from the top of the wall! "State your name!" it said.

"My name is Alice!" she replied, stopping.

"What does Alice mean?" said the voice.

 "I don't know!" Alice responded.

The voice then goes on a lecture! "You don't know? Every name means SOMETHING. My name is Humpty Dumpty, and my name means the shape I am in, and a good handsome shape it is, too. With a name like yours, you might be any shape, almost!"

Alice noticed a strip of cloth tied around his midsection and decide to compliment him on it, though she wasn't sure if it was a necktie or a belt.

"That's a nice necktie," she said.

"It's a cravat, child, and a beautiful one. It is a present from the White King and Queen," Humpty replied.

"Why did they give it to you" Alice asked.

"It was an un-birthday present," Humpty replied.

"What is an Unbirthday Present!"

"A present given when it isn't your birthday, of course. How many days are there in a year?", he answered.

"365"

"And, how many birthdays do you have each year?"

"1"

"So how many un-birthdays do you have each year?"

"364"

After her awkward chat with Humpty-Dumpty, Alice traveled west and somewhat south, coming across the wooden restaurant in the shore. Curious, she walked in and saw that there were three tables as well as a kitchen in the background. Alice wanted to apply for a job at the W&C Restaurant. She cautiously filled out an employment form and found it relatively easy to fill out. Alice was unsure of her qualifications, but this wasn't a high-paid position, just something to help her get by in this world. The

Carpenter accepted her application and she became an employee of the Restaurant. The Walrus called Alice over. He told her that the Restaurant needed new supplies. "We need, 10 salmon,10 carrots,10 corn cobs,10 pea salmon," he said, handing Alice some money meant to purchase the Vegetables before leaving her.

After moving to the Great Lake, Alice saw The Frog.

Emerging from the Restaurant, Alice walked East, through the meadow and the point where she started her journey from. She continued past that point and past the abandoned shop close by. Instead, Alice saw the small lake with a dock and rowboat tied as well as the Latin Teacher she met earlier.

After chatting with the Teacher and traveling east, Alice Found herself in the Flower Garden. She looked around but it appeared to be devoid of any sentient being. She could see Daisies, Larkspur, Roses, Tiger Lilies, Dogwood. There was even some weeds known as Kudzu! Alice began to look around. She began to hear giggling, but when she turned to the direction of the laughter, she couldn't see who it could have been.

"Whose there?" she asked.

"We are" said a voice.

It startled Alice to know that it came from the flowers themselves! After moving to the Flower_Garden, Alice saw Daisy1, Lily, Daisy2, Daisy3, Tiger-Lily, Rosey, and Daisy4. Daisy1 was a white flower with multiple pedals and a yellow center. Lily was a red flower with long thin pedals and long sprouts. Daisy2 was a white flower with multiple pedals and a yellow center. Daisy3 was a white flower with multiple pedals and a yellow center. Tiger-Lily was a bright orange flower with long thin pedals. Rosey was a red flower with thick pedals. Daisy4 was a white flower with multiple pedals and a yellow center.

traveling south from the Flower Garden, it was not long before Alice found herself in the Vegetable Garden. It had the appearance of a small central park with various features in the corners and the center. In the center, there was a product stand, where vegetables were being sold. In one corner, there was a shed, where Alice suspected many gardening supplies were stored. There was a Green House on another corner. The remaining corners contained a Demo Area and a "Hydroponics" area, according to the sign.

Alice slowly approached the Product stand, where the Vegetables were. An individual she didn't recognize stood at the stand. "Excuse me sir" Alice said cautiously. "Are you selling?"

"Yes mamm!" the man said. "What will it be?"

" ten carrots, ten cobs of corn, and ten servings of peas," Alice said.

"Coming Right up" the man said, after Alice placed the money on the desk. "Here you go!" he said, placing a box on the stand that Alice took.

"Thank You sir" she said, before continuing on her way.

Alice was now in a wooded area, moving towards a clearing. In the clearing, she saw 2 tiny cottages, each a mirror image of the other. Alice heard a sound like the puffing of a large steam-engine. She left

the clearing and walked into the woods. There was a man fast asleep. He was wearing a tall red night cap with a tassel and he was snoring loudly. She approached the man.

"Hello!" Alice said cautiously.

Suddenly, the man wakes up. Opening his eyes slightly, he yawned and stretched before standing up with difficulty. He pulled the night cap off of his head. Underneath the night cap he is wearing a red crown. He spoke, "What day is it?"

"At this point, I don't know," Alice said.

"It's November 4 and tomorrow is the big bonfire. You ARE here for the bonfire, aren't you? And, you look like a pawn. I assume you are trying to become a queen in time for the bonfire!"

"How do I become a queen?" Alice asked.

"By entering one of the four castles in the corners of the realm, of course! Now, I must be going!" he replied.

"But Wait!" Alice called. "Is there any requirement for entering a castle or ca I simply walk in?"

"Well, each castle requires a payment of seven coins, but there are four types of coins corresponding to each Castle," he answered. "Green coins will get you into the North-West Castle. Blue coins to the North-East Castle, Red coin o the South-West Castle, and the South East Castle will take Yellow Coins that don't even exist." With that, he was gone.

She began the process of putting the food away. The vegetables (peas, corn, and carrots) would go in the Restaurant's refrigerator. The salmon would go into the freezer to be thawed later. After moving to the Flower_Garden, Alice saw Daisy1, Lily, Daisy2, Daisy3, Tiger-Lily, Rosey, and Daisy4.

As Alice watered the plant, the plant revealed a Golden Coin and tossed it to her. But wait! Wasn't she told that such coins don't exist? Would these work in getting her into the South-East Castle? Would it really be as simple as watering all of these plants?

"I thought these didn't exist!" Alice commented.

"No one knows about them!" said the Lily.

"Except for the Lion", said the Tiger-Lily."Ahh, the Lion" Lily said.

"But why give them to me?" Alice asked.

"We've been waiting for someone to water us" Rose said. "And it's not like we need these coins to be secret, anyway!"

As Alice received the latest set of Yellow Coins, she realized that her tally was now 7 Yellow Coin's, which would allow her to enter the South East Castle. This would be the first Castle that she would be able to visit. She wondered if she should go to the Castle immediately, or wait and remain at her current location.

After moving to the Lower Woods, Alice saw Tweedle-dee, and Tweedledum. Tweedle-dee was a teen boy with a yellow and reddish shirt and red overalls. He also wore a cap. Tweedledum was a teen boy with a reddish and yellow shirt and red overalls. He also wore a cap.

The Excitement at the chance to become Queen nearly overtook her. She said good bye to the flowers before departing the Garden, briefly passing through the Vegetable Garden on her way to the South East Castle and was feeling a sense of awe when it finally appeared. Alice cautiously approached the Castle Gates. She was approached by a knight in used-silver armor. "How many Yellow Coins do you have?" the knight asked.

Alice cautiously counted her coins. 7 Yellow Coins, YES! 0 more than she needed. She also had 4 Blue Coins, 0 green keys, and 0 red keys, but they didn't matter for this castle. She showed the knight the Yellow Coins she had collected.

"Come with me!" the knight said, to which Alice obliged.

Alice followed the knight to a gathering of knights gathered in a throne room, one of whom held a crown. Once Alice was a few feet away from the knight, she instinctively kneeled. The knight placed the crown on Alice's head and said "Rise, Queen Alison." She rose and turned to the crowd of knight, who began kneeling before her.

References

Alexander, L. (2014, October 22). *The joy of text – the fall and rise of interactive fiction.* Retrieved from The Guardian: https://www.theguardian.com/technology/2014/oct/22/interactive-fiction-awards-games

Carter, J., Dubbin, R., Eve, E., Heller, E., Jayzee, Mishima, K., . . . Wities, Z. (2009, June 5). *Alabaster*. Retrieved from The Interactive Fiction Database: http://ifdb.tads.org/viewgame?id=b2g8je1xxtqzei4u

Dougas, J. Y. (2001). *The End of Books - Or Books without End? Reading Interactive Narratives.* Ann Arbor, MI: University of Michigan Press.

Granade, S. (2010). *Introducing Inform 7*. Retrieved from Brass Lantern: The Adventure Game Website: http://brasslantern.org/writers/howto/i7intro.html

Infocom, Inc. (1989). *Learning ZIL.*

Mateas, M., & Stern, A. (2010). Writing Façade: A case study in procedural authorship. *Second Person: RolePlaying and Story in Games and Playable Media*, 183-208.

McCoy, J., Treanor, M., Samuel, B., Reed, A. A., Mateas, M., & Wardrip-Fruin, N. (2014). Social Story Worlds With Comme il Faut. *IEEE TRANSACTIONS ON COMPUTATIONAL INTELLIGENCE AND AI IN GAMES*, 97-112.

McCoy, J., Wardrip-Fruin, N., Treanor, M., Samuel, B., Reed, A. A., & Mateas, M. (2013). *Prom Week: Designing past the game/story dilemma.* Santa Cruz, CA: University of California Santa Cruz, Expressive Intelligence Studio.

Montfort, N. (2003). *Twisty Little Passages: An Approach to Interactive Fiction.* Cambridge, MA: The MIT Press.

Nelson, G. (2001). *The Inform Designer's Manual.* St. Charles, Illinois: The Interactive Fiction Library.

Rauch, J. (2006, November). *Sex, Lies, and Videogames*. Retrieved from The Atlantic.

Samuel, B., Lederle-Ensign, D., Treanor, M., Wardrip-Fruin, N., McCoy, J., Reed, A., & Mateas, M. (2016). Playing the Worlds of Prom Week. In M. Hatavara, M. Hyvärinen, M. Mäkelä, & F. Mäyrä, *Narrative Theory, Literature, and New Media: Narrative Minds and Virtual Worlds.* New York, NY: Routledge.

Sheldon, L. (2004). *Character Development and Storytelling for Games.* Boston, MA: Thomson Course Technology.

Short, E. (2008, December 2008). *Alabaster Open Beta*. Retrieved from Emily Short's Interactive Storytelling.

Siebert, A. (2014, September 8). *A Brief History of Interactive Fiction Games….* Retrieved from
    The Dire Multiverse: http://www.gamesomniverse.com/brief-history-interactive-fiction/

Stuart, K. (2011, January 11). *Interactive fiction in the ebook era.* Retrieved from The Guardian:
    https://www.theguardian.com/technology/gamesblog/2011/jan/11/interactive-fiction-
    ebooks-apple-kindle

Van Scoy, F. (2016). *Through the Looking Glass Game: Design and Implementation Documents.*