



Interactive Fiction and Text Generation

Daphne Ippolito
Chris Callison-Burch

<http://interactive-fiction-class.org>

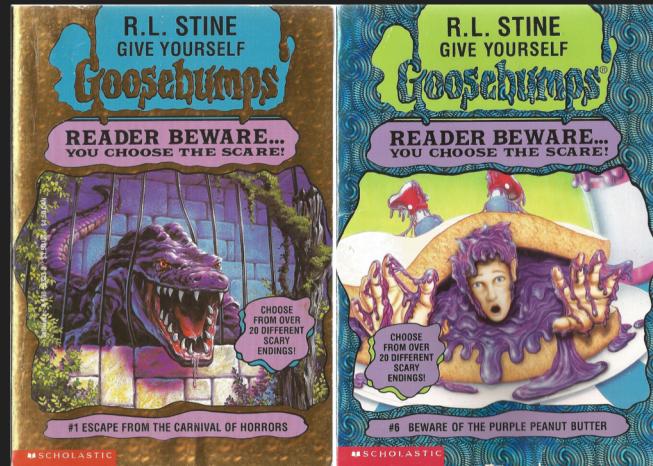
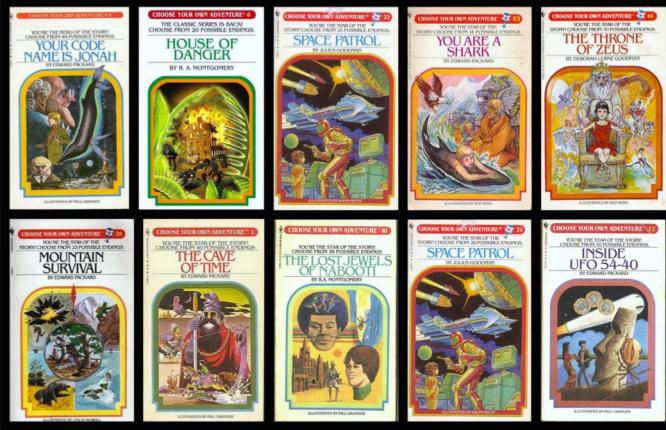
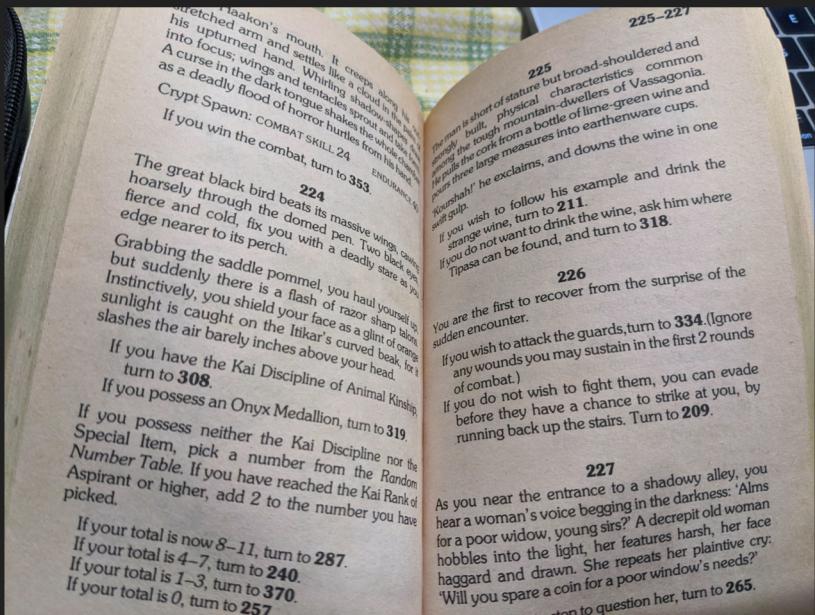
Today's Class

- Text Adventure Games
- NLP Introduction
- Course overview
- Activity

Text Adventure Games

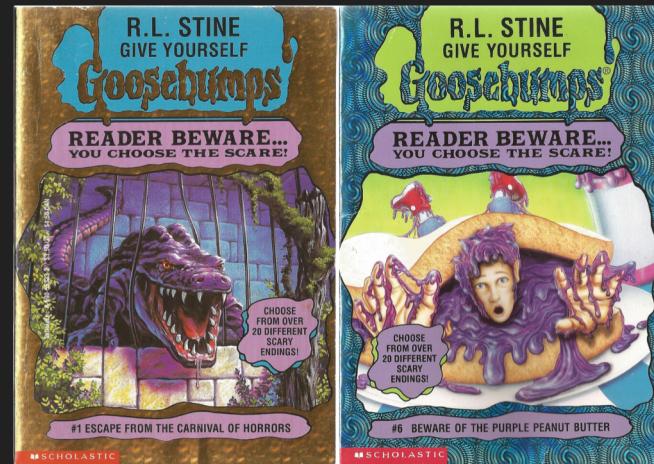
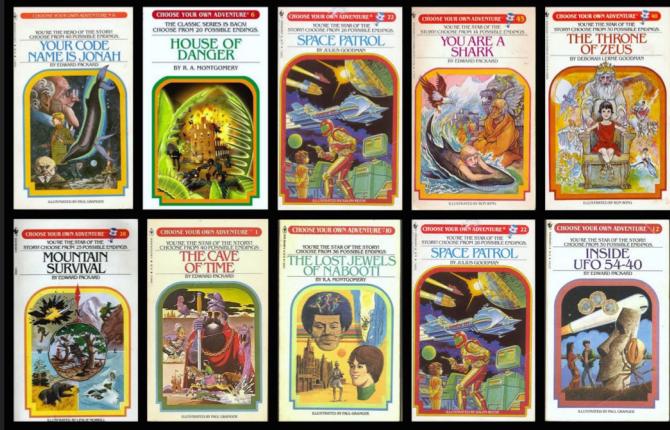
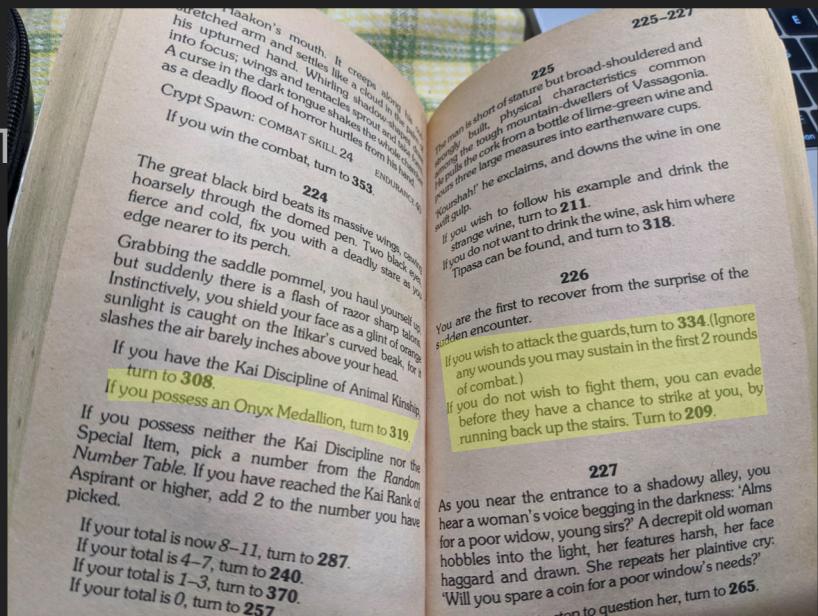
Paper Interactive Fiction

In the 1970s, “Choose your own adventure” books grew in popularity.



Paper Interactive Fiction

In the 1970s, “Choose your own adventure” books grew in popularity.



CHOOSE YOUR OWN ADVENTURE™ 5

YOU'RE THE STAR OF THE STORY!
CHOOSE FROM 36 POSSIBLE ENDINGS

THE MYSTERY OF CHIMNEY ROCK

BY EDWARD PACKARD



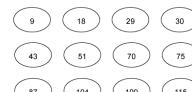
Special
Book
Fair
Edition

ILLUSTRATED BY PAUL GRANGER

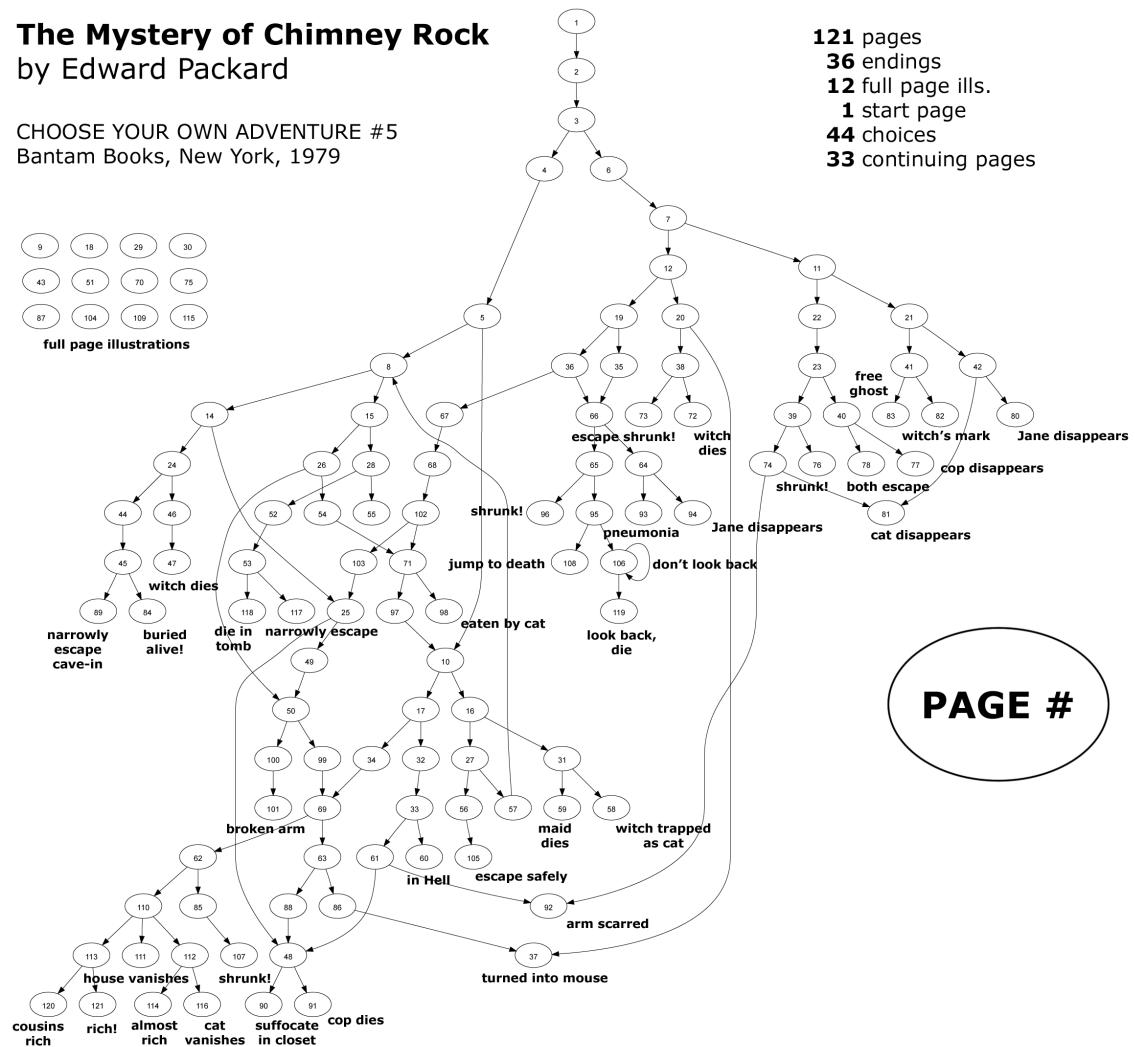
The Mystery of Chimney Rock

by Edward Packard

CHOOSE YOUR OWN ADVENTURE #5
Bantam Books, New York, 1979



full page illustrations

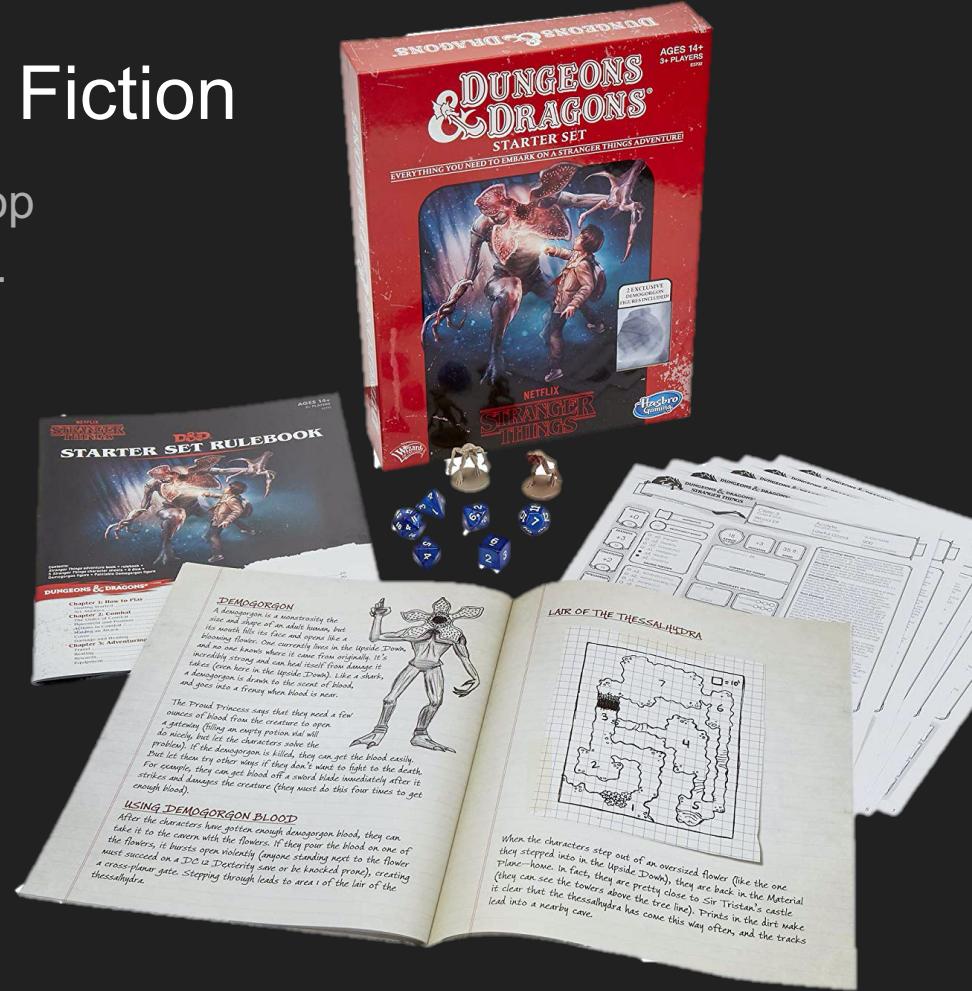


121 pages
36 endings
12 full page illus.
1 start page
44 choices
33 continuing pages

PAGE #

Paper & Pencil Interactive Fiction

Dungeons & Dragons is a fantasy tabletop role playing game first published in 1974.



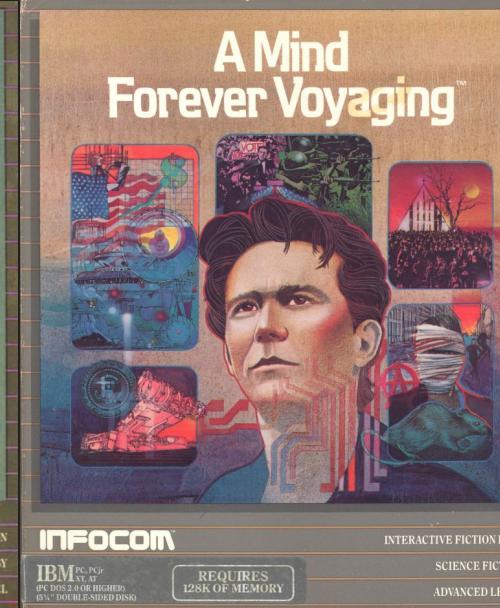
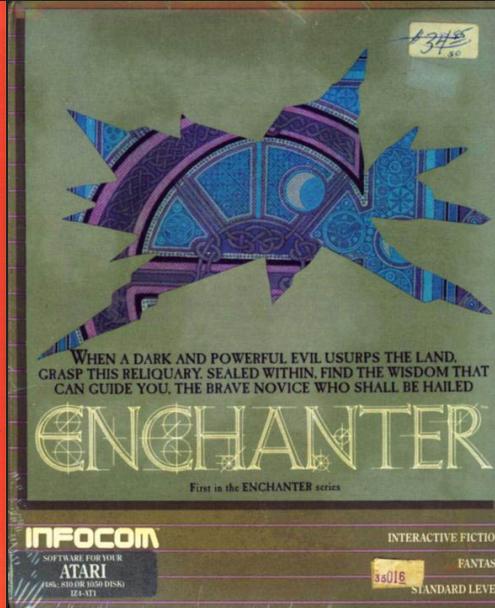
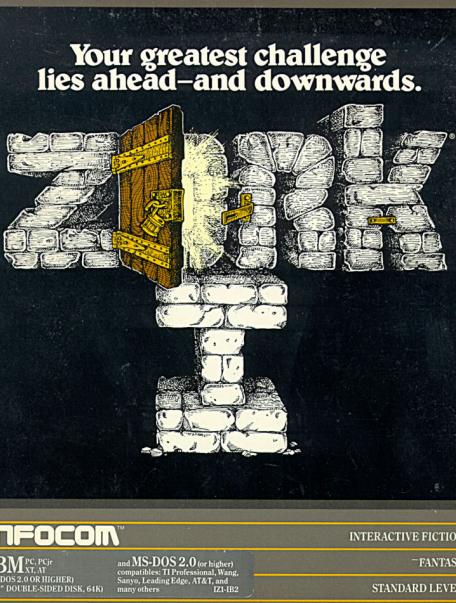
DUNGEONS AND DRAGON

D&D is an open-ended game in which the players assume the roles of characters in a story and can have them attempt any action they want. The game is controlled by a dungeon master, who uses tables, dice, and personal judgment to decide on the effect of a character's efforts.

The players say what their characters do within the world of the campaign. Dice are rolled to help determine the outcome of combat and other encounters. Over the course of many adventures, the members of a party advance in level and become more powerful-and the dungeon master devises new challenges for them.

Computer Games Before Graphics

Interactive Fiction/Text Adventure Games were the best-selling computer games of the 1980s. A company called Infocom created the most popular titles.



On the table is an elongated brown sack, smelling of hot peppers.

A clear glass bottle is here.

The glass bottle contains:

A quantity of water.

>w

You are in the living room. There is a door to the east. To the west is a wooden door with strange gothic lettering, which appears to be nailed shut.

In the center of the room is a large oriental rug.

There is a trophy case here.

On hooks above the mantelpiece hangs an elvish sword of great antiquity.

A battery-powered brass lantern is on the trophy case.

There is an issue of US NEWS & DUNGEON REPORT dated 28-JUL-80 here.

>get sword

Taken.

>break egg with sword

You rather indelicate handling of the egg has caused it some damage.

The egg is now open.

There is a golden clockwork canary nestled in the egg. It seems to have recently had a bad experience. The mountings for its jewel-like eyes are empty, and its silver beak is crumpled. Through a cracked crystal window below its left wing you can see the remains of intricate machinery. It is not clear what result winding it would have, as the mainspring appears sprung.

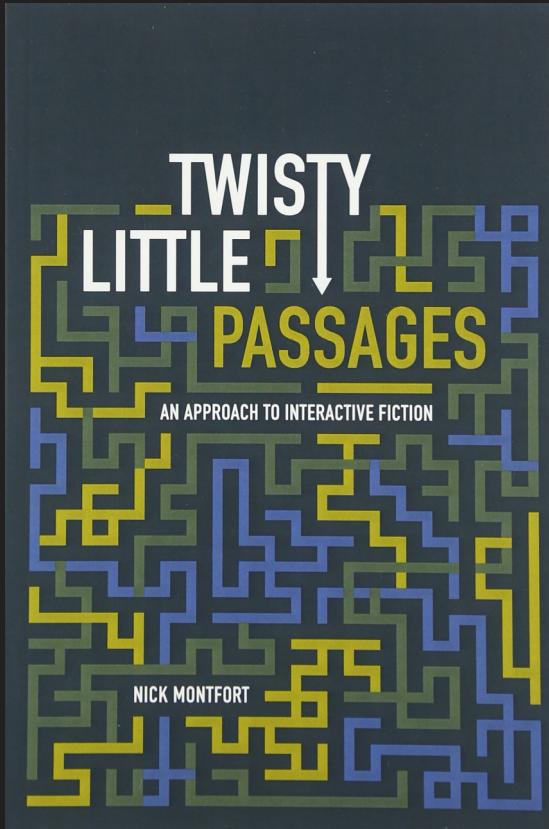
>_

Lance Micklus

Lance Micklus, Inc.

You had to appeal to their sense of
overcoming the odds and figuring things out.

Components of Interactive Fiction Games



- The **parser**, which is the component that analyzes natural language input in an interactive fiction work.
- The **world model**, which is setting of an interactive fiction work.

Natural Language Understanding

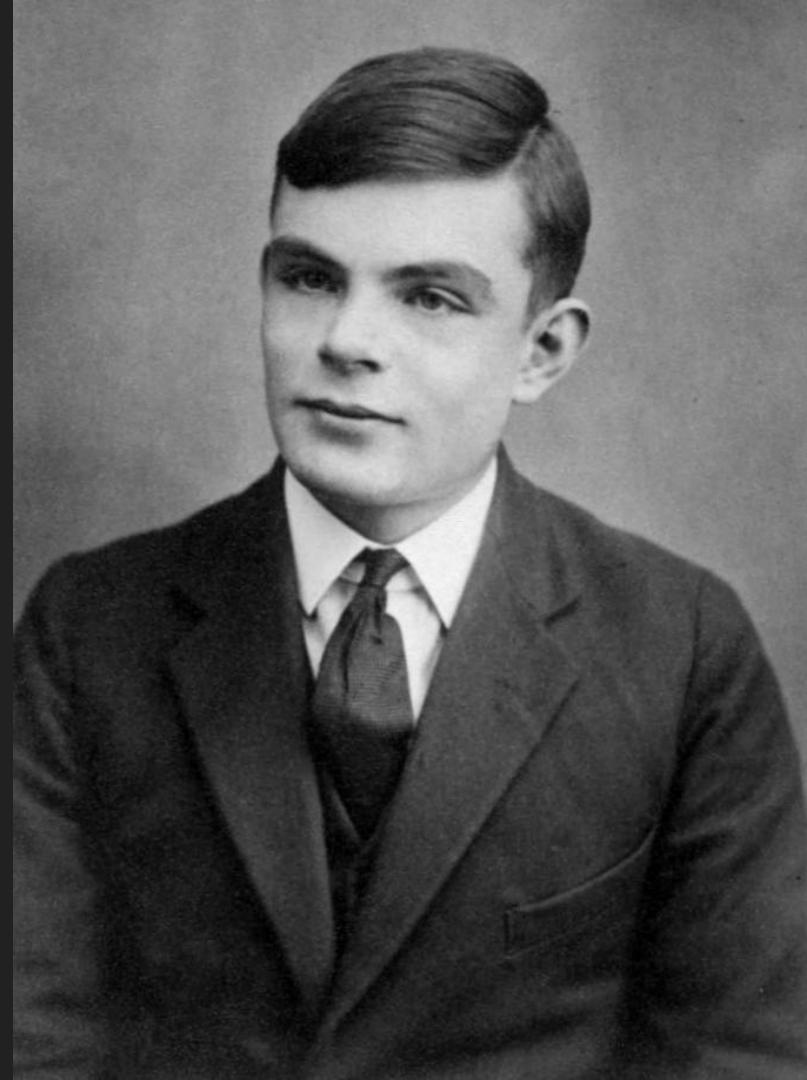
Will it rain tomorrow? Set an alarm for 6:30a.m. Play music from They Might Be Giants. How many pints are in a gallon? Add blueberries to my shopping list. When did California become a state? When is the Winter solstice? What is the weather like in Hawaii this weekend? Turn off the lights. Open the pod bay doors.



Turing Test

The idea of computer conversation was developed along with the general-purpose purpose digital compute. Turing presented it in his 1950 paper “Computing Machinery and Intelligence”. Which introduced the Turing Test. The question “Can machines think?” could then be replaced by one that was easier to answer: “Are there imaginable digital computers which would do well in the imitation game?”

Turing highlighted lighted the importance of conversational ability to our perception of intelligence, at a time when computers almost exclusively processed numbers rather than language.



Turing Test

A human evaluator would judge text-based conversations between a human and a machine designed to generate human-like responses. If the evaluator cannot reliably tell the machine from the human, the machine is said to have passed the test. The test results do not depend on the ability to give correct answers to questions, only how closely one's answers resemble those a human would give.



Early Natural Language Understanding

Welcome to

```
EEEEEE LL      IIII  ZZZZZZ  AAAAAA
EE      LL      II      ZZ  AA  AA
EEEEEE LL      II      ZZZ  AAAAAAAA
EE      LL      II      ZZ  AA  AA
EEEEEE LLLLLL IIII  ZZZZZZ  AA  AA
```

Eliza is a mock Rogerian psychotherapist.

The original program was described by Joseph Weizenbaum in 1966.

This implementation by Norbert Landsteiner 2005.

ELIZA: Is something troubling you ?

YOU: Men are all alike.

ELIZA: What is the connection, do you suppose ?

YOU: They're always bugging us about something or other.

ELIZA: Can you think of a specific example ?

YOU: Well, my boyfriend made me come here.

ELIZA: Is it important to you that your boyfriend made you come here ?

YOU: He says I'm depressed much of the time.

ELIZA: I am sorry to hear that you are depressed.

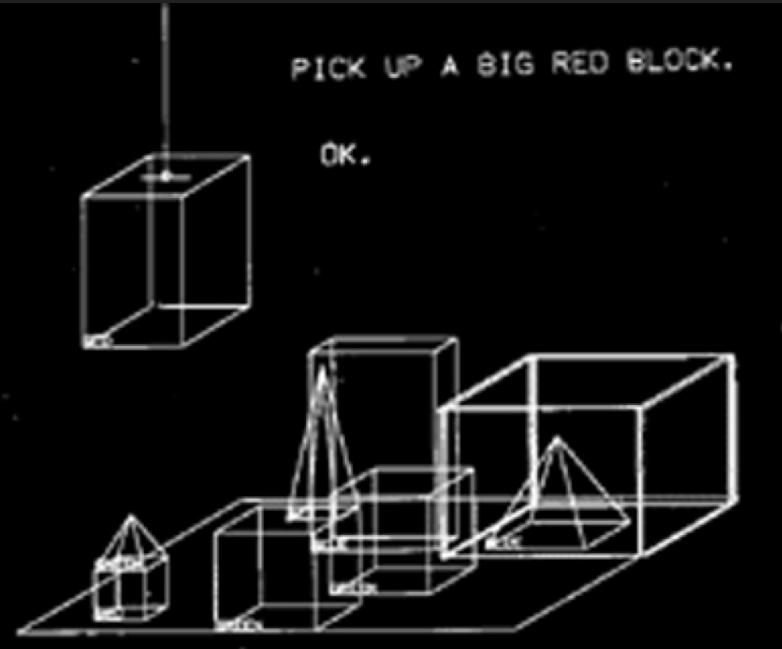
YOU: It's true. I am unhappy.

ELIZA: Can you explain what made you unhappy ?

YOU: ■

PICK UP A BIG RED BLOCK.

OK.



ELIZA

ELIZA is a language analysis program that can run scripts to have different sorts of conversations; the most famous script was DOCTOR, which was “designed to permit it to play (or parody) the role of a Rogerian psychotherapist engaged in an interview with a patient”.

ELIZA was the first computer conversationalist, the first chatbot. ELIZA searched for keywords in input. If some were found, the program would transform the input according to a rule and print out the transformed sentence. If none were found, it would provide a default (usually noncommittal) output like “What makes you say that?”

SHRDLU

Terry Winograd's SHRDLU had a dialog with the user via teletype and displayed an image of a tabletop with blocks on it. It could respond to natural language commands in real time. It would move the simulated blocks around if asked to; it could also answer questions about them.

SHRDLU used its knowledge of the simulated world to help it parse input. It represented its knowledge as procedures. SHRDLU had several different components. There was a module for generating replies to the user,, and one for simulating the blocks world.

Simulated World

What's the difference between a conversational agent and a work of interactive fiction? **The simulated world.**

It represents the physical environment, and things like

- Settings or locations
- Physical objects in each setting
- The player's character
- Non-player characters

It also represents and simulates the physical laws of the environment.

Locations

You are at a complex junction. A low hands and knees passage from the north joins a higher crawl from the east to make a walking passage going west. There is also a large room above. The air is damp here.

A location in *Colossal Cave Adventure* by Will Crowther (1975)

Colossal Cave Adventure

YOU ARE STANDING AT THE END OF A ROAD BEFORE A SMALL BRICK BUILDING. AROUND YOU IS A FOREST. A SMALL STREAM FLOWS OUT OF THE BUILDING AND DOWN A GULLY.

go south

YOU ARE IN A VALLEY IN THE FOREST BESIDE A STREAM TUMBLING ALONG A ROCKY BED.

Navigation in a text-based world

Cardinal Directions: Go North/South/East/West/
Northwest/Northeast/Southwest/Southeast

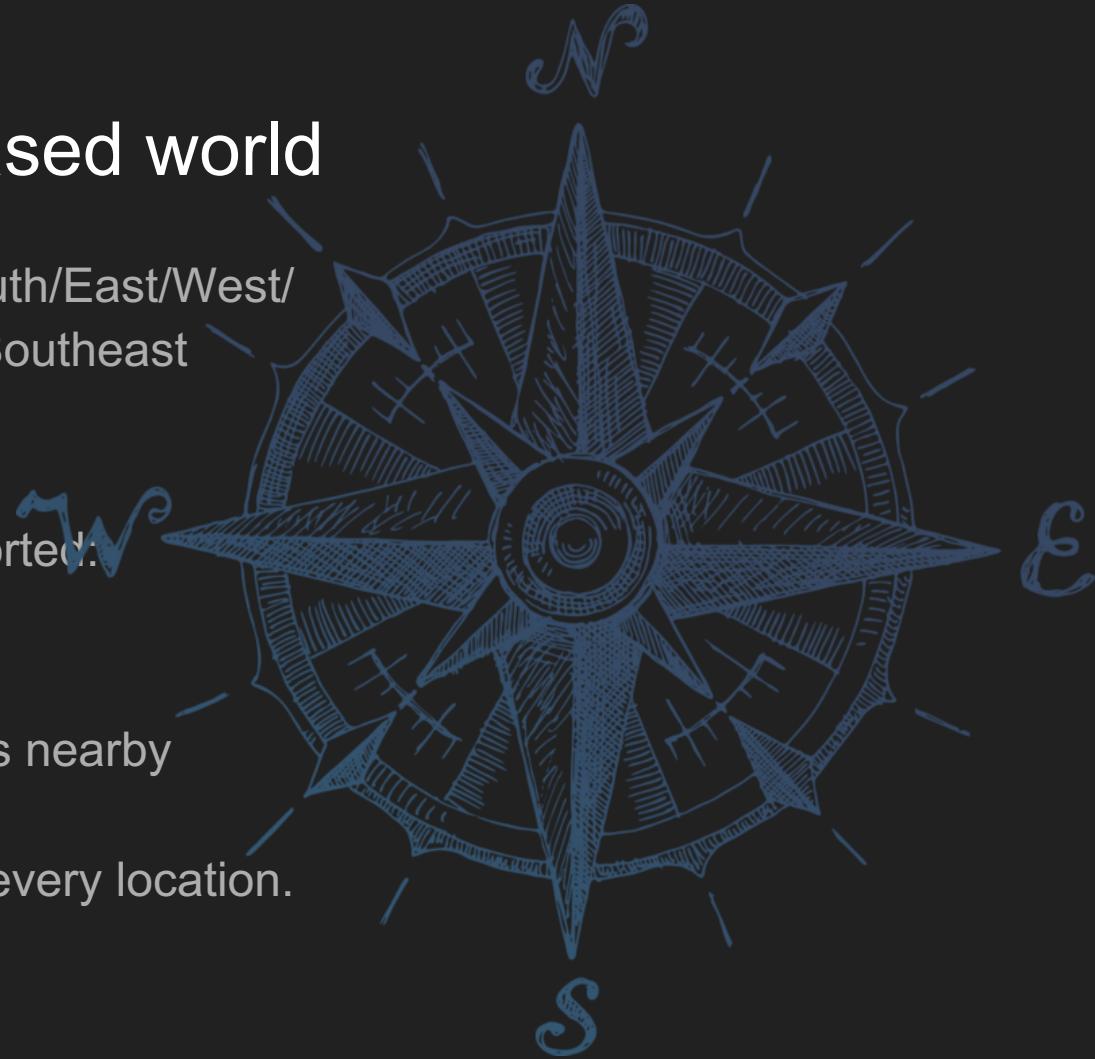
Also: Go Up, Down, In and Out

One letter commands were supported:

N/E/S/W/NE/SE/NW/SW

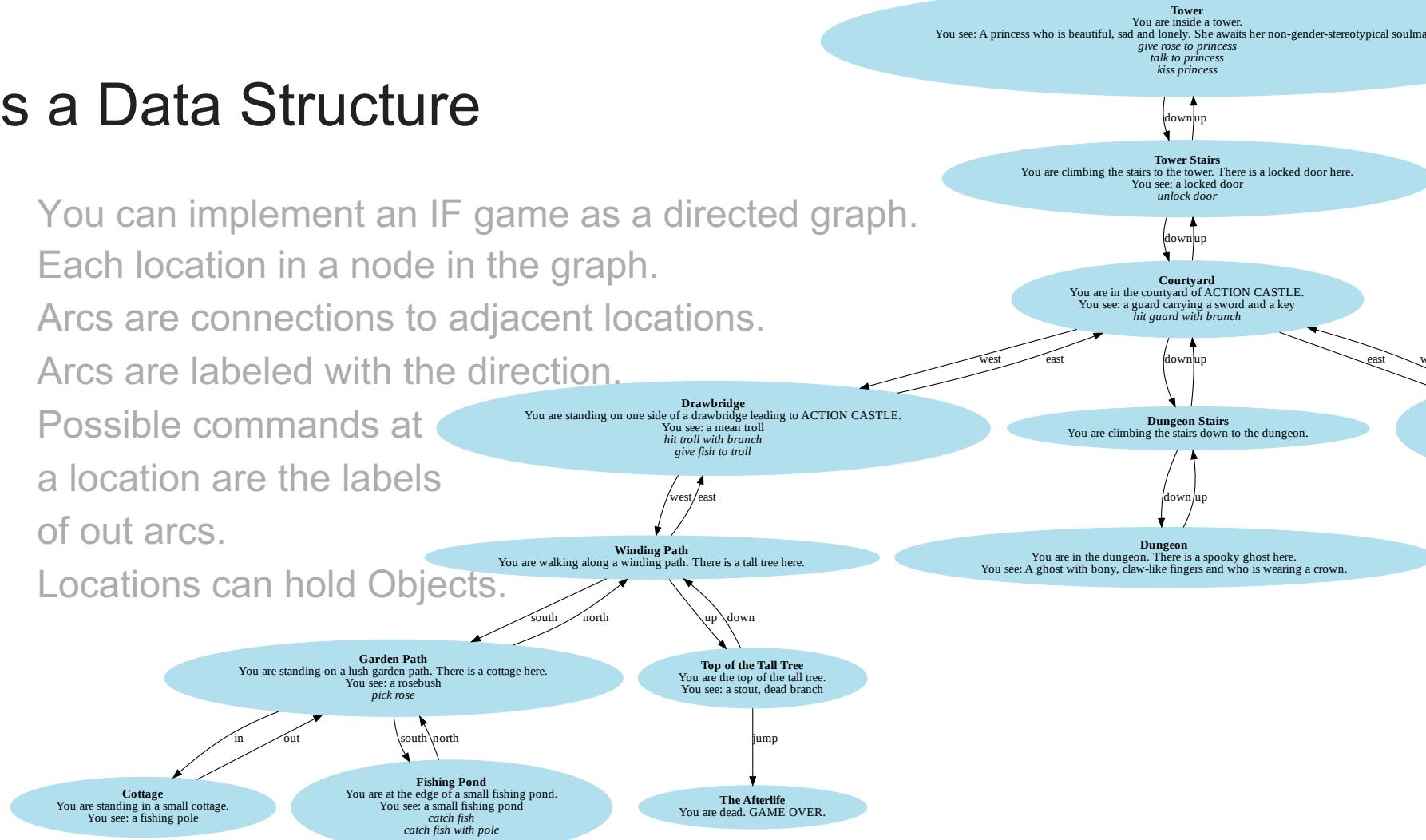
Look/L: look around to see what is nearby

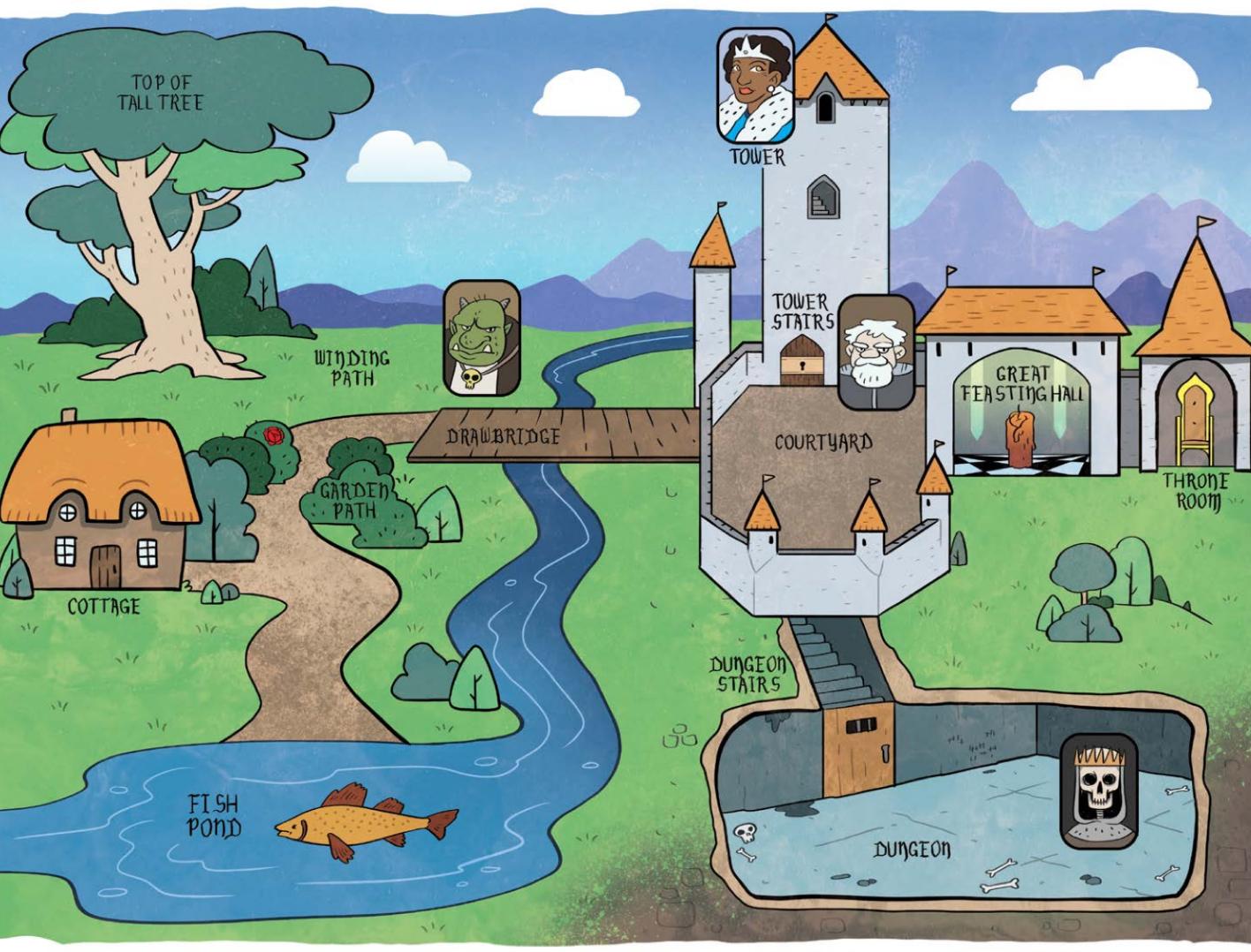
Not every direction is possible in every location.



As a Data Structure

- You can implement an IF game as a directed graph.
- Each location in a node in the graph.
- Arcs are connections to adjacent locations.
- Arcs are labeled with the direction.
- Possible commands at a location are the labels of out arcs.
- Locations can hold Objects.





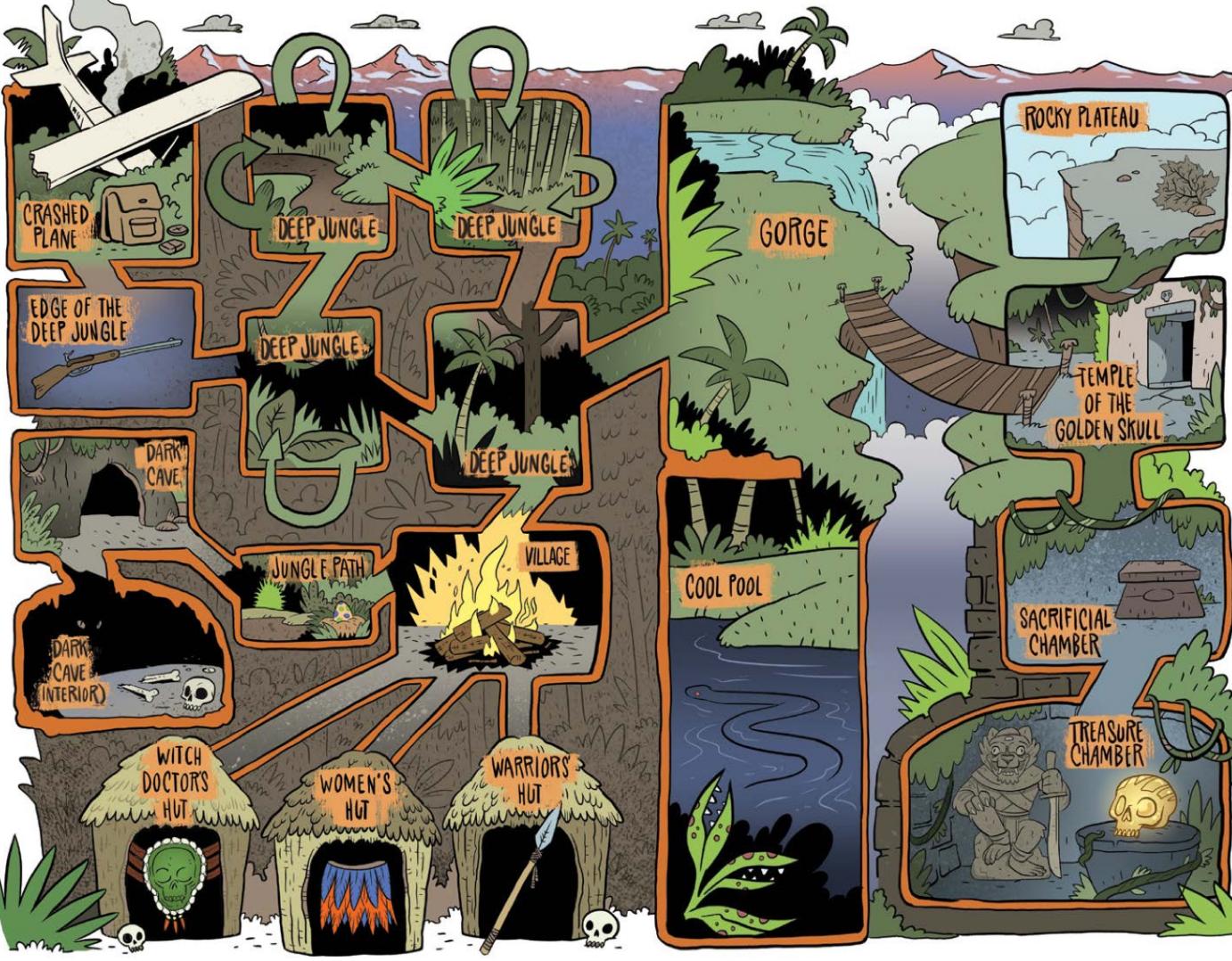
Your greatest challenge lies ahead—and downward.

ACTION CASTLE

The original Parsely game! Explore the lands of Action Castle, brave its dangers and claim the throne!

FANTASY
BEGINNER

CONTENT RATED BY PEGI
EVERYONE (10+)



DEEP JUNGLE

You are lost in the deep jungle.

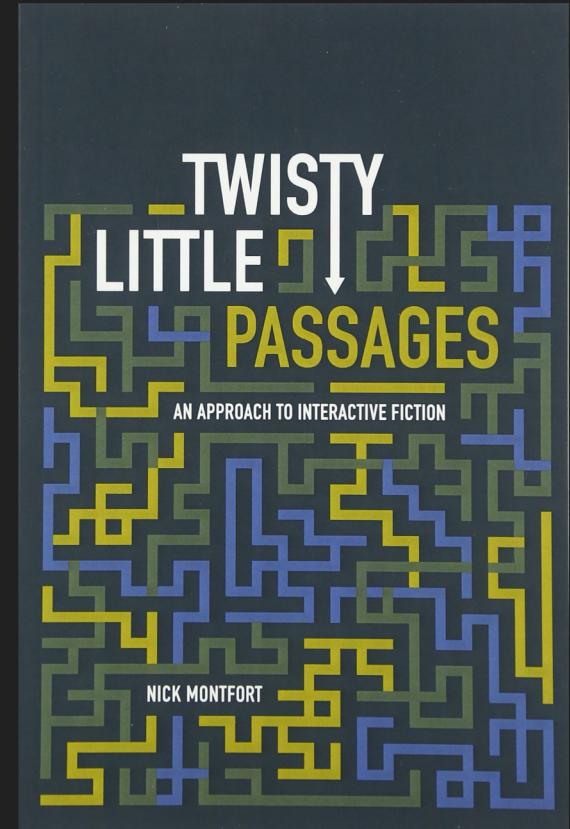
> Each location within the Deep Jungle has four exits: NORTH, SOUTH, EAST and WEST. Some of these—marked with U-turn arrows—lead back to the previously entered area.

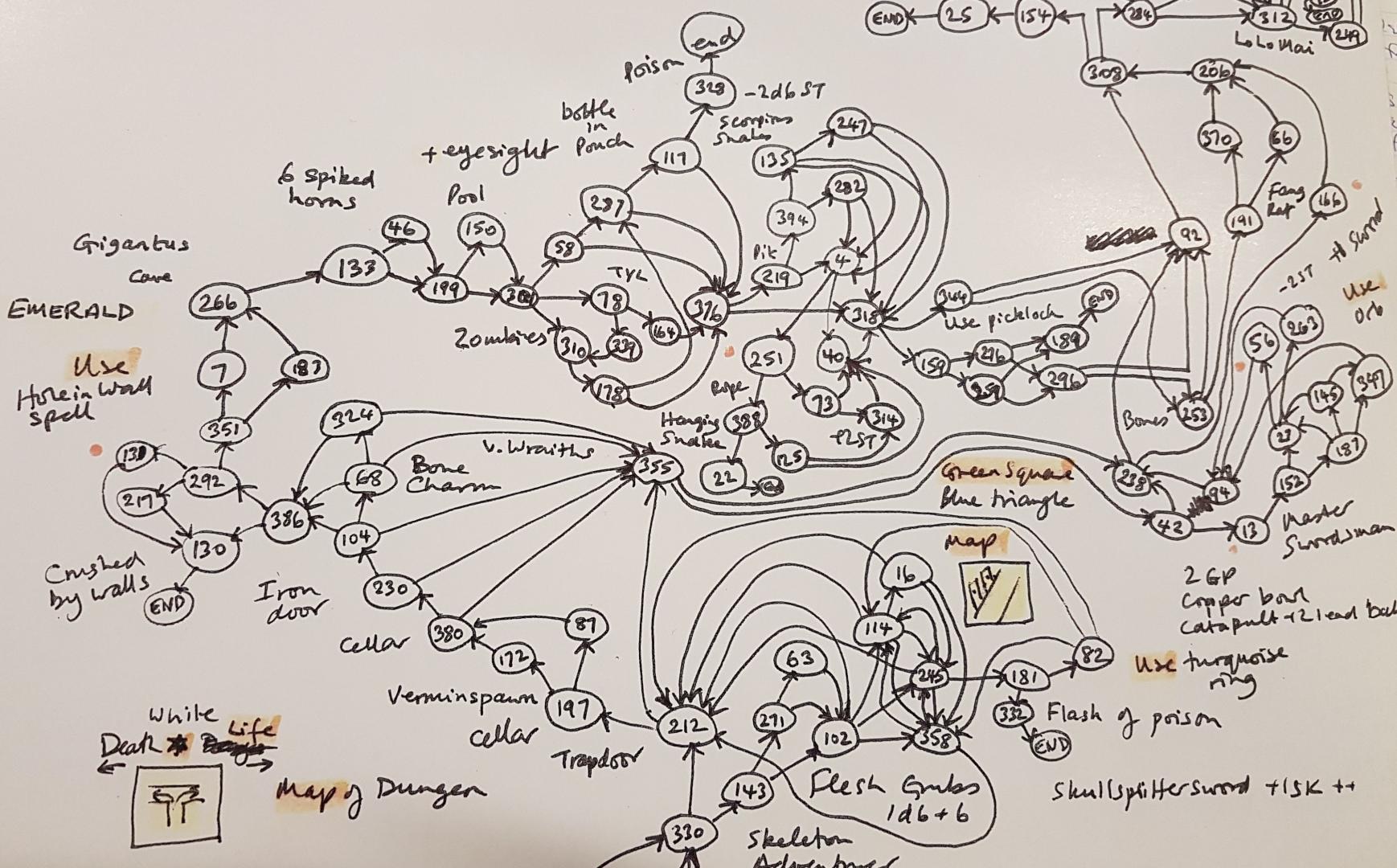
> If the player examines the compass while lost in the Deep Jungle, the actual exits are revealed and the U-turns are ignored.

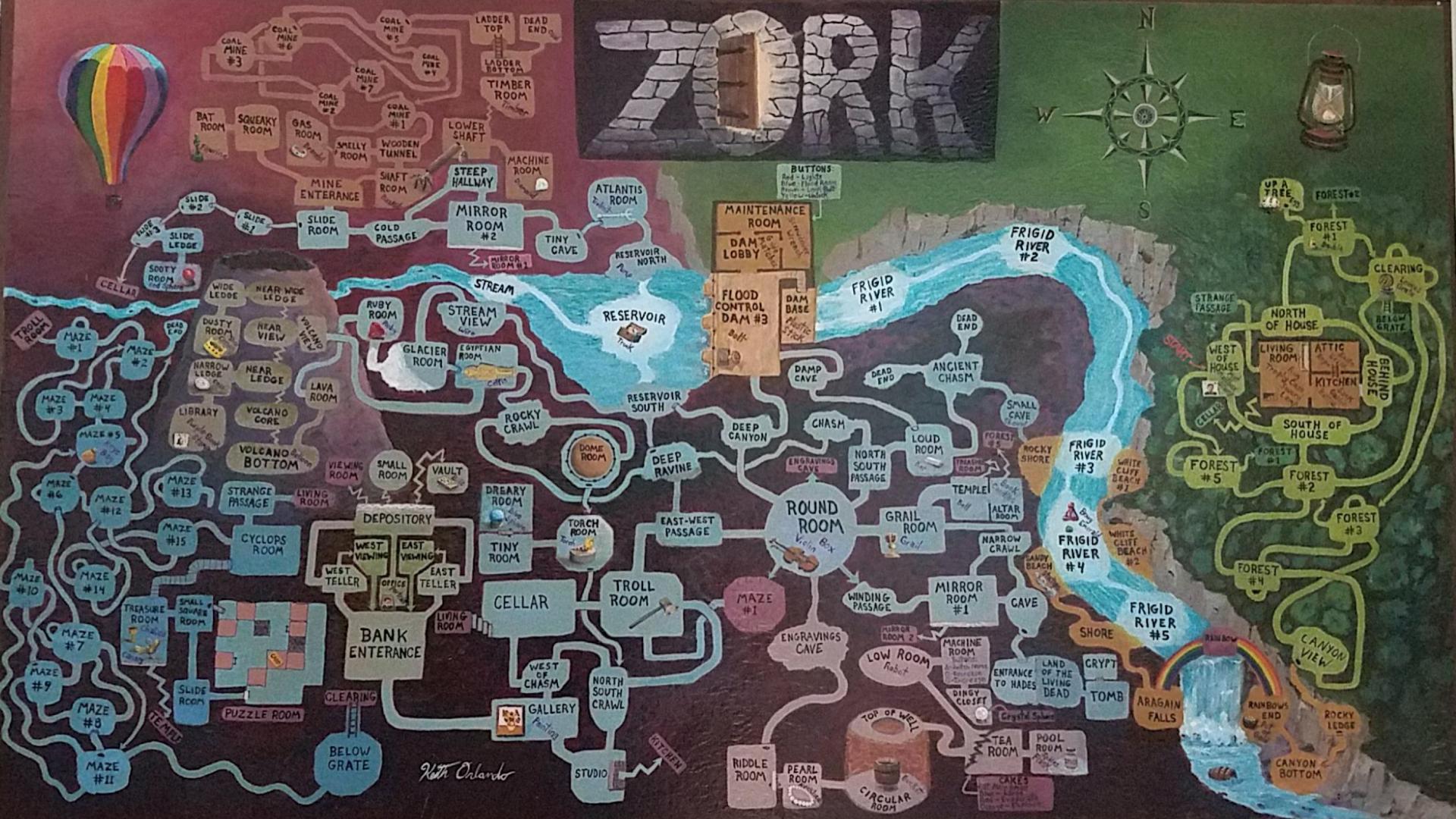
Maps and Mazes

There was a maze in Adventure that was essentially impossible to get through without making a map. The pirate's maze offered rooms that were all uniformly described as “a maze of **twisty little passages**, all alike.”

To figure out which room was which, the player character had to drop objects to mark the different rooms. The rooms, once all alike, could then be differentiated based on their contents, and mapping ping of the usual sort was possible.







Objects

In Adventure, instead of a realistic simulation of caving, the author placed five treasures within as an incentive to explore the cave.

The player had to figure out how to get past a snake to move deeper into the cave. The player is attacked by dwarves and their treasure is stolen by a pirate.

Examine Lamp

Items represent things in the game world. Usually they are mentioned in the description of a location like:

You are on the dungeon stairs. From above, you can make out some of the king's guards talking about current events. Someone's left an old lamp here.

Players can inspect them with the “EXAMINE” command.

> EXAMINE LAMP

This old lamp seems like it's seen some use. It ran out of oil ages ago.

Get Lamp

Players can pick up objects in the world, and they are added to the player's inventory with the "GET" command. The inventory is the set of things that the player has collected along the way.

Oftentimes, they are used to solve puzzles. For instance, you must have a lamp in your inventory in order to explore a dark cave.

To list all items that you have, you can issue the "INVENTORY" command (or just the letter "I").

Light lamp

Objects often have special commands associated with them. For instance, in order to solve the darkness puzzle you must say “LIGHT LAMP”.

Some special commands require more than one object in your inventory.

BREAK EGG

BREAK EGG WITH SWORD

On the table is an elongated brown sack, smelling of hot peppers.

A clear glass bottle is here.

The glass bottle contains:

 A quantity of water.

>w

You are in the living room. There is a door to the east. To the west is a wooden door with strange gothic lettering, which appears to be nailed shut.

In the center of the room is a large oriental rug.

There is a trophy case here.

On hooks above the mantelpiece hangs an elvish sword of great antiquity.

A battery-powered brass lantern is on the trophy case.

There is an issue of US NEWS & DUNGEON REPORT dated 28-JUL-80 here.

>get sword

Taken.

>break egg with sword

You rather indelicate handling of the egg has caused it some damage.

The egg is now open.

There is a golden clockwork canary nestled in the egg. It seems to have recently had a bad experience. The mountings for its jewel-like eyes are empty, and its silver beak is crumpled. Through a cracked crystal window below its left wing you can see the remains of intricate machinery. It is not clear what result winding it would have, as the mainspring appears sprung.



People are objects too

In Zork, a handful of living opponents thwart the adventurer: **the troll**, who stays put in a single room and serves as an obstacle; **the vampire bat**, who can carry off the adventurer; **the cyclops**, who can dine on the adventurer; and **the thief**, who wanders around the underground areas stealing items from the adventurer.

These people/opponents can be implemented as objects too. People objects often have a special command for dialogue via “TALK TO”.

Commands

Players input simple sentences such as “get key” or “go east”, which are interpreted by a text parser. Parsers may vary in sophistication; the first text adventure parsers could only handle two-word sentences in the form of **verb-noun** pairs.

Action Words

Farmer and Mrs. Pig certainly have a lot of children. And they are all doing something.



Lance Micklus

Lance Micklus, Inc.

You had to appeal to their sense of
overcoming the odds and figuring things out.

“open the red box with the green key then go north”.

Later parsers, such as those built on ZIL (Zork Implementation Language), could understand complete sentences. They could handle more complex inputs.

Early Parsers



Adventure's verb-noun parser was extremely primitive but removed ambiguity.



Ambiguity means that there are multiple interpretations of a sentence, which denote distinct meanings.

Lexical Ambiguity

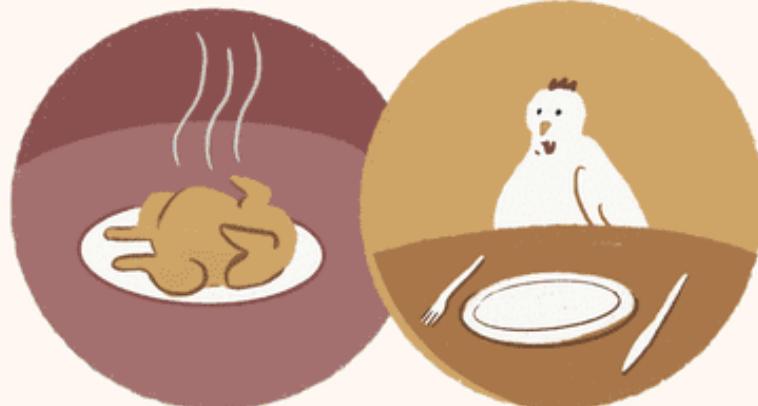
The presence of two or more possible meanings within a single word.



"I saw her duck."

Syntactic Ambiguity

The presence of two or more possible meanings within a single sentence or sequence of words.



"The chicken is ready to eat."

I saw the girl with the telescope.



I saw the girl with the telescope.



Z-machine

Infocom developed a virtual machine to deploy standardized “story files” on many platforms. The Infocom parser was the best of its era. It accepted complex, complete sentence commands when its competitors’ parsers were restricted to simple two-word verb-noun combinations. Its parser was actively upgraded, and later games would ‘understand’ multiple sentence input: ‘pick up the gem and put it in my bag. take the newspaper clipping out of my bag then burn it with the book of matches’.

Lance Micklus

Lance Micklus, Inc.

You had to appeal to their sense of
overcoming the odds and figuring things out.

Why were parsers so bad?



Limited computational resources. Computers had ≤ 128 KB of memory



Language is difficult. There are many things that make human languages genuinely challenging for a computer to process.



Keyword-based commands. Only exact matches worked properly. No synonyms, no paraphrases.



Everything was manual. Game developers had to anticipate all possible commands, and manually code the responses.



No Artificial Intelligence. This was prior to the advent of machine learning based natural language processing

Puzzles

In most interactive fiction, puzzles (sorts of challenges or obstacles) are part of the world the player character moves through. In order to complete the IF work, the interactor must figure out how to meet these challenges.

Lance Micklus

Lance Micklus, Inc.

You had to appeal to their sense of
overcoming the odds and figuring things out.

Puzzle Solutions

The solutions may be arrived at through the player character's senses or by having the player character manipulate things in the surroundings and then observe the results to determine the workings of the world.

Most interactive fiction does not have great replay value. You cannot simply "replay" a riddle if you know its answer.

However, once you learn to play a board game, the knowledge gained from playing it once game doesn't ruin the experience of playing it again.

Guess the Verb

A few puzzles require the player “guess the verb” and perform an action that would not be obvious from the commands available. The game’s parser does not understand unless the player uses a particular way to phrase the command (sometimes non-obvious). For example, if there is a crate to be opened with a crowbar and the only way to open it is to **“pry crate with crowbar”**, other actions like **“open crate with crowbar”** give a misleading response. Usually this is a deficiency in the parser.

Some games like Ad Verbum by Nick Montfort intentionally include guess-the-verb puzzles to good effect in contexts where the puzzles are explicitly about language.

Writing Style

Interactive fiction features two distinct modes of writing: the player input and the game output.

Player input is expected to be in simple **command form** (in linguistics, these are called **imperative sentences**).

> look in tea chest

“That was the first place you tried, hours and hours ago now, and there’s nothing there but that boring old book. You pick it up anyway, bored as you are”. –Graham Nelson *Curses* (1993)

The responses from the game are usually written from a **second-person point of view**, in present tense.

"I could not unlove him now, merely because I found that he had ceased to notice me".
– Charlotte Brontë's *Jane Eyre* (1847)

First-person – A first-person narrative is a mode of storytelling in which a narrator relays events from their own point of view using the first-person pronouns "I" or "we".
The narrator is the protagonist, or retelling events that they witnessed.

"You are not the kind of guy who would be at a place like this at this time of the morning. But here you are, and you cannot say that the terrain is entirely unfamiliar, although the details are fuzzy."

Jay McInerney's *Bright Lights, Big City* (1984)

Second-Person – The reader is character within the story. This is done with the use of second-person pronouns like *you*. Stories and novels in second person are comparatively uncommon.

Third-Person – The narrator refers to all characters with third person pronouns like *he*, *she*, or *they*, and never first- or second-person pronouns. This makes it clear that the narrator is an unspecified entity or uninvolved person who conveys the story and is not a character of any kind within the story.

Stories

And it was clear to me from my small experience of Adventure,
the description of Zork, the stuff I saw
on these monochrome monitors



Robert Pinsky
Poet Laureate of the
United States, 1997-2000

Artificial Intelligence

From: goetz@acsu.buffalo.edu (Phil Goetz)
Subject: Re: Adventure generators (skippable)
Newsgroups: rec.arts.int-fiction
Date: 29 Oct 92 04:40:05 GMT
Sender: nntp@acsu.buffalo.edu
Organization: State University of New York at Buffalo/Comp Sci

morpheus@sage.cc.purdue.edu (Morpheus Nosferatu) writes:

>Has anyone ever worked on, or even heard of, an adventure generator?
>
>I'm not talking about an adventure design language like TADS or Alan,
>but rather a stand-alone adventure generator that produces complete
>adventures, where the user need only give a minimal degree of input,
>such as the level of complexity, type of adventure (mystery, treasure
>hunt, etc.), size of adventure, and so forth?
>...
>But as anyone ever heard of someone trying to come up with a generator
>which would produce infocom-style text adventures? I can just imagine
>what kind of limitations it would have, but I'm curious to know if
>anyone has tried this, and if so what degree of success they've had.

From: goetz@acsu.buffalo.edu (Phil Goetz)

Subject: Re: Adventure generators (skippable)

Newsgroups: rec.arts.int-fiction

Date: 29 Oct 92 04:40:05 GMT

Sender: nntp@acsu.buffalo.edu

Organization: State University of New York at Buffalo/Comp Sci

morpheus@sage.cc.purdue.edu (Morpheus Nosferatu) writes:

>Has anyone ever worked on, or even heard of, an adventure generator?

>

>I'm not talking about an adventure design language like TADS or Alan,
>but rather a stand-alone adventure generator that produces complete
>adventures, where the user need only give a minimal degree of input,
>such as the level of complexity, type of adventure (mystery, treasure
>hunt, etc.), size of adventure, and so forth?

>...

>But as anyone ever heard of someone trying to come up with a generator
>which would produce infocom-style text adventures? I can just imagine
>what kind of limitations it would have, but I'm curious to know if
>anyone has tried this, and if so what degree of success they've had.

No. ... The generator you speak of is not written, not being written, and not anywhere on the horizon. In 50 years, maybe. In 20, definitely not. The problem of writing interesting stories, which adhere to someone's definition of a plot (with goal explanations, conflict, resolution, complication, climax, etc., all occurring at appropriate intervals) is very hard, and I don't expect a solution soon. But the problem of writing clever puzzles involves much greater creativity, and I have seen NO evidence that ANYBODY has a clue in these creativity issues; the most you will find in the field are a few vague theories of creativity.

This problem is what Stuart Shapiro calls "AI-complete": Solving it would be equivalent to solving all the other problems of AI.

Phil

goetz@cs.buffalo.edu

Text World

In 2018, Microsoft released a system that automatically generates text adventure games. These games can be played by human gamers, but they are really intended to be a testbed for automatic agents.



For more information go to
aka.ms/textworld

Natural Language Processing

Word Embeddings

TODO: Move slides over from other presentation

Text Classifiers

Given a corpus of text excerpts that each have a label associated with them, a classifier can be trained to predict the label given the text.

Example applications include sentiment classification, style prediction, and **TODO**.

Training a classifier

- Simple method
 - Compute a bag-of-words vector for each text excerpt
 - Vector's length is the number of vocab words and each position contains the frequency of that word in the excerpt.
- State-of-the-art method
 - Finetune a large neural language model (BERT, ELECTRA, GPT-2, etc.) that has been pre-trained for some other task.

Dependency Parsing

Co-reference Resolution

Course Overview

Topics to be covered

- Building a text adventure game
- Reinforcement Learning for solving text-adventure games
- Common-sense reasoning for interactive fiction
- World representations
- Automatic extraction of narrative structure from text
- Text Generation for stories and open-ended dialog
- Controllable text generation

Topics to be covered (continued)

- Human-computer interaction principles for collaborative storytelling
- Bias in generative language models
- Connections to computer vision, speech, and other multimodal areas
- History of storytelling and artificial intelligence

Grading

- Five homeworks worth a total of 50%
 - Can be completed in groups ≤ 2
- One final project worth 40%
 - Can be completed in groups ≤ 4
- Class participation worth 10%

How to contact us

- Slack channel (for discussing homeworks, project, anything related broadly to the class)
- Mailing list for announcements about homeworks and lectures.
- Email us with specific questions (ccb@seas.upenn.edu, daphnei@sea.upenn.edu)

Daphne will have office hours after class at 4:30 PM.

Student Presentations

- Prepare a 10-15 minute presentation on a paper from the reading list.
- Your presentation should summarize the work and discuss the ways it's applicable to either interactive fiction or automatic story generation.
- Grading:
 - 50%: **Send us your slides by Monday at 3 PM the week of**
 - 50%: presentation to the class

Presentation Signups

<https://bit.ly/36I1nAs>

Fill out this form by the end of today.



Homework 1

Implement a simple text adventure game in Python.

Due 01/23/20 at 3PM

Homework 2

Improve your text adventure game using NLP!

Due 01/30/20 at 3PM

Homework 1: Build a Text-Adventure Game

Instructions

In this homework assignment, you will write your own classic text adventure game. This homework can be completed in groups of up to 2 people. You will implement two text adventure games. One will be a re-implementation of the **Action Castle** game, and one will be a game that you design yourself. The game that you design can be on any topic, or can tell any story of your choice. We will play the games that you design during class, and part of your grade will be awarded based on how creative/exciting your classmates think your game is.

Starter code

We have provided [starter code for a basic text adventure game](#). You are free modify it however you want, and bring in any dependencies you feel will be useful.

Task 1: Implement Action Castle

Action Castle is a game by Jared A. Sorensen. It is included in his book [Parsley](#), which is a collection of games inspired by the text-adventures of the 1980s. Parsley is a party game where you take on the role of the parser, and the players shout out commands like GO NORTH, LIGHT LAMP or GET SWORD. You obtusely follow player's commands, simulating a computer's limited vocabulary. The effect is surprisingly hilarious and fun.

We got Jared's permission to distribute the Action Castle module in class, and you can also [buy the Parsley book for \\$20](#) if you'd like to support an awesome indie game developer.

You should modify the provided code to:

1. Create the 13 locations from Action Castle (Cottage, Garden Path, Fishing Pond, Winding Path, Top of the Tall Tree, Drawbridge, Courtyard, Tower Stairs, Tower, Dungeon Stairs, Dungeon, Great Feasting Hall, Throne Room).
2. Create the items for the game (fishing poll, rosebush, club, fish, the troll etc.).
3. Update the code so that it can handle the actions/commands/preconditions that are described by the Action Castle module.

Need a hint on how to get started? I was able to re-implement the whole of the Action Castle game* using the starter code by modifying the `build_game` function, the `check_preconditions` function, and by adding a few new methods to the [Special functions section](#). None of the other starter code needed to be modified. It took me about 5 hours total to implement the game.

*Except for this part: *The ghost will reach out for the player to stop his heart if the player lingers here.* I skipped that part of the game.