Instituto Tecnológico y de
Estudios Superiores de Monterrey

Campus Querétaro

Curse

The Prolog Text Adventure

Documentation

Programming Languages
August – December 2020

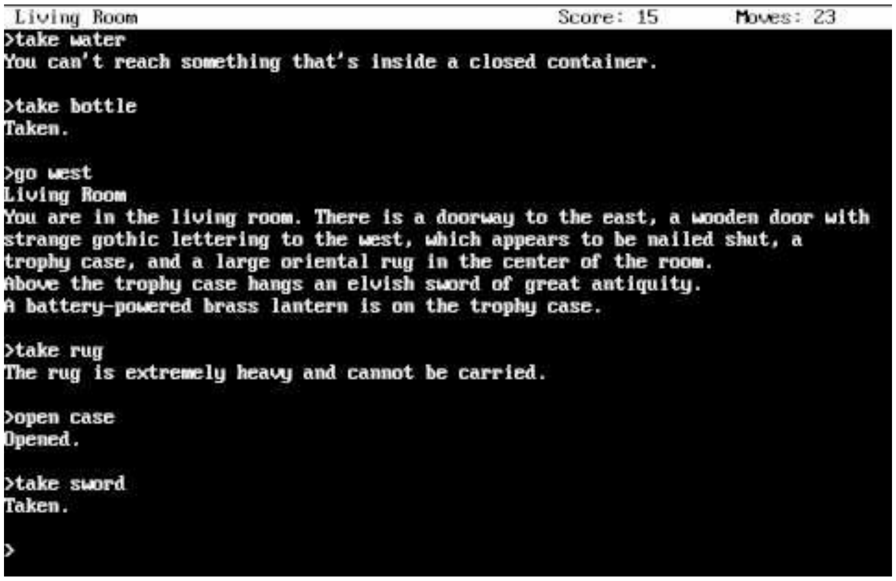
Eduardo González Melgoza
A01701446

Index

What Is A Text Adventure?

Text adventures, also referred to as interactive fictions, date back to the year 1976. They are a form of software in which the user is presented with a text-based description of various environments, situations and/or characters. These elements can be interacted with and influenced using simple text commands. Interactive fictions are also viewed as a form of video games, usually belonging to the adventure games or role-playing game (RPGs) genres, thanks to the nature of their functionality. In text adventure games, the player normally takes on the role of a main character in the narrative, interacting by first reading what the program describes, and then inputting a command congruent to what is outputted onto the screen.

Even though these games lack a use of a graphic interface, there is a physical dimension in which the game takes place. In-game descriptions include imagery of the player's surroundings, such as the current room, interactive items and other non-player characters (NPCs). The player can input commands such as "go north" or "take object" to move from one room to another or interact with the characters and objects in each room, prompting the program to give a feedback description of how the world changes as the player affects the environment. Some notable examples of famous text adventures include Colossal Cave Adventure (1976), the Zork series (1980, 1981, 1982) and Cypher. Figure 1 shows a screenshot taken from an online version of Zork I[1]. In it, the gameplay can be appreciated, the player's input is shown after the '>' and the game feedback and descriptions follow each given command.



```
 Living Room                                       Score: 15        Moves: 23
>take water
You can't reach something that's inside a closed container.

>take bottle
Taken.

>go west
Living Room
You are in the living room. There is a doorway to the east, a wooden door with
strange gothic lettering to the west, which appears to be nailed shut, a
trophy case, and a large oriental rug in the center of the room.
Above the trophy case hangs an elvish sword of great antiquity.
A battery-powered brass lantern is on the trophy case.

>take rug
The rug is extremely heavy and cannot be carried.

>open case
Opened.

>take sword
Taken.

>
```

Figure 1

[1] https://classicreload.com/zork-i.html

Objective

The objective of this project is to apply and expand on the acquired knowledge of the logic programming paradigm to design and code a functional text adventure. The game will be coded using the Prolog programming language.

Result

The project resulted in a text adventure game named Curse (the source file's name is curse.pl). Curse is a fantasy adventure RPG text adventure game with light horror elements. It was coded using the Prolog programming language and the logic programming paradigm. The player can advance through the game's plot by inputting some of the classic commands also present in other famous text adventures like the "look." command to look about the player's surroundings or "i." to check their inventory. The game has a concrete plot, an objective and an ending.

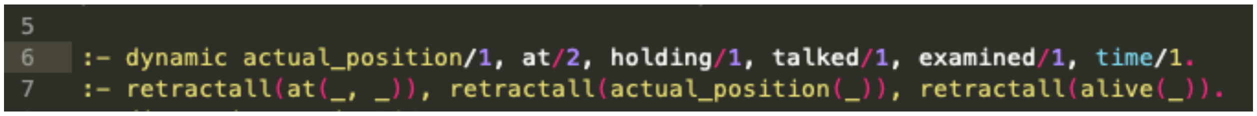
Functionality

Since Curse was programmed in Prolog, the functionality of the game is based on a series of rules and facts, as well as unification of different values to various variables. To understand the basic concepts of Prolog, a tutorial for beginners is available in the following link: http://www.doc.gold.ac.uk/~mas02gw/prolog_tutorial/prologpages/. This tutorial will explain some central concepts like facts, arguments, unification and recursion, as well as interactive examples that will help get a better understanding of them.

In the source code of Curse, there are some techniques and predicate properties being used that are not explained in the aforementioned tutorial. These techniques work as follows:

dynamic/1

`dynamic/1` is a predicate property that indicates that the definition of the indicated predicate may change during execution. To cycle between the various possible definitions of a dynamic predicate, the use of the `assert/1` and `retract/1` terms is needed. An example of `dynamic/1` can be observed in line number six of the curse.pl source file.



```
5
6  :- dynamic actual_position/1, at/2, holding/1, talked/1, examined/1, time/1.
7  :- retractall(at(_, _)), retractall(actual_position(_)), retractall(alive(_)).
```

Figure 2

Figure 2 shows the sixth line of Curse's source code. The predicate property `dynamic/1` is being used to modify the predicates `actual_position/1`, `at/2`, `holding/1`, `talked/1`, `examined/1` and `time/1`.

assert/1

`assert/1` is a meta-predicate in Prolog that adds the argument it receives to the Prolog knowledge base. It is useful when a new rule or fact needs to be added to the knowledge base during the execution of a Prolog program. It is important to note that it can only be applied to dynamic predicates. It has two variants: `asserta/1` and `assertz/1`. The former pushes its argument to the beginning of the knowledge base, while the latter adds it at the end. Though deprecated, `assert/1` functions the same way as `assertz/1`. A use case example for `assert/1` is shown below:

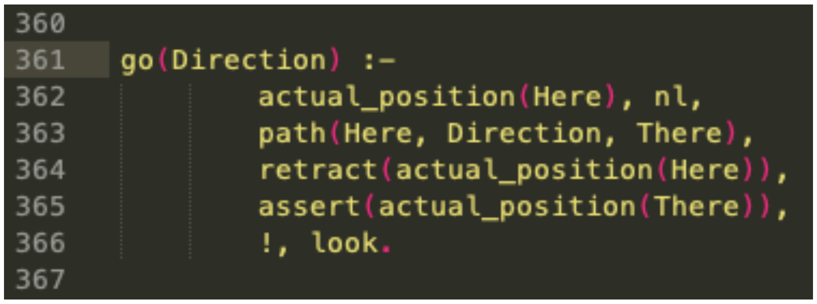
```
?- assert(rich(mary)).
true.

?- rich(mary).
true.

?- assert((happy(X) :-
      rich(X),
      healthy(X))).

?- assert(healthy(mary)).
true.

?- happy(X).
X = mary
```

retract/1

`retract/1` is a meta-predicate in Prolog that removes the argument it receives from the Prolog knowledge base. It works in conjunction with `assert/1`, making it important to note that it can only be applied to dynamic predicates as well. A use case example for `retract/1` is shown below:

```
?- assert(likes(mary, pizza)).
true.

?- likes(mary, pizza).
true.

?- retract(likes(mary, pizza)).
true.

?- likes(mary, pizza).
false.
```

`assert/1` and `retract/1` are both used various time throughout the source file of Curse. Figure 3 shows an example of a situation in which they are used.



```

go(Direction) :-
        actual_position(Here), nl,
        path(Here, Direction, There),
        retract(actual_position(Here)),
        assert(actual_position(There)),
        !, look.

```

Figure 3

Here, `assert/1` and `retract/1` work together to change the players current position, retracting is previous position fact from the knowledge base and asserting the new one.

retractall/1

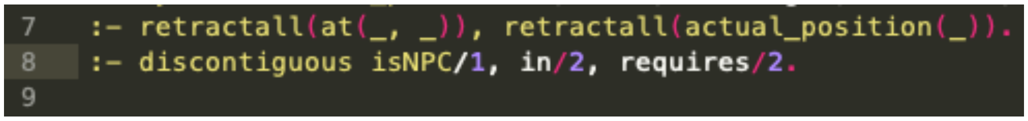
`retractall/1` functions in the same manner as `retract/1`, except it eliminates from the knowledge base all of the predicates that match its argument. Figure 4 shows the usage of `retractall/1` in the source code of Curse. In this instance, it is being used to clean the knowledge base before assigning actual definitions to some of the dynamic predicates.

discontiguous/1

`discontiguous/1` is a meta-predicate that indicates that the defining clauses of the predicate that it receives as argument, are not restricted to be consecutive, whereas they can appear anywhere in the source file. This is an example of how to use `discontiguous/1`:

```
:- discontiguous dog/1.

dog(spike).
cat(mittens).
cat(tabby).
dog(spot).
```

In this example, the different predicates for dog are not defined one after the other. Without the first line of the example, this would output a warning. In the source file for Curse, `discontiguous/1` is used as follows:



```
:- retractall(at(_, _)), retractall(actual_position(_)).
:- discontiguous isNPC/1, in/2, requires/2.
```

Figure 4

Figure 4 shows how `discontiguous/1` is being applied to the `isNPC/1`, `in/1` and `requires/1` predicates.

How to Play

Output Format

Curse's output will always be formatted in a specific way. To explain its format, the game's first output will be used as an example below:

```
-------------------------------------
Enter commands using standard Prolog syntax.
start.                     to start the game.
n.  s.  e.  w.             to move in given direction.
take(Object).              to pick up an object.
talk(Character).           to talk to other characters.
```

```
i.                       to check your inventory.
look.                    to look around you again.
controls.                to show controls again.
halt.                    to end game and quit.
-------------------------------------

The year is 1310. You are the town of Eadburgh's local priest.
Lately, people have been disappearing, there have been rumors of a
vampiric curse menacing the town. You can trust no one.

The time is late at night when you hear someone knocking at your door...
asking for help. As the priest, you cannot deny help to townspeople.
You open the door to find a cloaked figure who suddenly attacks you,
too fast to even try to defend yourself!

While lying on the floor, you notice the cloaked figure stepping inside
the church
closing the doors behind it.

You black out...

You wake, lying on the floor before your church. You feel an acute pain
to the neck. You touch it with your fingers to discover you have been
bitten! "I've been cursed with vampirism", you figure, "I've got to
find a way to revert the curse, and kill whatever thing cursed me."
-------------------------------------

You are in the church plaza.
You can hear violent noises coming from inside the church.

- To the north is the church.
- To the south is the main plaza.
- To the west is your house.
-------------------------------------
```

The various parts in which the output is divided are the following:

1. Controls
2. Context
3. Current location
4. Nearby locations

Controls

The Controls Section is automatically shown only upon inputting the `start` command; it will be the first section to be printed on the console. Afterwards, the player can input the command `controls` whenever they wish to display this section again.

Context

The Context Section is only accessible through the `start` command. It will give the player information about how the story begins, and how the main character got to the point of the story in which it begins. After the game starts, there should be no need to use the `start` command again, which is why this section can only be accessed to once in each playthrough to allow for the correct functioning of the game.

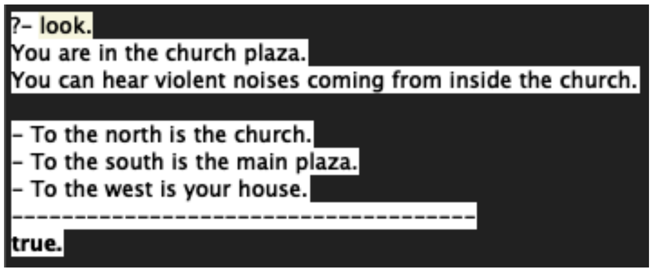
Current Location

This section, along with the Nearby Locations section, will be the ones that the player will see most often during gameplay. Whenever the player moves to a new location in the game's world, the Current Location section will be displayed before anything else. It will display a short description of the location in which the player currently is.

Some locations will have either objects laying around or NPCs with which the player can interact in different ways. These objects or characters will be displayed one after the other following the location's description.

Whenever the player wishes to consult their current location again, this can be done by using the `look` command. The current location will be displayed again, followed by the contents of the Nearby Locations section.

The following image shoes an example of how the `look` command is used in game:



```
?- look.
You are in the church plaza.
You can hear violent noises coming from inside the church.

- To the north is the church.
- To the south is the main plaza.
- To the west is your house.
-----------------------------------
true.
```

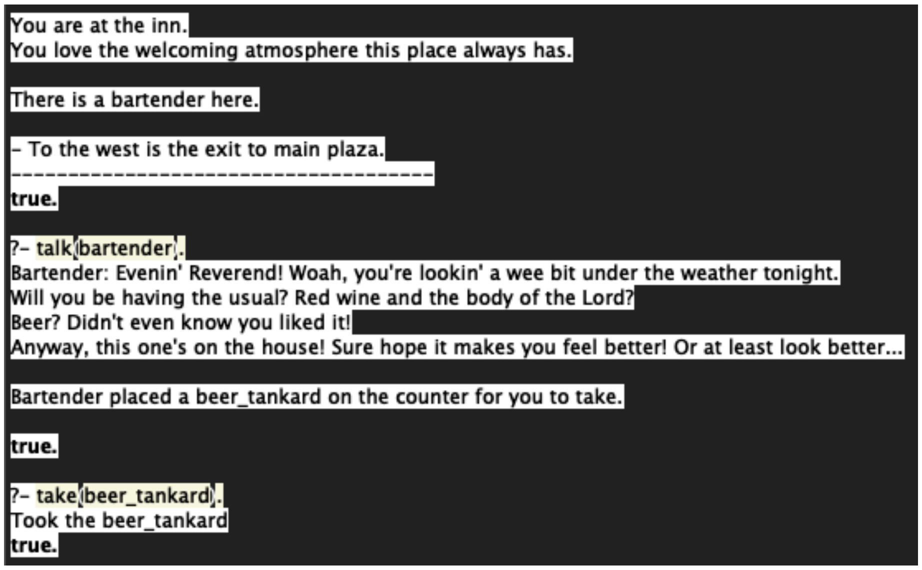
Figure 5

Interacting With NPCs and Objects

To interact with non-player characters, the `talk` command is used, followed by the name of the character to interact with in between parentheses. After talking with them, NPCs will drop items that the player can take using the `take` command, followed by the name of the object to interact with between parentheses. Objects that are already laying in a certain location can also be interacted with using the `take` command.

There are certain items that can only be taken by trading them for a specific object. To interact with tradable items, use the `take` command followed by the name of the object to interact with between parentheses. The trade will only proceed if the specific required object is found in the player's inventory.

Figure 6 displays an example of how the `talk` and `take` commands are used in the game:



```
You are at the inn.
You love the welcoming atmosphere this place always has.

There is a bartender here.

- To the west is the exit to main plaza.
-----------------------------------
true.

?- talk(bartender).
Bartender: Evenin' Reverend! Woah, you're lookin' a wee bit under the weather tonight.
Will you be having the usual? Red wine and the body of the Lord?
Beer? Didn't even know you liked it!
Anyway, this one's on the house! Sure hope it makes you feel better! Or at least look better...

Bartender placed a beer_tankard on the counter for you to take.

true.

?- take(beer_tankard).
Took the beer_tankard
true.
```

Figure 6

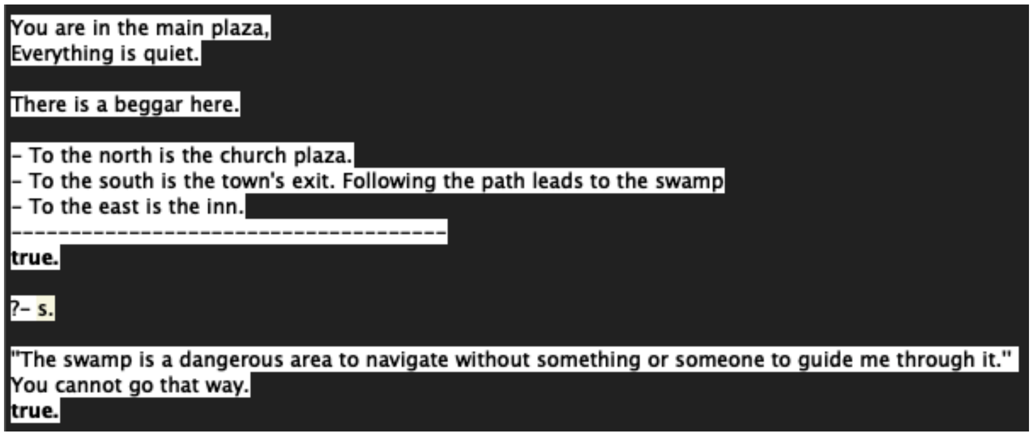
Please note that objects offered to the player by NPCs will not appear as items present in a given location if the player has not spoken to the NPC beforehand. In the same manner, if an NPC offers an item to the player, and before taking it, the player executes the `look` command, the offered item will be displayed as one of the items found in the current location.

Nearby Locations

The Nearby Locations section will always be displayed after the Current Location section. Due to this, the player can access this information by running the `look` command whenever they wish. This section displays information on the various locations that can be accessed to from the current location, by moving in each given direction. If there is nowhere to go by moving in a certain direction of the four possible ones (north, south, east and west), this direction will not be displayed in the possible movable directions. Also, if the player attempts to move in a non-existent direction, a feedback message informing this will be shown.

During the events of Curse, some locations exist but may not be accessible until a certain condition is met or a certain item has been acquired. These locations will be effectively shown using the `look` command, however, if the condition for moving in that direction is not met yet, a message will inform this.

The following image shows the event in which the player attempts to move into a location which is restricted since the condition to pass has not yet been met:

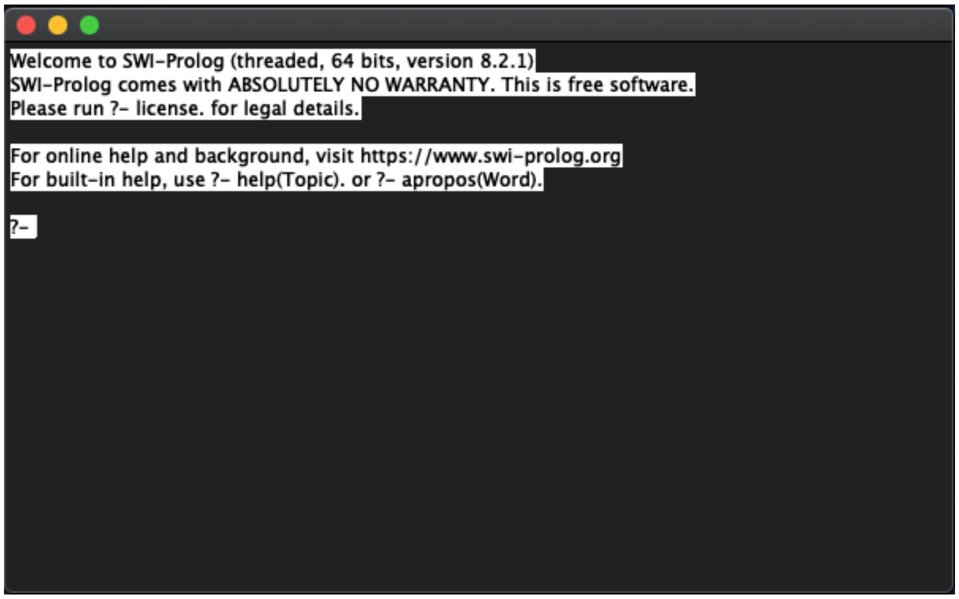


```
You are in the main plaza,
Everything is quiet.

There is a beggar here.

- To the north is the church plaza.
- To the south is the town's exit. Following the path leads to the swamp
- To the east is the inn.
------------------------------------
true.

?- s.

"The swamp is a dangerous area to navigate without something or someone to guide me through it."
You cannot go that way.
true.
```

Figure 7

Setup

To be able to play Curse, it is recommended to run the code using the SWI-Prolog IDE. This is available to be downloaded from the following link: https://www.swi-prolog.org/download/stable. Please note that there exists an online version of SWI-Prolog, however, this version does not allow for the adequate functionality to work properly[2]. Once SWI-Prolog has been installed, running it should open a window like the one shown in figure 8 (screenshot taken on macOS).



```
Welcome to SWI-Prolog (threaded, 64 bits, version 8.2.1)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?-
```

Figure 8

Due to the nature of the Prolog language, the game's code is a series of predicates that have to be consulted and queried in order to play through it. To load the game's source code file into SWI-Prolog and start playing, run the following command, using the absolute path of where the source file is saved:

```
?- consult('<file/absolute/path.pl>').
```

Example:

```
?- consult('/Users/lalogonzalez/repos/Curse/curse.pl').
```

World Map

[2] The game works by asserting and retracting the different predicate definitions, and this does not work properly in the online IDE.

.

Walkthrough (and all possible paths)

Beginning

To begin to play Curse, the aforementioned setup must be completed prior to beginning. Once the setup is done, enter the following command to start the game:

```
     ?- start.
```

After the command has been executed (which in reality is a query), the game controls will be explained, followed by the initial game scenario description. The following output should be displayed:

```
-------------------------------------
Enter commands using standard Prolog syntax.
start.               to start the game.
n.  s.  e.  w.       to move in given direction.
take(Object).        to pick up an object.
talk(Character).     to talk to other characters.
i.                   to check your inventory.
look.                to look around you again.
controls.            to show controls again.
halt.                to end game and quit.
-------------------------------------

The year is 1310. You are the town of Eadburgh's local priest.
Lately, people have been disappearing, there have been rumors of a
vampiric curse menacing the town. You can trust no one.

The time is late at night when you hear someone knocking at your door...
asking for help. As the priest, you cannot deny help to townspeople.
You open the door to find a cloaked figure who suddenly attacks you,
too fast to even try to defend yourself!

While lying on the floor, you notice the cloaked figure stepping inside
the church
closing the doors behind it.

You black out...
You wake, lying on the floor before your church. You feel an acute pain
to the neck. You touch it with your fingers to discover you have been
bitten! "I've been cursed with vampirism", you figure, "I've got to
find a way to revert the curse, and kill whatever thing cursed me."
-------------------------------------

You are in the church plaza.
You can hear violent noises coming from inside the church.

- To the north is the church.
- To the south is the main plaza.
- To the west is your house.
-------------------------------------
```

Church Plaza

```
You are in the church plaza.
You can hear violent noises coming from inside the church.

- To the north is the church.
- To the south is the main plaza.
- To the west is your house.
-------------------------------------
```

The player starts the game off in the Church Plaza. From this area, there are three possible paths to take, two of which are not yet accessible to the player. To the north is the entrance to the Church, however, if the player attempts to move in this direction, the following feedback message will be displayed:

```
Are you crazy? Attempting to enter a holy place with a vampiric curse
laid upon yourself.
You would burn whole!
You cannot go that way.
```

In addition, the player will fail to move in direction to the north. To make it possible for the player to move north into the Church, the player must advance through the game's events to find the `doll` item. Another possible direction is west, into the player character's house. This direction is also inaccessible for the time being, attempting to move west from the Church Plaza will display the following message:

```
"It's locked. I left the key in my study in the left wing of the church.
I need to find a way to get inside..."
You cannot go that way.
```

To be able to move west from this location, the player first needs to retrieve the `house_key` item from the Study. The player will fail to move west, leaving south as the only possible direction to move in.

Main Plaza

```
You are in the main plaza,
Everything is quiet.

There is a beggar here.

- To the north is the church plaza.
- To the south is the town's exit. Following the path leads to the swamp
- To the east is the inn.
-------------------------------------
```

South to the Church Plaza is Eadburgh's Main Plaza. From this location, the possible paths to take are north, back into the Church Plaza, east into the town's local Inn and exit south into the Swamp. At this point the Swamp is inaccessible, attempting to move in this direction will display the following message:

```
"The swamp is a dangerous area to navigate without something or someone
to guide me through it."
You cannot go that way.
```

In the Main Plaza, the player will find the first NPC, a beggar. Like with every other non-player character, the player can interact with the beggar by using the `talk` command. Doing this will display the following dialogue:

```
Beggar: What's the matter Reverend? Can't go inside your church?
I'll tell you what. I heard an old hag lives in the swamp south to
Eadburgh.
They call her Old Jezabelle. She might be able to help you with your...
situation.
Oh, but you won't get there so easily. You will need a compass to traverse
those wetlands.
Luckily for you, I've got one right here. Sure, you can have it...
```

```
What? You thought I would give it away so easily? Gwahaha
FOOD? CLOTHES? Do I look like a charity case to you?
BEER IS WHAT I WANT! Go get me some delicious beer from the Inn and I'll
give you my compass.

The beggar offered a compass for you to take in exchange for some beer.
```

After talking with the beggar, he will drop a compass in the Main Plaza. If the player uses the `look` command, the `compass` should appear as one of the objects/people in the current location. The player cannot take the `compass` yet, that is until they find some beer to trade it for the `compass`.

Since the only possible direction that the player can move to other than going back to the Church plaza is east into the Inn, this is the next area that should be visited. Please note that there will be no conflicts if the player visits the Inn even before talking to the beggar.

(Once the Beer Tankard is Obtained)

After the player has found a `beer_tankard` in the Inn and is holding it in their inventory, they can now use the `take` command on the compass to trade it for the `beer_tankard`, doing so will display the following dialogue and feedback:

```
Beggar: Took you long enough. Yes yes take it, just give me my beer.

Handed the beer_tankard to the beggar.
Took the compass.
```

Taking the `compass` while holding the `beer_tankard` will remove the `beer_tankard` and add the `compass` to the player's inventory. The `compass` item will allow the player to move south from the Main Plaza into the Swamp area.

Eadburgh Inn

```
You are at the Inn,
You love the welcoming atmosphere this place always has.

There is a bartender here.

- To the west is the exit to main plaza.
```

```
-------------------------------------
```

The Eadburgh Inn is both a resting place for those who have got nowhere to stay, but also the town's local bar. This is where the player will encounter the second NPC, the bartender. The player can use the `talk` command on the bartender to display the following dialogue:

```
Bartender: Evenin' Reverend! Woah, you're lookin' a wee bit under the
weather tonight.
Will you be havin' the usual? Red wine and the body of the Lord?
Beer? Didn't even know you liked that!
Anyway, this one's on the house! Sure hope it makes you feel better!
Or at least look better…

Bartender placed a beer_tankard On the counter for you to take.
```

After talking with the bartender, he will leave a `beer_tankard` on the counter for the player to take. It is important to notice that since this item has two words in its name, they should be joined with an '_' (underscore) to make it possible for the `take` command to work properly. The player is free to take the `beer_tankard` without another item to trade for it. Doing so will display the following feedback:

```
Took the beer_tankard.
```

After talking the tankard, there is nothing else to possibly do in the Inn, leaving the player with no other option than to go back into the Main Plaza.

Swamp

```
Used the compass to move through the swamp.
You are at the Inn.
The moist air is think here, you find it hard to breathe.

- To the north is the path back to Eadburgh's main plaza.
- To the east you see an old cabin, almost consumed by the swamp's
vegetation.
- To the west is a Bald Cypress tree family.
-------------------------------------
```

The swamp is an area that functions as a connection between the Bald Cypress tree family and the old cabin. There are no items nor NPCs here, so the player may decide if they want to move to the tree family or the old cabin first, however, the recommended direction to move into first is the old cabin.

Old Cabin

```
You are inside the cabin.
The walls are covered with shelves,
topped with all sorts of weird stuff like plants and critters.

There is a witch here.

- To the west is the exit back to the swamp.
-------------------------------------
```

The old cabin is home to Old Jezabelle, the old witch that the beggar tells the player about. She is the NPC that will help the player get into the Church. The player is to talk to her, doing so will display the following dialogue:

```
Old Jezabelle: I'm a witch, but that you might have already figured, old
cabin in the swamp and all heheh.
Cursed, huh? Heheheh, how careless of you! So, you want to know how to
get rid of that curse, yes?
Heheheh...
Every now and then, some pesky traveler comes to Old Jezabelle, looking
for something great she can do for them.
Some come seeking revenge on their most hated ones.
Some others come searching for unnatural love, a treacherous bond if you
ask me.
What now? A priest, coming to such a godless being, to recover his
holiness...
Heheheh, listen. Old Jezabelle will help you, but just because she enjoys
the irony of the situation.
Old Jezabelle cannot lift the curse, but she can make a doll that will...
shall we say, camouflage your current state from the eyes of your God.
While holding the doll, you should be able to go inside the church and
slay the thing that laid that curse upon you.
Yes, doing that should revert the curse permanently.
Funny coincidence heheheh, Old Jezabelle was just finishing a doll like
the one she's telling you about.
The only thing missing is a piece of bald cypress tree bark. Get me some
of that to complete the doll, and you are free to take it.
```

```
Old Jezabelle dropped the doll for you to take once you get the missing
piece.
```

Bibliography

Lu, J. (N. D.) Prolog A Tutorial Introduction. Computer Science Department. Bucknell University. Accessed on: October 2020. Available at: http://academicos.azc.uam.mx/cbr/Cursos/UEA_12p_Log/PrologIntro_eng_good.pdf

Burnett, I. (2015) The Text Adventure: Relic of Gaming History, or Timeless Medium? The Artifice. Accessed on: October 2020. Available at: https://the-artifice.com/text-adventure-gaming-history/

Ireson-Paine, J. (2020) Why Use Prolog? Joselyn Ireson-Paine Consultancy. Accessed on: October 2020. Available at: http://www.j-paine.org/why_prolog.html

N. A. (2020) Predicate consult/1. Prolog Documentation. Accessed on: October 2020. Available at: https://www.swi-prolog.org/pldoc/man?predicate=consult/1

N. A. (2020) Predicate dynamic/1. Prolog Documentation. Accessed on: October 2020. Available at: https://www.swi-prolog.org/pldoc/man?predicate=dynamic/1

N. A. (N. D.) Zork I. Classic Reload. Accessed on: October 2020. Available at: https://classicreload.com/zork-i.html