

***Coraline* als begehbare Roman: Ein Englisch-Informatik-Projekt aus dem Distanzunterricht**

Rau, T.
Graf-Rasso-Gymnasium Fürstenfeldbruck

DOI: 10.18420/ibis-01-01-07

Zusammenfassung

In der besonderen Situation des Distanzunterrichts im Frühling 2020 entstand mit einer 9. Klasse auf der Grundlage der aktuellen Lektüre des Englischunterrichts ein Computerspiel aus dem Genre *Interactive Fiction* (auch: Textadventures). Die Implementierung übernahm dabei die Lehrkraft, Text und Ideen des Spiels stammen weitgehend von den Schülern und Schülerinnen; auch informatische Inhalte ließen sich dabei veranschaulichen, insbesondere Modellierung und Relationen.

Einleitung: Die besondere Ausgangssituation

Im Frühling 2020 unterrichtete ich eine 9. Klasse sowohl im Fach Englisch als auch in Informatik. Für Englisch war bereits als Lektüre der Kurzroman *Coraline* von Neil Gaiman bestellt und an die Klasse ausgeteilt, im Fach Informatik war das Thema im damaligen Lehrplan Datenbanken und Relationen sowie die Modellierung von Datenbanken mit UML-Klassendiagrammen. Zu diesem Zeitpunkt wurden die Schulgebäude pandemiebedingt geschlossen, Unterricht fand von zuhause aus statt; die Bedingungen für synchronen Unterricht, etwa in Form von Videokonferenzen, waren sehr unterschiedlich. Vor diesem Hintergrund beschloss ich, mit der Klasse ein einfaches textbasiertes Computerspiel zur Lektüre zu erstellen. Die Programmierung würde dabei weitgehend ich übernehmen, dennoch gab es informatische Aufgaben für die Schülerinnen und Schüler.

Hintergrund: Die Romanvorlage

Coraline von Neil Gaiman ist ein kurzer phantastischer Roman um das Mädchen Coraline, das mit ihren Eltern in ein altes Haus umzieht. Dort findet sie eine Tür, die in eine Art paralleles Haus führt, in dem es die gleichen Zimmer, Hausbewohner und Bewohnerinnen gibt, aber alles auf unheimliche Weise verfremdet. Auch eine andere Mutter gibt es dort, mit angenähten Knöpfen statt Augen, die um Coraline wirbt, sogar Coralines echte Eltern entführt und gefangen hält, um Coraline bei sich zu behalten. Mit Hilfe einer Katze, die in beiden Welten lebt, befreit Coraline schließlich ihre Eltern und kann der anderen Mutter entkommen.

Die Handlung des Romans eignet sich gut für die Umsetzung in ein einfaches Spiel:

- Die Welt ist relativ überschaubar.
- Dennoch spielt Exploration eine große Rolle.
- Es gibt eine Tür mit Schlüssel als klares Hindernis (an den Schlüssel gelangt Coraline erst durch eine Kombination von Küchenstuhl und Besen), daneben einen verschließbaren Spiegel als Gefängnis; das typische Suchen nach versteckten Gegenständen (drei einzelnen Murmeln).
- Es gibt relativ wenig Personen, und vor allem wenig Interaktionen und überhaupt wenig Interaktionsmöglichkeiten zwischen ihnen (zwischen Coraline und den Nachbarn und auch den Eltern findet wenig Kommunikation statt; die Erwachsenen hören Coraline nie richtig zu).
- Es gibt nur eine Hauptperson, keine Perspektivenwechsel.

Hintergrund: Das Genre Interactive Fiction

Seit Mitte der 1970er Jahre gibt es das Genre textbasierter Spiele. Sehr bekannt sind die Spiele der Firma Infocom wie *Zork* (1977, 1980) oder *The Hitchhikers' Guide to the Galaxy* (1984), bis 1988 erschienen allein dort mehr als 30 derartiger Spiele. Beim Spielen gibt es zuerst eine kurze Beschreibung der aktuellen Situation: „You are standing in an open field west of a white house. There is a small mailbox here.“ Es gibt meistens überhaupt keine Grafik, sondern wirklich nur Text. Ein Prompt fordert zum Eingeben von Anweisungen auf, die je nach dem im Spiel verwendeten Parser unterschiedlich komplex sein können: „Open the mailbox/take the leaflet/read it.“ Üblicherweise läuft man bei diesen Spielen durch eine Reihe von Räumen und interagiert dort mit Gegenständen; die Spielwelt ist oft voller Objekte, die – vielleicht – gegessen oder angezündet geworfen werden können, mit denen man Türen aufschließt und Schatztruhen aufbricht.

Ab Mitte der 1980er Jahre verlor das Genre an Beliebtheit und Bedeutung, heute gibt es nur noch Reste einer kommerziellen Verbreitung. Aber seit Mitte der 1990er Jahre entwickelte sich eine große Szene von Hobbyautoren und -autorinnen mit vielen, auch groß angelegten Spielen und der entsprechenden Infrastruktur (Archive, Programmierumgebungen, Foren, Wettbewerbe). Auch in der Informatikdidaktik taucht das Programmieren solcher Spiele gelegentlich auf, *Objects First With BlueJ* von David J. Barnes und Michael Kölling enthält zum Beispiel ein Projekt „World of Zuul.“

Hintergrund: Die Programmiersprache

Inform 7 (ab 2006) ist eine domänenspezifische Sprache, die der englische Mathematiker Graham Nelson speziell für Werke der *interactive fiction* entwickelt hat. Technisch ist sie eine Makrosprache, die auf Nelsons älterem Inform 6 (1993) aufsetzt, das wiederum für die virtuellen Maschinen Z-Machine (Joel Berez, Marc Blank 1979) oder Glulx (Andrew Plotkin 1999) kompiliert. Für beide virtuellen Maschinen gibt es viele Implementierungen. Die Sprache ist seit 2022 Open Source und wird von einer großen Community genutzt und unterstützt.

Inform 7 kennt Objekte, Klassen, Attribute, Datentypen, Einfachvererbung, Polymorphie, Kontrollstrukturen, Rekursion. Methoden gibt es, allerdings nur globale, die nicht Objekten zugeordnet sind; sie spielen insgesamt eine kleine Rolle. Das Verhalten wird hauptsächlich über regelbasierte Programmierung gesteuert. Eine große Anzahl an solchen Regeln ist bereits vorgegeben: Sie bestimmen, was geschieht, wenn die Spielfigur angewiesen wird, Objekte zu nehmen, zu essen, zu trinken, oder sie zu werfen versucht. Dabei werden die Klassen und Attributwerte der Objekte berücksichtigt – ob sie tragbar oder essbar, ob sie Behälter oder Fahrzeuge sind. Das Programmieren besteht hauptsächlich im Anlegen von Klassen, Objekten, Attributwerten und zusätzlichen Regeln, die das bestehende Weltmodell ergänzen und einfache oder komplexe Situationen abdecken können.

Hello World in Inform 7

Zum besseren Überblick wird hier kurz geschildert, wie ein Spiel in Inform 7 programmiert wird. Nach dem Installieren und Starten der Entwicklungsumgebung wird in Form von textuellem Programmcode das Spiel erzeugt. Das kleinstmögliche Spiel lautet in vollständiger Form: „The World is a room.“ In diesem Fall besteht die Spielwelt aus einem einzigen Raum namens „The World“. Die Spielwelt ist davon abgesehen leer, aber die Spielfigur wird in diesem Raum platziert und kann bereits Anweisungen wie „jump, wait, sleep, look at yourself, kick yourself“ ausführen, auf die vordefinierte, aber natürlich änderbare Reaktionen folgen. Optional kann man die vordefinierten Raum-Attribute mit Werten belegen, etwa eine kurze und eine ausführliche Raumbeschreibung, und Ausgänge zu anderen Räumen. Für die Wertzuweisung gibt es syntaktisch etliche verschiedene Möglichkeiten. Hier zum Vergleich der Inform-7-Code und das ungefähre Äquivalent in Java:

Coraline's Room is a room. The description is "This is your room. A window lets you see something of the world outside."

The Hallway is a room. It is north of Coraline's Room. "You are standing on a carpet that covers the floor."

```
Room coralinesRoom = new Room();
coralinesRoom.setDescription("This is
your room...");
Room hallway = new Room();
Hallway.setDescription("You are
standing...");
coralinesRoom.setNorth(hallway);
hallway.setSouth(coralinesRoom);
```

Sobald man mehrere Räume hat, kann man sich mit Anweisungen wie „go north“ zwischen ihnen bewegen.

Dann werden Objekte angelegt. Auch diese kommen mit vordefinierten Attributen und Attributwerten, die überschrieben werden können:

The window is a thing in Coraline's room. The window can be either open or closed. It is closed. The description is "You see trees, fields, and beyond them, on the horizon, distant purple hills."

```
Thing window = new Thing();
coralinesRoom.add(window);
window.closed = true;
window.setDescription("You see trees...");
```

Regeln bestimmen das Verhalten der modellierten Spielwelt:

*Instead of opening the window:
if the window is open, say "It's already open."
otherwise:
say "You open the window a little bit."
now the window is open.*

```
@Override void open(Thing t) {
    super.open(t);
    if (t==window) {
        if (window.open) print("It's already open."); else { print("You open it a little bit."); window.open = true; }
    }
}
```

Das ungewöhnlichste Element an Inform 7 ist dabei vielleicht die Nähe zur natürlichen Sprache. Die Elemente der Programmiersprache entsprechen in weit größerem Maß der englischen Sprache als üblich; Groß- und Kleinschreibung spielen keine Rolle, Leerzeichen in Bezeichnungen sind möglich und erwünscht, es gibt viele Synonyme. Einfachen Code zu schreiben fällt dadurch sehr leicht, und komplexer Code ist leicht zu verstehen; das setzt die Vorstellung von „considering programs to be works of literature“ (Knuth 1984) auf eine ungewöhnliche Art um.

Allerdings ist die Nähe zur natürlichen Sprache trügerisch und verleitet bei komplexeren Programmieraufgaben mitunter zu falschen Annahmen, was wie interpretiert oder auch nicht interpretiert wird. Dennoch können erfolgreich kleine Informatikprojekte in Inform 7 geschrieben werden, im Distanzunterricht ist das allerdings kaum möglich, so dass die Implementierung ganz auf die Lehrkraft beschränkt war.

Das eigentliche Projekt

Rahmenbedingungen

Das Projekt wurde mit einer Klasse von 27 Schülerinnen und Schülern durchgeführt; es lief über einen Zeitraum von etwa fünf Wochen, und durchgehend im Distanzunterricht. Dabei wurde gemeinsam ein Spiel modelliert

Informatische Inhalte: Modellierung

Die hauptsächliche Aufgabe war die Modellierung der Welt entsprechend der Romanhandlung. Dass die Hauptfigur Coraline ihr eigenes Zimmer hat, ist klar. Eine Schülerin wünscht sich dazu noch ein Badezimmer, weil es wahrscheinlich ist, dass es das gibt. Sobald das implementiert ist, vermisst die Schülerin, dass Coraline sich darin nicht Zähne putzen, waschen oder duschen kann. Duschen impliziert das mögliche Ausziehen von Kleidung – aus pädagogischen Gründen problematisch, und Kleidung wiederum macht ein ganzes Fass an Modellierungsproblemen auf: Wenn es eine Hose gibt, gibt es dann auch Unterwäsche und Socken? Welche einzelnen Kleidungsstücke soll es geben, und in welcher Reihenfolge werden sie an- und aus- oder übereinandergezogen? Bald stellt sich heraus, dass die Modellierungstiefe darauf ankommt, was für das Gesamtspiel am sinnvollsten ist. So haben Schülerinnen und Schüler die Gelegenheit zu lernen, dass es bei der Modellierung eines – hier: fiktionalen – Systems darauf ankommt, bestimmte Elemente und Eigenschaften wegzulassen, zu abstrahieren, die für den Verwendungszweck unnötig oder störend sind.

Eine Schülerin schreibt als Beschreibung des Wohnzimmers: „In the room you can find your grandmother’s furniture, along with a wooden coffee table, a side table, a heavy glass ashtray and the oil painting of a bowl of fruit.“ Nicht unbedingt klar ist dabei am Anfang, dass diese Beschreibung dem Spieler nahelegt, dass mit den genannten Objekten auch interagiert werden kann, und sei es auch nur, sie zu betrachten. Sie müssen also modelliert werden, und zwar meistens mit zwei Beschreibungen: einmal in der realen Welt und einmal in der verfremdeten Parallelwelt.

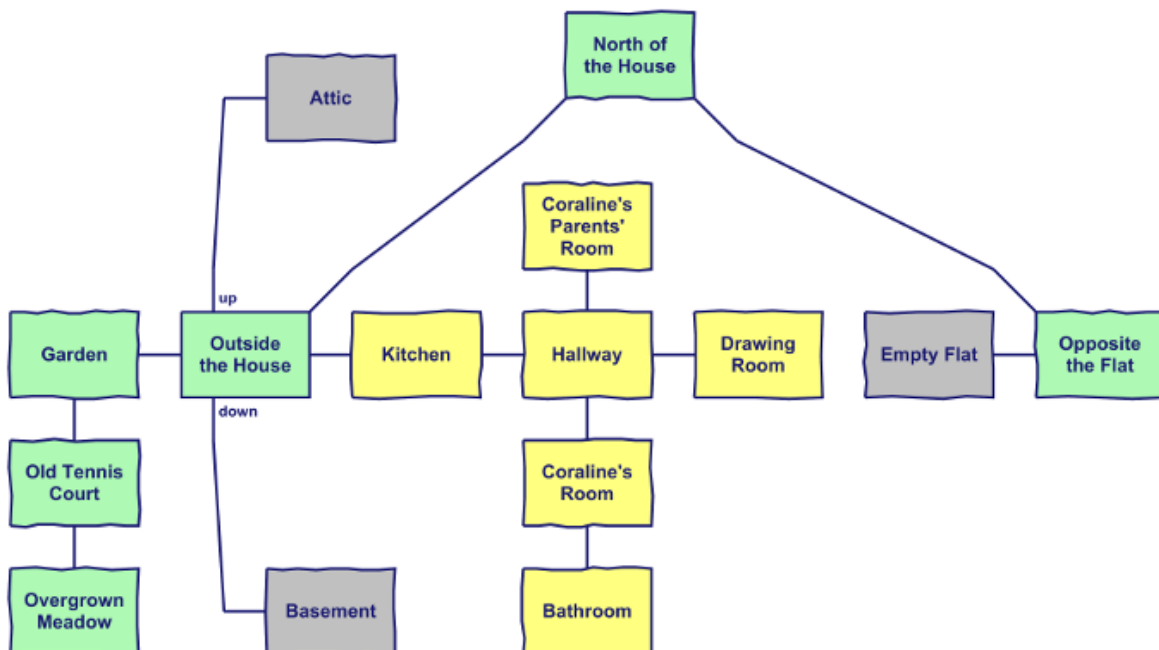


Abbildung 1: Die modellierte Spielwelt mit ihren Räumen

Informatische Inhalte: Relationen

Der zum Zeitpunkt des Projekts aktuelle Lehrplan für die 9. Jahrgangsstufe enthält Relationen als Grundlage für relationale Datenbanken sowie Datenbankmodellierung mit UML-Klassendiagrammen (ohne Methoden). Konkret heißt es: „Die Schüler strukturieren Daten an Beispielen aus ihrer Erfahrungswelt. Dabei erkennen sie, dass die Struktur der Klassen sowie deren Beziehungen sehr übersichtlich in Klassendiagrammen dargestellt werden können.“ Zwar geht es in diesem Projekt nicht um Beispiele aus der Erfahrungswelt, sondern um das Modellieren einer fiktiven Welt, dennoch lässt sich das am Spiel üben oder wiederholen. Inform 7 arbeitet viel mit Relationen und bietet eine eigene Datenstruktur dafür an. („Liking is a relation that relates various people to one person.“) Bereits eingebaut in die Welt-Regeln sind räumliche Relationen etwa für Ist-enthalten-in und Befindet-sich-auf.

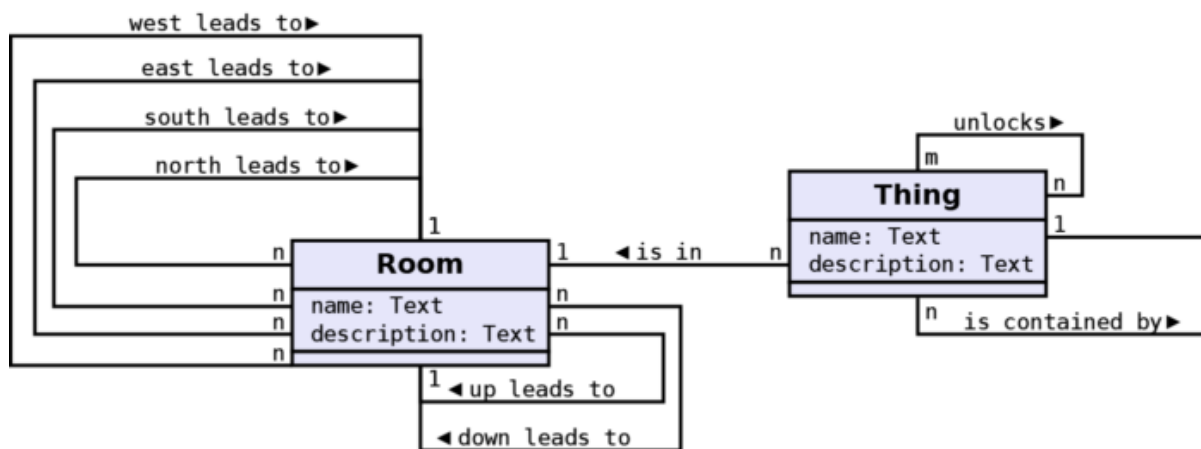


Abbildung 2: Klassendiagramme zur Überführung in ein relationales Modell

Eine Aufgabe dazu, noch vor der Einführung des eigentlichen Spiels, lege ein Klassendiagramm vor, das in Relationenschema und dann in Tabellen überführt werden sollte:

1. Überführe das Klassendiagramm in ein relationales Modell. Im verfeinerten Modell reichen dafür drei Relationen.
2. Zeichne für die Relationen jeweils eine Tabelle mit den Attributen als Kopfzeile und dazu jeweils drei Einträgen/Datensätzen.
3. Wenn der Inhalt deiner Tabellen Sinn ergibt, solltest du leicht eine kleine Karte zeichnen können!
4. Bonusfrage: Welche nicht sinnvollen Situationen könnten mit diesem System entstehen?

| Room | | | | | | | |
|--------------|-------------------------------|-----------------------|-----------------------|----------------------|----------------------|--------------------|----------------------|
| <u>Name</u> | <u>Description</u> | <u>North-Leads-To</u> | <u>South-Leads-To</u> | <u>East-Leads-To</u> | <u>West-Leads-To</u> | <u>Up-Leads-To</u> | <u>Down-Leads-To</u> |
| Küche | Essen machen | | | | Wohnzimmer | | Keller |
| Wohnzimmer | Entspannen, Fernsehen, etc. | | Schlafzimmer | Küche | | | |
| Schlafzimmer | Schlafen | Wohnzimmer | Büro | | | | |
| Büro | Arbeiten | Schlafzimmer | | | | Dachboden | |
| Keller | Aufbewahrung von Gegenständen | | | | | Küche | |
| Dachboden | Gerümpel | | | | | | Büro |

| Thing | | | |
|-------------|--------------------------|-----------------------------------|------------------------|
| <u>Name</u> | <u>Description</u> | <u>Is-Contained-By-Thing-name</u> | <u>Is-In-Room-name</u> |
| SchlüsselX | Aufsperrern von Kiste | | Wohnzimmer |
| SchlüsselY | Aufsperrern von WZ-Tür | Kiste | Wohnzimmer |
| SchlüsselZ | Aufsperrern von Büro-Tür | Regal | Schlafzimmer |
| Kiste | Behälter | | Wohnzimmer |
| Regal | Aufbewahrung | | Schlafzimmer |

| Unlocks | |
|-------------------|-------------------|
| <u>Thing-Name</u> | <u>Thing-name</u> |
| SchlüsselX | Kiste |
| SchlüsselY | Wohnzimmertür |
| SchlüsselZ | Schlafzimmertür |

Abbildung 3: Schülerlösung zur Relationen- und Tabellen-Aufgabe

Informatische Inhalte: Kommunizieren und Kooperieren

Neben dem Modellieren lag der Schwerpunkt bei diesem Projekt sicher auf dem Kommunizieren und Kooperieren. Welche Objekte sollte es im Spiel geben, welche Beschreibungen, und zwar jeweils einmal für die Objekte in der normalen Welt und einmal in verfremdeter Form für die gruselige Parallelwelt? In welcher Form wird mit den Objekten über das bloße Betrachten hinaus interagiert; welche Rückmeldungen soll das System dabei geben? Was soll zum Beispiel geschehen, wenn die Spielerin oder der Spieler „open the window“ eingibt, oder „jump through the window“ oder „break window“? Der Reiz dieser Spiele liegt ja nicht nur darin, eine zum Ziel führende Lösung zu finden, sondern darin, unterhaltsame Antworten auf möglichst viele Interaktionen zu erhalten. Standardmäßig folgt auf die Anweisung „break window“ die Antwort „Gewalt ist keine Lösung“, aber mit zusätzlichen Regeln wird anderes Verhalten erzeugt: „Instead of attacking the window, say "But it is your favourite

window!“ –. Eine Schülerin schreibt beim Feedback: „[I] tried everything I could find. I liked the answers of the game when you tried to do something impossible.“ Ein gutes Spiel zeichnet sich darin aus, möglichst viele solcher Handlungen vorausszusehen und originell darauf zu reagieren. Das Programmieren ist dabei oft relativ einfach, aber natürlich ist Text dazu nötig.

Deshalb wurden geteilte Online-Dokumente angelegt, in die Objekte und ihre Beschreibungen eingetragen werden konnten. Alle Schüler und Schülerinnen konnten sich beim Vervollständigen beteiligen, Abbildung 4 zeigt einen Ausschnitt aus dem Dokument, in dem Objekte, ihr Ort und ihre Beschreibung vorgeschlagen und gesammelt wurden.

| Thing | Location | Other name | Description | Other description |
|-------------------------|-----------------|------------|---|---|
| yourself | Coraline's Room | | You are a serious, resourceful girl, somewhat small for your age. | |
| pair of pajamas | Coraline's Room | | A comfortable pair of pajamas. | |
| slippers | Coraline's Room | | Ordinary slippers, made to keep your feet warm. (If you go outside, you might want to wear warmer shoes.) | |
| window | Coraline's Room | | You see trees, fields, and beyond them, on the horizon, distant grey hills. The trees look as if they were drawn in crayon by a little child. | |
| flower vase | Coraline's Room | | A blue flower vase with a shrub of nettles in it to decorate Coraline's room. | |
| shrub of nettles | Coraline's Room | | | |
| wardrobe | Coraline's Room | | The wood is of quite a dark brown colour. You like it. | The wood is of quite a dark brown colour. You like it. |
| costumes | Coraline's Room | | You don't like any of these clothes and wouldn't dream of wearing any of them. | |
| yellow wellington boots | Coraline's Room | | Classic wellies. Eminently suitable for walking around outside. | The yellow has a more greenish tint than you remember. Still, they are quite suitable for walking around outside. |
| blue dressing-gown | Coraline's Room | | Your old dressing-gown, blue, and very comfortable. | |
| bed | Coraline's Room | | It's a really old bed and it's not really big, because Caroline's mother is small. | It's a very, very big bed and has a wonderful purple color, because her other mother wants all the best for you. Because she wants you to be happy. |
| carpet | Hallway | | The carpet looks wonderful: it's an old one with a pattern. | The carpet is plain and simple. It's only black and white. |
| wallpaper | Hallway | | | |
| picture | Hallway | | The picture shows a boy in old-fashioned clothes staring at something. | When you take a better look you recognize that the expression of the boy in the picture is different - he is looking at the bubbles as if he is afraid. |
| mirror | Hallway | | At the end of the corridor hangs a full-length mirror. It was, a long time before, the inside of a wardrobe door. The mirror shows the corridor behind you. | |
| big double bed | Parents' Room | | Sometimes you go here when you are afraid or when you have a bad dream. | Somewhat, this bed doesn't look the same. |
| parents' wardrobe | Parents' Room | | It's a big blue wooden wardrobe with some handles. | It is a big blue wardrobe, made of wood. But somehow it doesn't look the same... |
| chair | Parents' Room | | | |
| parents' window | Parents' Room | | | |
| blue carpet | Parents' Room | | It's a blue carpet with fluffy fringes. Your parents' room is blue. | The carpet is blue, but not the same blue like the one of your real parents and it doesn't have those fluffy fringes. |
| desk | Parents' Room | | | |
| computer | Parents' Room | | When you turn on the computer, you have to enter a password. | This computer looks really old. If you try to turn it on, it won't work. |

Abbildung 4: Gegenstände im Spiel und ihre Beschreibungen

Die Implementierung der Objekte übernahm die Lehrkraft, allerdings hätten das unter regulären Unterrichtsbedingungen durchaus auch Schüler und Schülerinnen selbst durchführen können. (Kleinere Inform-7-Projekte mit anderen Klassen wurden bereits durchgeführt.) Kaum umzusetzen wäre das aber bei den Regeln für die Interaktion gewesen, die mehr Vertrautheit mit Inform 7 voraussetzen. Diese Ideen wurden ebenso gesammelt und implementiert, wiederum in einem geteilten Dokument wie in Abbildung 5 gezeigt.

Englisch-Inhalte

Neben den informatischen Aspekten spielten bei diesem Projekt sicher die Englisch-Inhalte die größere Rolle. Dazu gehören, hier nur kurz genannt: Die Romanvorlage auswerten und eigene kurze Texte verfassen unter Verwendung der Vorlage. Daneben Fragen der literarischen Analyse: Welche Elemente sind wichtig, welche können weggelassen werden; wie entsteht die unheimliche Atmosphäre in der Parallelwelt? Um welche Themen geht es im Roman, wie verändern sich die Figuren in dessen Verlauf?

Ideas for the game (in progress)

| Idea | Status |
|---|--|
| When you turn on the computer, you have to enter a password. Which one could it be? | Good idea. Right now, if you do turn on the computer, the type/ font (=Schriftart) changes so it looks more like an old computer game. But we could protect that with a password. Any idea which? (We could hide it in a cookbook in the kitchen... the father likes cooking.) |

Probably not - but you can still convince me!

| Idea | Status |
|--|--|
| If you try to kiss the mother: She shortly mutates to her end form, screams and goes back to her normal form as nothing happed. | Possible, but you would have to write text for it. "She shortly mutates to her end form, screams and goes back to her normal form as if nothing happened." is not a good description, because the end form is not described. |

Rejected - decided against

| Idea | Status |
|--|---|
| If you try to kiss the other mother she will be a wet Koala bear for a fewseconds. | No, because I don't understand why that would happen. |

Implemented (=done, programmed)

| Idea | Status |
|---|---|
| You need to take off your boots before getting into the bed | Done. You can't wear your slippers either - is that okay? |
| Another thing that doesn't make sence yet is that when you go to your father the discriptions points out that you annoy when you are bored, but when try to write "annoy father" nothing happens. But one thing that made me smile was that you can actually clean the floor. I really enjoyed that. I forget to mention what you can do if you try to annoy the father. "You try to get his attention by asking him stupid questions over and over. Unfortunatley he doesn't seemed to be bothered by that." | Done. |

Implemented, but unfinished - work for you, that is, you should do it

| Idea | Status |
|---|---------------------------------|
| And it would perhaps be good to have a description of the "North of the house". I know, it is not very important, but it is a possibility to improve the game a little bit. | Yes, one of you should to that! |

Abbildung 5: Ideensammlung mit Status-Rückmeldung

Ausblick

Das war ein einmaliges Projekt, das sich in dieser Form wohl nur schwer wiederholen lässt. Aber weniger anspruchsvolle Spiele, bei denen es hauptsächlich um die Modellierung der Räume und der Objekte geht und wenig um Sonderwünsche bei den Regeln lassen sich auch im Rahmen von Partnerarbeit als kleines Projekt in wenigen Doppelstunden umsetzen – wenn die Lehrkraft einigermaßen vertraut mit Inform 7 ist.

Quellen

Alle Webseiten/Links wurden zuletzt geprüft am 01.05.2023.

Gaiman, Neil: Coraline (2002). HarperCollins, New York.

Donald E. Knuth (1984): Literate Programming. In: The Computer Journal. Band 27, Nr. 2, 1984, S. 97–111.
Online: <http://www.literateprogramming.com/knuthweb.pdf>

Nelson, Graham (2022): Inform 7. <https://ganelson.github.io/inform-website/>

Rau, Thomas (2020): Coraline. By Students of Herr Rau (and Herr Rau). <https://herr-rau.de/coraline/index.html>

Rau, Thomas (2020): Coraline: Das Spiel. <https://www.herr-rau.de/wordpress/2020/06/coraline-das-spiel.htm>

Lizenz



Dieser Artikel steht unter der Lizenz CC BY NC 4.0 zur Verfügung.

Kontakt

Thomas Rau

Graf-Rasso-Gymnasium Fürstenfeldbruck

thomas.rau@ifi.lmu.de