

IF Transcript Parser

The IF Transcript Parser is a python script that will format Interactive Fiction transcripts (or logs) as HTML files to make them easier readable for authors and testers.

The background

Sometimes I test IF text adventures and struggle with reporting my findings to the author. Usually I send the transcript and some notes, but it's not always easy to explain the things you encounter and provide a good reference point.

In the past I formatted the transcript in a text editor, such as MS Word, and made annotations and highlighted my commands. This is a time consuming process. Another way would be to just give the transcript to the author and let him figure it out, but the plain text transcripts are often not easy to analyze and small details are sometimes overlooked.

I wanted to create something that does the formatting for me, maybe even allow me to write notes directly in the game when I'm testing and display those in a way that makes them easy to find.

This python script does that for me and allows me to send a simple report to the author in a well formatted format.

What the script does

The script takes a transcript and generates a HTML page from it. It will highlight the user input, tries to identify room changes, generate an index of the rooms visited and allows you to make annotations.

The output will be a single HTML file that can be opened and viewed in any modern browser. It uses CSS to format the output.

Supported Transcript Formats

So far the script has been tested with Zcode and Glulx transcripts created with various interpreters. It might support other formats, but this hasn't been tested yet.

The script might not always identify everything right away. Commands that are specific to an adventure game or control the editor might not always be recognized. The same applies to room changes.

So far the only supported language is English.

If you have transcripts that you experience issues with, feel free to send them to me and I'll try to improve the script parser.

Requirements

The script is a python script and was developed for Python 2.7. It might work with Python 3, but wasn't tested with it.

Any platform (Linux, Windows, MAX) should work.

Usage

The script will be executed from the command line.

Usage: if_transcript_parser.py <transcript_file> [output_file]

The script requires at least one command line parameter: the name of the transcript that you want to parse. You can specify the relative or absolute path if the transcript is not in the same directory as the parser.

The second command line parameter (output_file) is optional. If you specify it the target HTML will be written into this file. If you didn't specify the second parameter the output will be printed in the console.

Output Formatting

Game name

The script tries to identify the name of the game you were playing from the transcript description at the beginning. If everything was successful the title of the HTML output will look like this:

Transcript for "Sting of the wasp"

Room Navigation

The script tries to detect the rooms that the player visited while playing the game. It will mention the rooms as an index in the beginning of the transcript with links to the part where the room was visited in the context.

If you revisit rooms while playing they will be mentioned more than once, in the order that the player visited them.

Transcript for "Sunburn"

Room Navigation

- [Corner Office](#)
- [Narrow Hallway](#)
- [Tiny Closet](#)
- [Narrow Hallway](#)
- [Tiny Closet](#)
- [Narrow Hallway](#)
- [Tiny Closet](#)
- [Narrow Hallway](#)
- [Narrow Hallway](#)
- [Narrow Hallway](#)

Player Input

The player input (command) will be highlighted, so that it's easier to find. The highlighting is fairly simple and checks for the line beginning as indicator for the input command.

```
>turn off audio player
You press the red button. The audio player ceases its noise.

>examine card
(the business card)
It's a standard plastic leaf business card. Looks like it must be a sample card,
unless someone named "John Doe" really did work in this office as a ...
"Business card."

>examine door handle
It's a sturdy plasteeel door. The handle is a bit warm to the touch.
There's a business card holder on this side of the door.
```

Annotations

Annotations are not directly supported by the interpreter, so there needs to be a key component that will tell the parser that a note or annotation is coming and not a normal command for the game.

For simplicity reasons this is the sign “#” at the start of the input command.

So to write a note in the game you would input something like this:

```
>close blouse
You hastily button up your blouse, careful not to break a nail.

># great attention to detail
That's not a verb I recognise.

>inv
You are carrying:
  your Louis Vuitton bag (which is open)
  your make-up
  a pair of banquet tickets
  your watch (being worn)
  your ensemble (being worn)
```

and the parser would format it like this:

<pre>>examine blouse Nothing but the best for you. Pumps from Prada, skirt by Yves Saint Laurent, a gorgeous silk blouse from the much-coveted Vera Wang collection which is currently hanging about your shoulders, exposing your three thousand dollar chest. >close blouse You hastily button up your blouse, careful not to break a nail. >inv You are carrying: your Louis Vuitton bag (which is open) your make-up a pair of banquet tickets your watch (being worn) your ensemble (being worn)</pre>	<div>great attention to detail</div>
--	--------------------------------------

Please note that the error message that the game generated (because the interpreter didn't know you were going to write a comment) has been suppressed by the parser.

Contact

For comments, questions, bug reports or feature requests, please contact me at:
dirk@brainbombs.de