



HPI mgzn

Hasso-Plattner-Institut

Ausgabe 23 - Wintersemester 2018/19

Erfahrungen bei der Spieleentwicklung

Neue Professoren für Digital Health

Nachhaltigkeit am HPI

Software-Umfrage





Neue Gesichter am HPI

Interview mit Prof. Arnrich und Prof. Lippert, die im Herbst 2018 ans HPI berufen wurden.

50

HPIspiele

- 4 | **Reverse-Engineering Anno 1602**
Wie modernisiert man einen Klassiker?
- 10 | **VR Gaming**
Über Bond und Bomben: Spiele für Puzzler
- 16 | **ACardShooter**
How to Develop an Indie Game

HPIkultur

- 22 | **Sprachgeschichten**
- 24 | **Lost and Found**
- 26 | **Rätsel**
- 28 | **Vom Weihnachtsschaf**
- 32 | **Der Hahn und der Wolf**

HPIintern

- 36 | **Windows, Python, IntelliJ IDEA**
Die HPIimgzn-Software Umfrage
- 42 | **Fruchtradar**
Rumlaufen, Früchte finden und essen
- 46 | **Nachhaltigkeit am HPI**
Ein Interview mit Lilith Diring
- 50 | **App statt Arzt**

HPIwissen

- 56 | **Armin empfiehlt**
Warum wir alle ins kalte Wasser springen sollten
- 58 | **Grün trifft Gold**
Wie kann ich mein Geld nachhaltig anlegen?
- 66 | **Monaden sind wie Burritos**
Eine Einführung in „Fluffy Little Things“
- 74 | **Dr. Krohns Rechtstipps**



Liebe HPI-ler, wir haben euch wieder viel zu berichten. Mittlerweile konnten wir an der Fakultät zwei weitere Professoren für den Studiengang Digital Health begrüßen und ab sofort wird Nachhaltigkeit durch den neuen Klub großgeschrieben. Im Interview berichten wir über die anstehenden Pläne.

In dieser Ausgabe folgt außerdem schon der zweite Teil von VR-Spiele-Empfehlungen, dieses Mal dreht sich alles um Puzzle-, Rätsel- und Knobelspiele. Doch was ist noch schöner als Spiele zu spielen? Spiele entwickeln! Genau das haben vier Kommilitonen in ihrer Freizeit getan – sogar mit selbst-kreierten Soundeffekten mittels Knacken von Knien. Zusätzlich hat ein weiterer Kommilitone schwelgend in Kindheitserinnerungen einen Remake von Anno 1602 entwickelt.

Zahlen, Daten und Fakten findet ihr außerdem in unserer erneut durchgeführten Umfrage zum Thema: Betriebssysteme, Software und IDEs am HPI. Wir haben diese Ergebnisse mit denen von vor fünf Jahren verglichen und konnten ein paar Veränderungen feststellen.

Wir möchten uns an dieser Stelle bei unserer fleißigen Redaktion bedanken, ein Teil ist sogar zum 35. Chaos Communication Congress nach Leipzig gefahren und hat davon auf hpimgzn.de unter anderem mit Video-Beiträgen berichtet. Außerdem konntet ihr auf unserem Blog zur Weihnachtszeit zum ersten Mal einen digitalen Adventskalender finden.

– Für den Zeitungsclub

Lisa Ihde, Tobias Markus und Lukas Wagner

Die Redaktion (Gesichter v.l.n.r.)

Joana Bergsiek
 Florian Fregien
 Moritz Schneider
 Leonard Geier
 Maximilian Stiede
 Jonas Bounama
 Lukas Wagner
 Lisa Ihde
 Jonathan Kreidler
 Noel Danz
 Lilith Diringler
 Marcel Garus
 Ulrike Herwig
 Silvan Verhoeven
 Isabell Kraus

Außerdem dabei:

Grit Fessel
 Christian Flach
 Tobias Markus
 Ronja Wagner
 Johannes Wolf



Reverse-Engineering Anno 1602

Der Schein trügt

Eine ganz normale Partie
Anno 1602? Nein, diese
Version läuft im Browser.

»Euer Volk bekommt kein Baumaterial.« – »Es mangelt an Tabakwaren.« – »Euer Konkurrent hat ein neues Schiff gebaut.« – Wer schon einmal Anno 1602 gespielt hat, der hat bei diesen Sätzen bestimmt die Stimme des Sprechers im Ohr. Der erinnert sich vielleicht auch an den Soundtrack, vor allem an Greensleeves natürlich, oder an viele Stunden des friedlichen Städtebaus. In diesem Artikel berichte ich über meine Bemühungen, Anno 1602 zu reverse-engineeren und ein browserbasiertes Remake zu schaffen.

Anno 1602 ist der erste Teil der Anno-Serie, erschien 1998 und benötigt zum Spielen mindestens Windows 95, eine 100-MHz-CPU sowie 16 MB RAM. Auf Anno 1602 folgten Anno 1503, 1701, 1404, 2070 und schließlich 2205. Aktuell wird Anno 1800 entwickelt, das 2019 erscheinen soll. Über die Jahre haben sich Systemanforderungen und Grafik zwar geändert, der Kern des Spiels ist aber immer gleich geblieben: Man startet mit einem Schiff, beladen mit ein wenig Baumaterial, zusammen mit ein paar Computergegnern in einer einsamen Inselwelt. Man gründet eine Siedlung auf einer Insel, stellt Nahrung und Baumaterialien her und baut dadurch sein Städtchen aus. Die Einwohner steigen im Verlauf des Spiels von der untersten Bevölkerungsstufe immer weiter auf. Damit Einwohner höhere Bevölkerungsstufen erreichen, dadurch mehr Steuern zahlen und größere Häuser bauen, müssen

ihre immer weiter wachsenden Bedürfnisse nach teils exotischen Gütern gestillt werden. Im Verlauf des Spiels besiedelt man weitere Inseln, handelt mit Ureinwohnern und Mitspielern und kämpft gegen lästige Piraten. Ähnlich wie bei Minecraft gibt es kein wirkliches Ziel; der Hauptspielmodus heißt treffend »Endlosspiel«.

Anno 1602 war das erste Computerspiel, das ich je gespielt habe. Ich weiß noch, wie ich mit meinem Bruder zusammen oben auf dem Dachboden saß und meine 30 Minuten Computerzeit dafür genutzt habe, ein kleines Städtchen in Anno 1602 aufblühen zu lassen. Für mich hat Anno 1602 bis heute seinen Charme nicht verloren. Das liegt neben guten Kindheits Erinnerungen vor allem auch an der mitreißenden Musik, dem komplexen, aber nicht zu überladenen Wirtschaftssystem und zu guter Letzt daran, dass es keine Lootboxen und DLCs gibt.

Schon gewusst?

Die Quersumme der
Jahreszahlen aller Anno-
Titel ergibt immer neun.

Grund genug also, heute noch Anno 1602 zu spielen. Doch wieso sollte man es reverse-engineerieren mit dem Ziel, einen voll funktionsfähigen Klon zu erschaffen? Das liegt an zwei Dingen, die mir an Anno 1602 nicht gefallen: Die Computergegner sind recht einfach. Gerade für erfahrene Anno-Spieler sind sie keine wirkliche Herausforderung. Und dann ist da noch etwas recht Triviales, das mich dennoch sehr nervt: Als Anno 1602 auf den Markt kam, gab es noch keine 16:9-Monitore. Das Spiel unterstützt eine maximale Auflösung von 1024x768. Spielt man es auf einem heutigen Monitor, ist das Spiel also nicht nur verpixelt, sondern hat auch links und rechts schwarze Streifen.

Sehr hilfreich bei meinen Recherchen waren die vielzähligen Anno-Foren und -Fanseiten. In den letzten 20 Jahren gab es schon so einige Threads, die sich um die Datenformate und die innere Struktur von Anno 1602 drehten. Ein User namens Sir Henry hat über die Jahre mehrere Tools geschrieben, die die Anno-Programmdateien decodieren und Interessantes zutage führen: Ein Tool schafft es, alle Texturen aus Anno zu extrahieren. Ein weiteres Tool erlaubt die Editierung des Spielstandformats. Dankenswerterweise hat Sir Henry den Quellcode seiner Tools Ende 2017 auf GitHub veröffentlicht.

Wichtig war mir, meinen Code unter einer Open-Source-Lizenz veröffentlichen zu können. Dafür darf er natürlich keine Copyright-geschützten Dateien aus Anno 1602 enthalten. Ich kann also nicht einfach alle Grafiken aus Anno 1602 exportieren und in mein GitHub-Repository laden. Stattdessen muss mein Remake ohne eine einzige Originaldatei starten und den Spieler dann einmalig dazu auffordern, seine originalen Anno-1602-Dateien in das Spiel zu laden. Das Spiel entpackt und extrahiert dann alles, was es aus den Originaldateien braucht und ist erst dann wirklich

spielbar. Dieser Ansatz bringt natürlich einige Herausforderungen mit sich: Während man, wenn man normalerweise ein Spiel entwickelt, in seinem Quellcode z. B. die Klassen »Getreidefarm«, »Mühle« und »Bäckerei« anlegt, muss mein Remake so aufgebaut sein, dass es erst zur Laufzeit die Liste der Gebäude aus Anno extrahiert und entsprechende Objekte instanziiert.

Zuallererst galt es die passende Programmiersprache bzw. Spieleengine für das Remake zu finden. Für mich stand von vornherein eines fest: Ich möchte kein C++ benutzen. Ich möchte nichts mit OpenGL zu tun haben und keinen Shader-Code schreiben. Ich möchte Code schreiben, der gut lesbar und einfach testbar ist und am Ende möglichst auf jeder Plattform läuft. Mein erster Versuch begann in der Spiele-Engine Godot. Sie läuft auf jeder Plattform und die Spiele können nicht nur für Desktop, sondern auch für Android, iOS und Web exportiert werden. Godot erlaubt die Programmierung in GDScript, das recht ähnlich zu Python ist. Doch ich stieß schnell an die Grenzen: GDScript erwies sich als sehr langsam, sodass der Import der Anno-Dateien mehrere Minuten unter voller CPU-Last zur Folge hatte. Zudem war es, auch aufgrund eines Bugs in Godot, schwierig, die extrahierten Bilder in Textur-Atlanten zusammenzufassen. Am Ende belegten die importierten Texturen über ein Gigabyte, während sie in komprimierter Form im Originalspiel nur etwa 20 MB groß waren.

Schon gewusst?

Ein Textur-Atlas ist eine große Textur, in der mehrere kleinere Texturen nebeneinander liegen. Das erlaubt nicht nur der GPU, effizienter auf die Texturen zuzugreifen, sondern spart am Ende auch Festplattenspeicher, da größere Bilder besser komprimiert werden können.

Pixelmüllhalden

In den Spieldaten sind auch einige Sprites enthalten, die im Spiel nicht verwendet werden.





Stille Wasser sind tief

Im Gegensatz zu Inseln wird der Ozean nicht mitgespeichert, sondern muss beim Laden der Karte aufgefüllt werden.

Schon gewusst?

TypeScript ist ein Superset von JavaScript, das nach JavaScript transpiliert wird. TypeScript bietet vor allem Typisierung und erlaubt damit bessere Autovervollständigung und die frühzeitige Erkennung mancher Fehler.

Also begab ich mich abermals auf die Suche und endete nach einem kurzen und relativ erfolglosen Versuch mit Unity schließlich im Webbrowser und bei TypeScript. Die Vorteile lagen für mich auf der Hand: Das Spiel läuft in jedem Webbrowser, braucht dadurch keine herkömmliche Installation und lässt sich mit modernen Tools und Frameworks entwickeln. Als Rendering-Engine stieß ich auf PixiJS, das einem die Kommunikation zu WebGL abnimmt. Gerade weil PixiJS keine vollwertige Spiele-Engine ist, ließ sich gefühlt einfacher mit ihr arbeiten.

Dateisystem im Browser

Wie oben beschrieben muss ein Spieler seine originalen Anno-1602-Dateien beim ersten Mal »hochladen«. Hochladen meint hier zwar, dass man im Browser auf einen »Hochladen«-Knopf drückt, aber natürlich sollen die Daten nicht an einen Server gesendet, sondern lokal im Browser gespeichert werden. Damit man die Dateien nicht bei jedem Aufruf neu hochladen muss, müssen sie im Browser des Spielers persistiert werden. Dafür wurden in den letzten Jahren verschiedene Möglichkeiten in die Browser eingebaut: Die älteste Methode zum persistenten Speichern sind die allseits beliebten Cookies. Allerdings ist die Größe eines Cookies stark begrenzt und reicht nicht annähernd für das Speichern mehrerer hundert Megabytes. LocalStorage ist eine

weitere Möglichkeit zur Speicherung, erlaubt allerdings je nach Browser nur bis zu 5 MB. IndexedDB ist eine einfache Datenbank ohne Größenbeschränkung, die in vielen modernen Browsern unterstützt wird. IndexedDB hat zwar die nötige Kapazität, ist aber immer noch eine Datenbank. Eigentlich möchte ich die Dateien aber in einer Art Dateisystem speichern. Und tatsächlich gibt es die nicht standardisierte und mittlerweile als veraltet gekennzeichnete FileSystem-API, die nur in Chrome so richtig funktioniert. Sie bietet ein isoliertes Dateisystem, das sich ansonsten aber wie ein normales Dateisystem »anfühlt«. Nach weiterer Suche fand ich ein Polyfill für die FileSystem-API, das unter Firefox IndexedDB zur Speicherung der Dateien benutzt. Mithilfe des Polyfills und einer weiteren Bibliothek zur Abstraktion lässt sich damit folgender Code im Browser ausführen, der einen Ordner anno erstellt und in die Datei 1602.txt die Zahl 42 schreibt:

```
console.table(await fs.ls(„/“));
await fs.mkdir(„/anno“);
await fs.create(„/anno/1602.txt“);
await fs.write(„/anno/1602.txt“, „42“);
```

Leider wird das Polyfill nicht mehr richtig gewartet und hat mehrere kleine Fehler. Zukünftig möchte ich daher auch in Chrome doch lieber auf IndexedDB umsteigen.

Konfigurationsdateien

Im Anno-Hauptverzeichnis befinden sich gleich mehrere Dateien mit der Endung `.cod`, z.B. `haeuser.cod` und `figuren.cod`. Öffnet man diese in einem Texteditor, erscheint nur unverständliches Kauderwelsch. `.cod` steht wahrscheinlich für »codiert« im Sinne von »verschlüsselt«. Die »Verschlüsselung« wurde aber schon vor vielen Jahren »geknackt«: Man iteriert über alle Bytes und subtrahiert den Wert jeweils von 256. Eine 42 wird also zur 214, eine 255 zur 1. Eine 0 bleibt 0, da $256 - 0 = 0b100000000$ und nur die letzten 8 Bit des Ergebnisses betrachtet werden. Hat man die Datei so entschlüsselt, muss man die Daten nur noch im Windows-1252-Encoding interpretieren.

In der `haeuser.cod`-Datei ist für jeden Bodentyp und jedes Gebäude definiert, welche Grafik zu verwenden ist, welche Baumaterialien benötigt werden und welcher Sound beim Bau abgespielt wird. Leider sind die Daten in einem mir unbekanntem Format abgespeichert. Man kann es den Entwicklern aber auch nicht übel nehmen: Das JSON-Format zum Beispiel wurde erst 2002 veröffentlicht, also vier Jahre nach Erscheinen von Anno 1602. XML entstand etwa zur selben Zeit wie Anno. Mir blieb also nichts anderes übrig, als einen Parser zu schreiben, der das Format mit Hilfe von regulären Ausdrücken in JSON konvertiert. In den Dateien werden verschiedene Objekte auf den ersten Blick ohne sichtbare Trennung untereinander definiert.

- Grundsätzlich wird in jeder Zeile einem Schlüssel ein Wert zugeordnet.
- Nummer zeigt den Beginn eines neuen Objekts an und ist pro Objekt eindeutig.
- Ein `@`-Zeichen vor einem Schlüssel bedeutet, dass der Ausdruck dahinter mit dem letzten aufgetretenen Wert dieses Schlüssels verrechnet wird.
- Die Werte können Zeichenketten, Zahlen,

Tupel oder Konstanten sein, die zusätzlich um einen festen Anteil erhöht oder verringert werden können.

- Mittels Objekt können Unterobjekte erstellt werden.
- Einrückung spielt keine Rolle.
- Kommentare beginnen mit `;`.

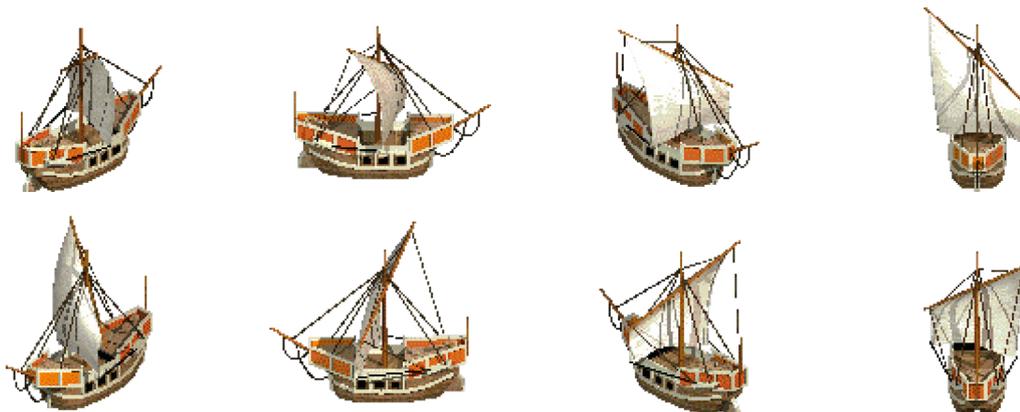
Mit Hilfe dieser Regeln ist es möglich, die `haeuser.cod`-Datei zu verstehen:

```
;---Hochschule---;
  @Nummer:    +1
  Id:         IDDIVERS+18
  ; Die Nummer der Grafik
  Gfx:       GFXSCHULE
  ; Der Sound beim Bau
  Bausample: WAV_SCHULE
  ; Größe des Gebäudes
  Size:      3, 3
  Objekt:    HAUS_BAUKOST
             Werkzeug: 6
             Holz:     5
             Ziegel:   19
             Money:    750
  EndObj;
  ; ...
;---Badehaus---;
  @Nummer: +1
  ; ...
```

Diese Dateien sind für ein Remake natürlich äußerst hilfreich, da dadurch alle wichtigen Daten zu Gebäuden einsehbar sind. In der `figuren.cod`-Datei sind zudem alle Animationen definiert. Beispielsweise ist definiert, welche Texturen die Holzfällerfigur beim Laufen, Fällen und Stehen benutzt, sowie die Reihenfolge der verschiedenen Animationen. Es wird dasselbe Format wie in der `haeuser.cod` benutzt. Die Animationen sind nicht nur kosmetisch, denn eine Holzfällertüte produziert nur dann Holz, wenn die Figur zum Baum gelaufen ist, den Baum gefällt hat und wieder an der Hütte angekommen ist.

Wenn der Wind sich dreht

Animierte Objekte müssen nicht nur für eine, sondern für bis zu acht Bewegungsrichtungen animiert werden.



So wie ich die Dateien deute, wird kein Holz produziert, wenn die Figur die Hütte nicht erreicht.

Musik

Das Abspielen der Anno-Musik ist gar nicht so einfach, wie es scheint. Die einzelnen Musikstücke sind zwar im verbreiteten RIFF-WAVE-Containerformat (.wav) gespeichert, allerdings in einem heute nicht mehr üblichen Datenformat. Viele Audioplayer spielen diese daher gar nicht ab oder die Musik klingt stark verzerrt und ist zu schnell. Nur der QuickTime-Player kommt aus unerfindlichen Gründen damit zurecht.

In den Anno-Musikdateien gibt es nur einen Kanal, weshalb Audioplayer diesen als Monospur interpretieren und abspielen. Allerdings sind in Wahrheit zwei Spuren in der Monospur kodiert: Das obere Nibble eines Bytes muss man als den ersten Kanal, das untere Nibble als den zweiten interpretieren. Heutzutage üblicher wäre die Kennzeichnung der Audiodatei als Stereo und die Speicherung der Spuren abwechselnd als ganze Bytes. Beim Import lese ich die WAV-Dateien ein, extrahiere die Audiospuren und speichere sie manuell in eine Datei, die als Stereo gekennzeichnet ist. Dadurch verdoppelt sich zwar die Dateigröße, allerdings ist die Musik jetzt im Browser abspielbar.

Schon gewusst?

Ein Nibble, auch Halbbyte genannt, besteht aus vier Bits.

Videos

In Anno gibt es 51 Videosequenzen, die das Spiel bei besonderen Ereignissen kurz unterbrechen. Diese Videos sind im Smacker-Video-Format (.smk) abgespeichert. Das Format wurde 1994 von RAD Game Tools entwickelt und war ein Vorgänger des heute verwendeten Bink-Video-Formats. Das einzige mir bekannte nicht proprietäre Programm, das mit diesen Dateien umgehen kann, ist FFmpeg. FFmpeg ist ein sehr mächtiges Kommandozeilentool, das quasi jedes Audio- und Videoformat in jedes andere umwandeln kann. Zum Glück unterstützt FFmpeg auch das .smk-Format und kann es in ein modernes Format konvertieren. Auch das ist nicht gerade trivial, da die Videos zum Teil mehrere Audio- und Videospuren besitzen. Ich habe mich dafür entschieden, die Audiospuren zusammenzuführen und alle bis auf die erste Videospur erst einmal zu ignorieren.

Doof nur, dass mein Spiel komplett im Browser laufen soll. FFmpeg ist ein in C geschriebenes Programm, das sich so natürlich nicht im Browser ausführen lässt. Oder doch? Tatsächlich gibt es das Emscripten-Projekt, eine ausgefuchste Toolchain, die es erlaubt, C(++)-Programme nach JavaScript zu kompilieren. Zum Glück war ich nicht der Erste, der FFmpeg für den Browser kompiliert hat, weshalb ich auf den Ansätzen anderer aufbauen konnte. Damit die kompilierte

Binary so klein und schnell wie möglich ist (auch für moderne Browser ist es schwierig, eine 100 MB große JavaScript-Datei auszuführen), musste ich länger mit den Compile-Switches herumspielen, um wirklich nur die unbedingt nötigen Features zu kompilieren. Die .smk-Dateien sollen gelesen und in .mp4-Dateien konvertiert werden. Dafür muss zuallererst der smacker-Demuxer aktiviert werden, der die .smk-Dateien in Audio- und Videostream aufteilt. Der Videostream wird dann vom smacker-Decoder decodiert, der Audiostream vom smackaud-Decoder. Man merkt schon, dass die Namen der Decoder eher »historisch gewachsen« anstatt wirklich durchdacht sind. Um die Audiostreams zusammenzuführen, müssen einige Filter aktiviert werden. Zum Speichern der Dateien im .mp4-Containerformat habe ich mich für das H.264-Videoformat sowie das AAC-Audioformat entschieden, da beide weit verbreitet und im Browser abspielbar sind. Zum Encodieren der Videodaten wird daher der libx264-Encoder, für die Audiodaten der aac-Encoder aktiviert. Schlussendlich werden Video- und Audiostreams vom mp4-Muxer in eine .mp4-Datei zusammengefasst.

```
emconfigure ./configure \
  [... many --disable-XYZ ...] \
  --enable-demuxer=smacker \
  --enable-decoder=smacker \
  --enable-decoder=smackaud \
  --enable-filter=scale \
  --enable-filter=aresample \
  --enable-filter=amerge \
  --enable-encoder=libx264 \
  --enable-encoder=aac \
  --enable-muxer=mp4
```

Nach etwas Hin und Her gelang es mir so, FFmpeg inklusive der libx264-Bibliothek in eine knapp 4 MB große JavaScript-Datei zu kompilieren und zur Konvertierung der

Videos im Browser zu nutzen. Emscripten kümmert sich dabei von selbst um die Emulation eines Dateisystems, sodass FFmpeg problemlos auch im Browser läuft. Eine Hürde gab es aber noch: Die Videos wurden zwar unter Chrome, nicht aber unter Firefox abgespielt. Nach einigem Suchen stieß ich in der FFmpeg-Dokumentation zu H.264 auf den rettenden Hinweis: »You may need to use [...] -pix_fmt yuv420p [...] for your output to work in QuickTime and most other players.«

Durch Angabe des -pix_fmt-Parameters war schließlich auch dieses Problem gelöst. Man könnte sich darüber streiten, wieso FFmpeg nicht standardmäßig das besser unterstützte Pixelformat benutzt.

Aktueller Stand

Unter cmfcmf.github.io/Anno2018 lässt sich der aktuelle Stand begutachten. Dort könnt ihr auch gerne eure Anno-Dateien einmal testweise hochladen. Wenn das funktioniert hat, öffnet sich das Hauptmenü von Anno 1602 und die Musik beginnt zu spielen. Es lassen sich sowohl die meisten Szenarien öffnen als auch ein gespeicherter Spielstand laden. Die Inselwelt wird bereits größtenteils korrekt angezeigt. Man kann mit der Maus umherschauen und stufenlos zoomen. Allerdings wird noch fast gar keine Wirtschaft simuliert, Gebäude produzieren keine Waren und Schiffe sind nicht sichtbar. Zudem gibt es noch einige Performance-Probleme, vor allem deshalb, weil immer die ganze Inselwelt gerendert wird, auch wenn nur ein kleiner Ausschnitt sichtbar ist (dafür müsste ich erst View Frustum Culling implementieren). Wenn Interesse besteht, kann ich sicher im nächsten HPImgzn über weitere Fortschritte berichten. In jedem Fall freue mich über jedwedes Feedback und Fragen zu meinem Remake!

– Christian Flach



VR GAMING

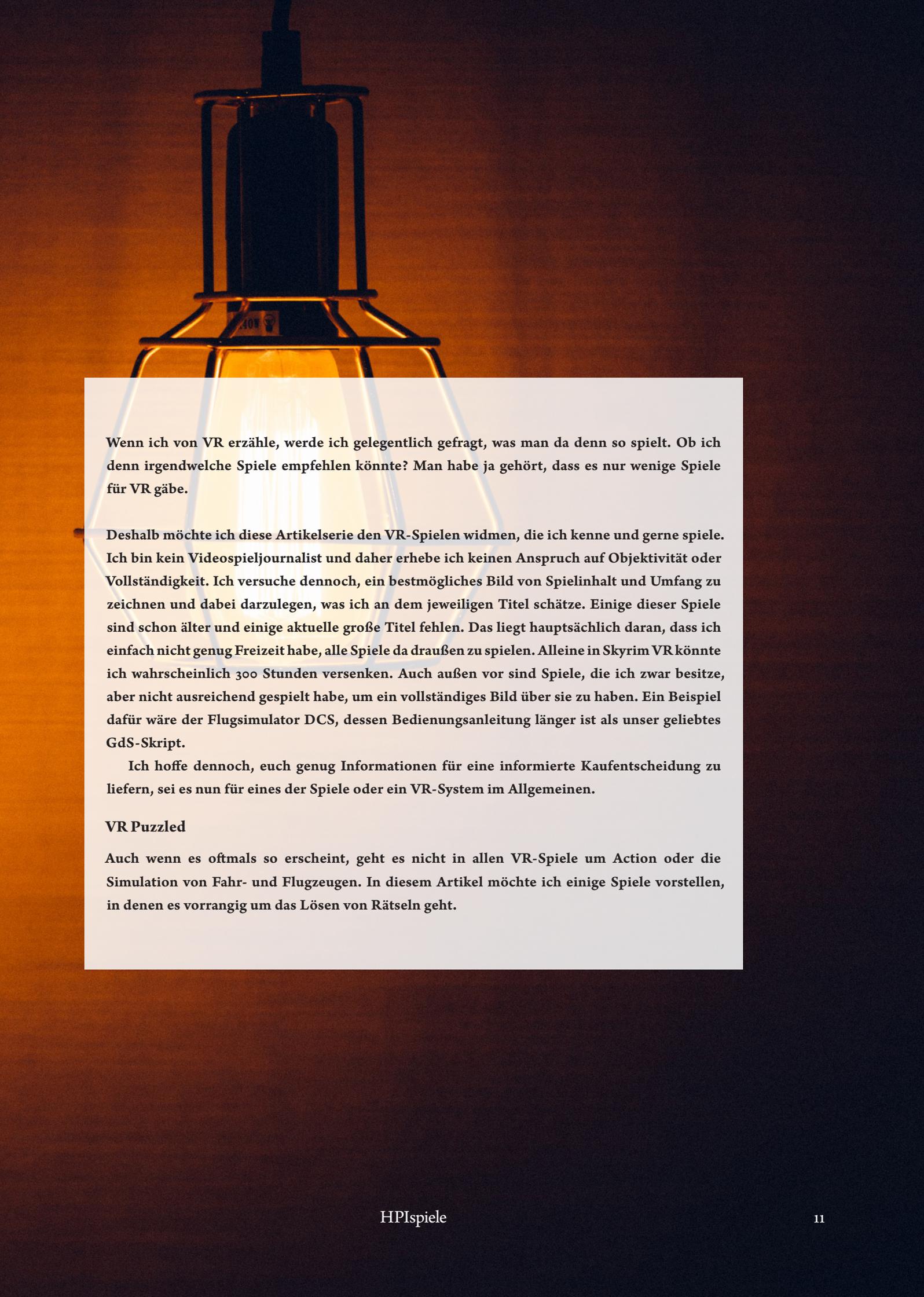
by Felix Thiel



Part 2

Puzzle





Wenn ich von VR erzähle, werde ich gelegentlich gefragt, was man da denn so spielt. Ob ich denn irgendwelche Spiele empfehlen könnte? Man habe ja gehört, dass es nur wenige Spiele für VR gäbe.

Deshalb möchte ich diese Artikelserie den VR-Spielen widmen, die ich kenne und gerne spiele. Ich bin kein Videospieldjournalist und daher erhebe ich keinen Anspruch auf Objektivität oder Vollständigkeit. Ich versuche dennoch, ein bestmögliches Bild von Spielinhalt und Umfang zu zeichnen und dabei darzulegen, was ich an dem jeweiligen Titel schätze. Einige dieser Spiele sind schon älter und einige aktuelle große Titel fehlen. Das liegt hauptsächlich daran, dass ich einfach nicht genug Freizeit habe, alle Spiele da draußen zu spielen. Alleine in Skyrim VR könnte ich wahrscheinlich 300 Stunden versenken. Auch außen vor sind Spiele, die ich zwar besitze, aber nicht ausreichend gespielt habe, um ein vollständiges Bild über sie zu haben. Ein Beispiel dafür wäre der Flugsimulator DCS, dessen Bedienungsanleitung länger ist als unser geliebtes GdS-Skript.

Ich hoffe dennoch, euch genug Informationen für eine informierte Kaufentscheidung zu liefern, sei es nun für eines der Spiele oder ein VR-System im Allgemeinen.

VR Puzzled

Auch wenn es oftmals so erscheint, geht es nicht in allen VR-Spielen um Action oder die Simulation von Fahr- und Flugzeugen. In diesem Artikel möchte ich einige Spiele vorstellen, in denen es vorrangig um das Lösen von Rätseln geht.

You're da bomb

Wer kennt es nicht? Man erwacht in einem heruntergekommenen Kellerraum und sieht vor sich, im Schein einer alten Glühbirne, eine Bombe, die es nun zu entschärfen gilt. Aber wir sind nicht allein, am Telefon haben wir einen Bekannten, der rein zufällig das Handbuch zu besagter Bombe hat. Er sieht nur nicht, was wir sehen. Also müssen wir das tun, worauf uns zahllose Soft-Skills-Kolloquien vorbereitet haben: Kommunizieren und im Team arbeiten!

Und im Grunde ist das auch schon der Kern des Spiels *Keep Talking and Nobody Explodes*. Wir durchleben das in Actionfilmen deutlich überstrapazierte Klischee einer Bombenentschärfung unter der telefonischen Anleitung eines Experten. Der Clou daran: unser Experte ist ein Freund, der Partner, die Eltern oder wen man auch sonst eingeladen hat. Die Bombe ist dabei auch etwas anders, als man aus Filmen gewohnt ist. Anstelle der Frage, ob es jetzt der rote oder der blaue Draht sein muss, besitzt jede Bombe bis zu elf Module, die jeweils eins aus elf Rätseln sein können. Alleine für das einfache »Durchtrenne einen Draht« braucht der Experte Informationen darüber, wie viele Drähte es gibt, welche Farbe sie haben und in welcher Reihenfolge sie vorkommen. Erst dann kann er bestimmen, welcher durchzuschneiden ist. Es müssen sich Wörter und Buchstaben gemerkt und Symbole und Positionen beschrieben werden. Und alles unter Zeitdruck, denn die Bombe tickt! Um das Ganze noch schlimmer zu machen, sind manche Rätsel explizit darauf getrimmt die Kommunikation zwischen Entschärfer und Experten zu stören. Aber das ist noch nicht alles, denn zu den normalen Modulen, die nach und nach entschärft werden müssen, kommen noch drei Arten von *Needy Modules*.

Diese kleinen Nervensägen lassen sich nicht entschärfen, sondern nur vorüberge-

hend deaktivieren. Sie sind zwar leicht zu lösen, können aber jederzeit wieder aufwachen. Wenn man sie vernachlässigt, detoniert die Bombe sofort. Als wäre das alles noch nicht genug, ist auch noch die Spielwelt gegen uns... die alte Glühbirne in unserem Kellerraum flackert, irgendjemand hat den Tischwecker so eingestellt, dass er im ungünstigsten Moment anfängt zu piepen und plötzliche Alarmsirenen inklusive Blinklicht sorgen für Bombenstimmung!

Kurzum, *Keep Talking and Nobody Explodes* ist oft stressig und spannend. Wer allerdings über Fehlkommunikation und eigene Patzer lachen kann und einen entsprechenden Experten zur Hand hat, wird sehr viel Spaß mit dem Spiel haben. Viele Hürden, die einem das Spiel stellt, kann man wegräumen, indem man kryptischen Symbolen Namen gibt oder sich auf bestimmte Regeln einigt, zum Beispiel, bei welcher Zahl man anfängt zu zählen.

Das Spiel gab es schon vor VR und es ist auch auf dem normalen Monitor großartig. Allerdings wird es durch seine ausgezeichnete VR-Umsetzung noch mal erheblich aufgewertet. Der Umstand, dass die Bombe wirklich nur im Headset angezeigt wird verhindert absichtliches oder versehentliches Schummeln des Experten und die Isolation der beiden Partner durch das Headset verstärkt das Gefühl, auf einander angewiesen zu sein. So geht es in *Keep Talking* nicht nur um effiziente Kommunikation, sondern auch darum in Stresssituationen einen kühlen Kopf zu bewahren. Es ist außerdem eines der besten VR-Spiele für eine Party, denn es können beliebig viele Leute mitspielen, obwohl nur ein VR-System verfügbar ist. Man kann ja nie genug Experten haben, nicht wahr?

Das Spiel ist mit der Oculus Rift, der HTC Vive und Sonys PSVR kompatibel. Es ist auf Steam, dem Sony Store und Oculus Home für 14,99 € erhältlich.

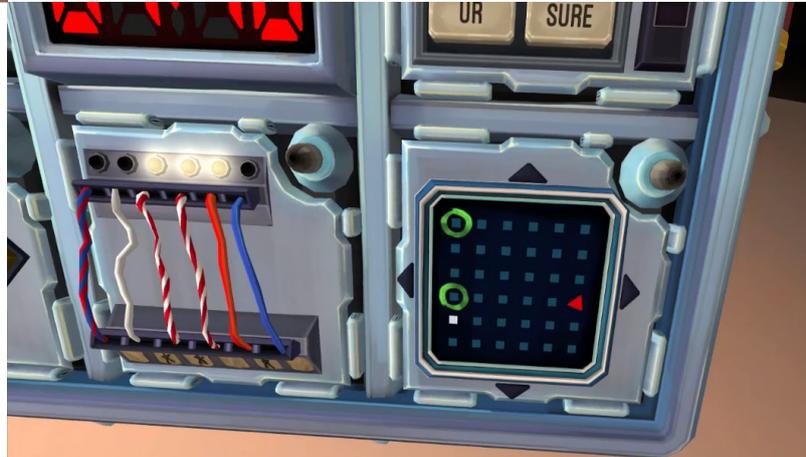


Bombenbaukasten

Auf jeder Bombe gibt es Platz für elf Module.
Fünf auf der Vorder- und sechs auf
der Rückseite.

Laborratte

Bei diesem Rätsel müssen wir das weiße
Quadrat durch das Labyrinth zu dem roten
Dreieck manövrieren. Nur leider kennt die
Wände nur der Experte und der muss durch
die grünen Kreise erstmal bestimmen welches
Labyrinth es ist.



Bobs Drucker aus der Hölle

Nein, ich weiß auch nicht warum die Bombe
einen Parallellport besitzt oder was Bob
damit zu tun hat. Allerdings ist bei einigen
Modulen die richtige Lösung von solchen
Umständen wie Labeln, Anschlüssen oder
Batterien abhängig. Also nicht ignorieren!

Bedürftige Nervensäge

Dieses Needy Module kann nicht entschärft
werden, zündet aber die Bombe wenn es
nicht regelmäßig entladen wird. Ich muss also
immer wieder wichtige Entschärfungszeit für
sein Wohlbefinden opfern.



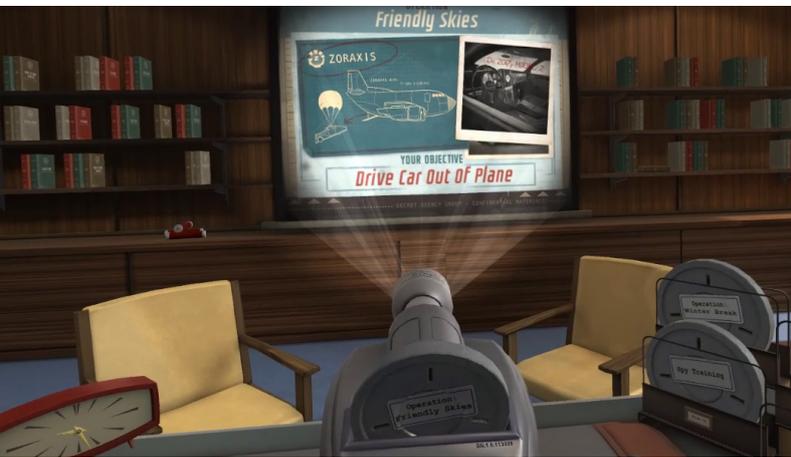
Bombenstimmung

Als würde eine tickende Bombe nicht reichen.
Kurz vor Schluss machen eine Sirene und die
rote Alarmleuchte noch mal zusätzlich Druck.

There is no such thing as overkill

Das geschmackvoll inszenierte Intro erinnert an die Opening Credits der Bondfilme und bleibt dennoch dem Humor des Spiels treu.

Säure, Haie, Laser, Kreissägen, Bohrer und ... noch eine Bombe?

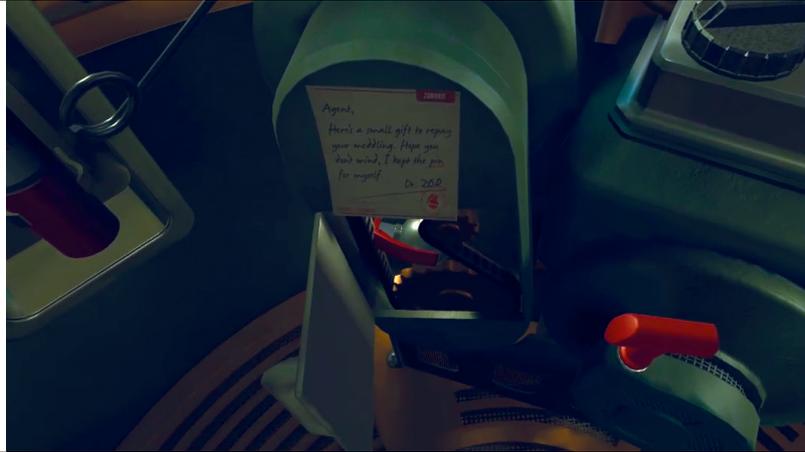


Diashow

Vor jeder Mission erhalten wir in unserem Büro eine Einführung. Natürlich auf dem modernsten Diaprojektor.

Agentenüberraschung

Der Erzbösewicht hat genug von unseren Störungen und uns kurzerhand eine Handgranate ins Getriebe geschmuggelt. Die müssen wir jetzt erstmal loswerden, bevor wir den Motor starten und fliehen können.



Aber warum sollte ich wissen wollen wie ... oh

Bei manchen der Rätseln muss man nur die richtigen Informationen gefunden haben. Bei anderen wiederum muss man um die Ecke denken oder schnell sein. Im Großen und Ganzen mangelt es weder an Abwechslung noch an Herausforderung.

Agent im Orient Express

Das im letzten Herbst veröffentlichte Bonuslevel lässt uns wie damals Hercule Poirot in einem luxuriösen Zug ermitteln.



Bonding Issues

Das Spiel *I expect you to die* kombiniert das Konzept von Exit-Games mit dem Charme alter Bond-Filme. Als Agent eines nicht näher genannten Geheimdienstes müssen wir die feindliche Organisation *Zoraxis* infiltrieren, die nichts Geringeres anstrebt als die Weltherrschaft – was auch sonst? Dabei müssen wir aus Flugzeugen und U-Booten entkommen, in Labore einbrechen, Interkontinentalraketen und Laser aufhalten und ganz nebenbei die Welt retten. Das geschieht im Verlauf von fünf kleinen Episoden, die im Schwierigkeitsgrad ansteigen. Dieser ergibt sich meistens aus dem Anspruch der Rätsel, gelegentlich werden wir allerdings auch unter Zeitdruck gesetzt.

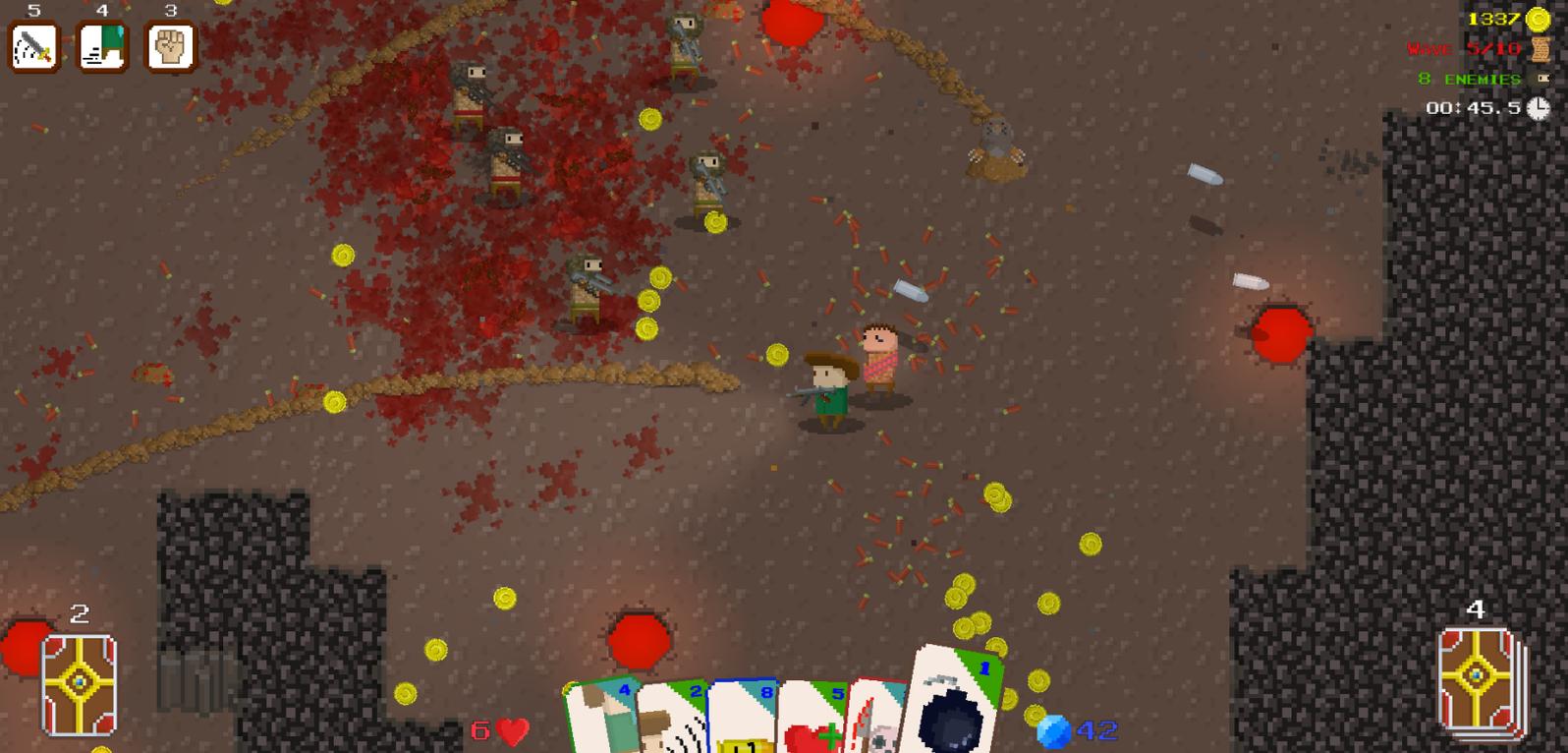
Die Gestaltung der Rätsel selbst erinnert stark an den Stil von Point-and-Click-Adventures: Wir sammeln Objekte und Informationen, um sie an anderer Stelle einzusetzen oder miteinander zu kombinieren. Die Level sind so designed worden, dass wir das Spiel komplett im Sitzen spielen können. Die meisten Objekte sind schon durch das Setting, wie zum Beispiel das Innere eines Autos oder U-Boots, in Griffweite. Für alles, was weiter entfernt ist, verfügen wir über telekinetische Fähigkeiten.

Stil und Inszenierung sind eine Hommage an die Bond-Filme des letzten Jahrhunderts. Der Titel selbst ist ein Zitat aus *Goldfinger* von 1964 und das Spiel begrüßt uns schon beim Start mit den, für die Filme typische, mehrminütigen und äußerst künstlerischen Opening Credits. Es gibt Gesang, Violinen und Saxophone. Es versprüht mit Filz und Messing den Charme vergangener Agententage, als die Bösewichte noch einfach böse, die Gadgets aberwitzig und Diaprojektoren »State of the Art« waren. Dabei ist sich das Spiel dieser Wurzeln auch stets bewusst und spart nicht an Selbstironie.

Schon im eben genannten Intro geht es um »Lasers and Sawblades with Sharks on Standby« und auch über den Rest des Spiels hinweg nimmt es sich nicht besonders ernst.

Seine einzige wirkliche Schwäche ist seine Länge. Wenn man sich geschickt anstellt, kann man die fünf Episoden in knapp drei Stunden durchspielen. Danach bleibt nur noch die Suche nach alternativen Lösungsansätzen, die es hin und wieder gibt, oder die Jagd nach Achievements und Collectibles. Es eignet sich allerdings auch wunderbar als Einführung, wenn man Freunden oder Bekannten zum ersten Mal VR zeigen möchte. Die Steuerung ist einfach verständlich und durch das Spielen im Sitzen wird das Risiko von »Motion Sickness« minimiert. Gleichzeitig ermittelt es einen guten Eindruck von der Präsenz, die VR so auszeichnet. Das Spiel unterstützt die Oculus Rift und die HTC Vive und ist auf Oculus Home (24,99 €) und Steam (22,99 €) erhältlich.

– Felix Thiel



ACardShooter

How To Develop An Indie Game

Montag 22:00 Uhr. Nachts im Wohnheim. Markus liegt auf dem Rasen und streckt sein Handy in die Höhe. Über ihm steht Erik und schwingt einen Besen hin und her. Schlag links! Schlag rechts! Drehung! Was für Außenstehende komplett irrsinnig aussehen muss, ist Teil der Arbeit und des Späßes, welche mit der Entwicklung eines eigenen Spieles einhergehen.

»Combine Two Incompatible Genres« – das war das Thema des 41. LudumDare Game Jams im April 2018 und gleichzeitig der Startschuss unseres Projektes. Zu siebt haben wir es uns im April zum Ziel gesetzt, innerhalb von 72 Stunden ein Spiel zu ebendiesem Thema zu entwickeln.

Nach mehreren Stunden Brainstorming, googlen, zufälligen Genre-Kombinationen aus Hüten ziehen (literally!) und diskutieren haben wir uns schließlich geeinigt: Eine Kombination aus Twin-Stick Shooter und Kartenspiel soll es werden. Die Idee: Man steuert in Vogelperspektive einen bewaffneten Charakter und schießt Gegner in einer Art Arena ab. Dazu setzt man Karten aus einem vor der Runde zusammengestellten Deck ein, um sich verschiedene Buffs zu geben. Die Anzahl an Karten auf der Hand ist dabei gleichzeitig die Lebensanzeige. Jedes Mal, wenn man

getroffen wird, verliert man also eine Karte. Und das Wichtigste: Dabei muss es ganz viele Screenshakes, Explosionen, Schockwellen, Screenshakes, Partikelsysteme und Screenshakes geben, damit sich das Ganze auch nach ordentlich Action anfühlt. Gesagt – Getan! In den kommenden drei Tagen schliefen wir wenig, verließen die Wohnung kaum und ließen die Uni Uni sein. Dafür haben wir aber umso mehr Mate getrunken, (meist schlechten) Code erzeugt, und Grafiken sowie Sounds erschaffen. Drei Tage später, morgens um vier, war dann endlich die letzte Zeile Code geschrieben, das letzte Pixel platziert, die letzte Runde gebalanced, sowie die Beschreibung geschrieben und abgeschickt.

Damals waren wir sieben Personen, was für ein solches Projekt eine ganze Menge ist. Während des Game Jams haben wir uns häufig dabei ertappt, dass wir mehr gespielt als

entwickelt haben. Das Spielen hat nicht nur uns, sondern auch einigen Testern richtig Spaß gemacht, weshalb sich vier von uns dazu entschieden haben, das Spiel weiterzuführen. Denn wie ihr euch bestimmt vorstellen könnt: Ein Spiel wird nicht innerhalb von drei Tagen fertig. Doch wir haben es uns zum Ziel gesetzt, das Spiel fertigzustellen und schließlich auch zu veröffentlichen!

I actually rolled my eyes when I saw your entry; a shooter with deck-building? This'll be a boring mess!

I was wrong. The running and gunning is challenging and the ability system adds quite a bit of depth to the survival struggle!

HuvaaKoodia (LudumDare-Bewertung)

Die Tage des Game Jams waren natürlich nicht genug Zeit, um durchdachte Strukturen für die Aufgabenbearbeitung anzulegen. Für Scrum und Konsorten ist während eines Ludum Dares einfach keine Zeit. Nachdem wir also drei Tage lang nach dem Motto »Hey, das könnte cool sein, ich mach das mal, mal sehen ob's den anderen gefällt« gearbeitet hatten und das Projekt dann in der heißen Phase der Bachelorarbeit und in den Ferien etwas eingestaubt ist, versuchten wir nun alles etwas besser zu strukturieren.

Wenn wir also heute finden, dass unser Spiel beispielsweise mehr Explosionen braucht (denn seien wir mal ehrlich, jedes Spiel ist besser mit Explosionen!), dann schreiben wir zuerst ein Proposal für mehr Explosionen auf.

Proposal: »needs more explosions«

Über solche Änderungswünsche reden wir dann online oder beim nächsten Treffen und überlegen in der Gruppe, ob der Vorschlag passt und wie man ihn am besten umsetzen

und integrieren könnte. In unserem Fall entschieden wir uns für eine Karte, mit welcher man eine Bombe wirft. Die Idee kann in unserem Board nun eine Spalte weiter geschoben werden und letzte Details werden hinzugefügt: Mana-Kosten und Kombinationen mit anderen Karten. Nachdem das Proposal final angenommen wurde, ist es bereit, implementiert zu werden.



Markus, Abteilungsleiter Kunst

Gibt es irgendwo wichtige und komplizierte Texturen zu erstellen, dann überlässt man das am besten Markus. Zusätzlich ist er das Oberhaupt des Entwicklungsteams, da er überhaupt erst die Idee hatte, unser Spiel weiterzuentwickeln.



Erik, Vermieter

Wie jedes Studio haben auch wir Räumlichkeiten gemietet, welche zufälligerweise genau bei ihm im Wohnzimmer liegen. Außerdem ist er neben Markus Leaddeveloper und sorgt dafür, dass man neben schönen Bildern auch schöne Effekte und Spielmechaniken bewundern darf.

Wilhelm, Leitung Konzeption

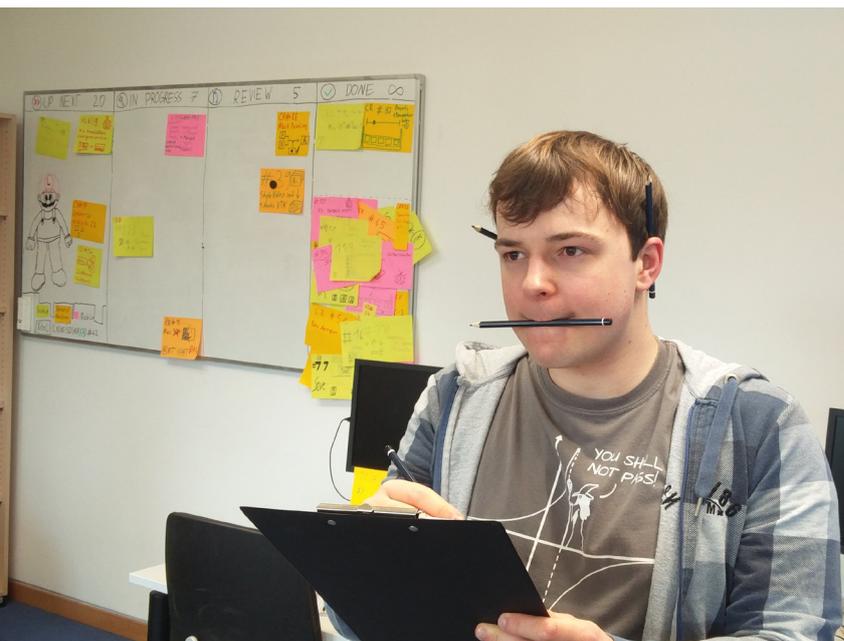
Er kümmert sich neben dem Abarbeiten der Issues darum, neue Ideen und Vorschläge in das Spiel einzubringen und ist der PR-Guru unseres Teams.



Fabian, Meister der Partikelsysteme

Wenn er nicht gerade Staub, Rauch oder Regen pixelgenau platziert, ist er auch derjenige, welcher den low-level Code schreibt, den der Rest nicht anfassen möchte.

Dann macht sich jemand aus dem Team daran, das Proposal in Code und Form zu gießen. Feste Zuständigkeiten gibt es dafür nicht – jeder nimmt sich die Aufgabe, die gerade passt. Unity kann einem dabei eine Menge abnehmen. So ist es schon vorgekommen, dass so manch' eine komplizierte Idee in fünf Minuten umgesetzt war, während manch' simple Idee Tage in Anspruch genommen hat.



Dazu kommt allerdings noch ein ganzer Haufen anderes: Bilder müssen gezeichnet, koffeinhaltige Kaltgetränke konsumiert und neue aus dem Feature hervorgehende Proposals notiert werden.

Sehr wichtig (doch häufig vergessen) sind Sounds. Wenn unsere Bombe explodiert, muss es natürlich einen Knall geben, bei dem der Nachbar lieber noch mal nachschaut, ob die Mikrowelle noch in einem Stück in der Küche steht. Dabei sind wir schon auf die besten Ideen gekommen: Für einen Charakter haben wir beispielsweise einen Axttreffer-Sound gebraucht. Nach einigem Nachforschen empfanden wir das Knacken von Markus' Knien als sehr passend und ließen ihn ein paar Kniebeugen machen, bis wir den perfekten Sound aufgenommen hatten. Woher der Sound für das Schwingen ebendieser Axt herkommt, könnt ihr euch noch einmal in der Einleitung durchlesen. Mit dem Fertigstellen des letzten Assets wird auf den Master gepusht. Durch die vielen Unity-Metadateien ist es schwer, mit Unity nicht trunkbased zu entwickeln, da andernfalls das Lösen der auftretenden Mergekonflikte an Magie grenzt. Zu guter Letzt wird in den Raum gerufen, das alle einmal pullen sollen und testen dürfen.

In der Praxis sieht das so aus: Jeden Morgen um 08:30 Uhr klopft es an der Tür zu Eriks Zimmer, welches über die Zeit praktisch zu unserem Büro geworden ist. Es gibt eine Schüssel Müsli zum Frühstück und dann geht es munter ans Werk. Einige von uns haben dieses Semester extra weniger Veranstaltungen belegt, um mehr Zeit für das Projekt zu haben. Ein nur dafür gestellter, sich wöchentlich wiederholender Wecker erinnert die manchmal tief in die Arbeit versunkenen daran, zur Uni zu gehen.

Zu so einem Spiel gehört deutlich mehr, als sich ein paar schicke Karten-Effekte auszu-denken. Eine große Menge Content muss bis



ins kleinste Detail ausgearbeitet und implementiert werden. Unter der Haube geschieht oft mehr, als dem Spieler auffällt.

Beleuchtung ist beispielsweise ein ziemlich komplexes Thema: Welche Kante wirft wann auf welche Fläche einen Schatten? Wie sieht es mit bewegten Objekten aus? Bei unserer 2,5D-Ansicht ist diese Frage nicht immer leicht zu klären. Das Errechnen und Befolgen von Pfaden ist eine Welt für sich. Die Umgebung ist in permanentem Wandel (Bäume kippen und Lava-Löcher erscheinen) und sowohl Start als auch Ziel des Pfades können beispielsweise durch Explosionen oder Rückstoß unvorhergesehen aus der Bahn gebracht werden. Das Schwierigste beim Entwickeln neuer Gegner, Level, Karten, UI und Mechaniken ist aber, dass sie intuitiv verständlich sein müssen und gleichzeitig Spaß machen sollen. Gerade die Verständlichkeit lässt sich aus der Perspektive des Entwicklers kaum einschätzen. Klar, dass man alles versteht, was man selbst entwickelt hat. Deshalb brauchen wir neben ein paar Kommilitonen, die das Spiel häufiger

spielen, auch Tester, die unser Spiel noch nie gesehen haben. In Sessions, die teilweise über zwei Stunden lang sind, lassen wir unser Spiel bis ins kleinste Detail auseinander nehmen. Alleine bei den ersten beiden Testern haben wir elf Seiten Feedback mitgeschrieben: Tex-

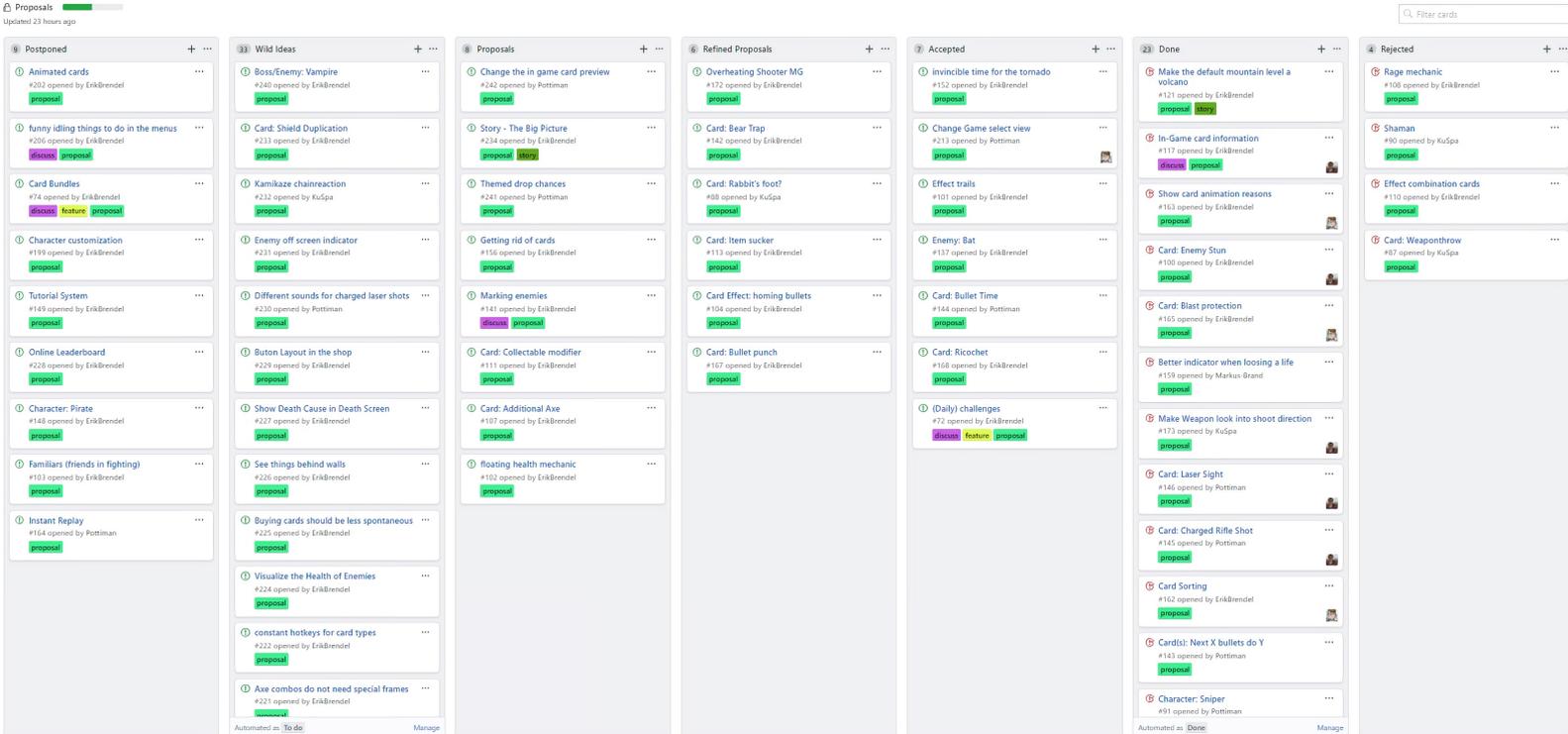


Version nach dem LudumDare (April 2018)

Im Vergleich zum Titelbild (aktuelle Version), bei welcher unter anderem die Beleuchtung, UI und Texturen verändert wurden.

Unser Aufruf nach Testern

Um genügend Tester zu finden, haben wir im Wohnheim nach Freiwilligen gesucht. Gut sichtbar vom Mikado aus, mit Schokolade als Lockmittel, sind einige unserem Aufruf gefolgt.



Proposal-Board

Hier sieht man die Menge an Vorschlägen, die wir im Team diskutieren, filtern, und feinschleifen. Bei so viel gebündelter Kreativität muss man die Dinge einfach aufschreiben, damit sie nicht im großen Strudel untergehen.

turen und Icons müssen deutlicher werden, die wichtigen Elemente müssen besser hervorgehoben werden, Soundeffekte müssen klarer und alles muss besser gebalanced sein. Ein großer Teil des Aufwands muss darauf verwendet werden, diese Dinge zu optimieren, um am Ende ein gutes Spiel zu erhalten. Gleichzeitig ist es aber auch jedes Mal wieder ein gutes Gefühl, wenn man sieht, dass ein Tester Spaß hat, oder dass ihm eines der unzähligen kleinen Details aufgefallen ist.

Jetzt haben wir eine ganze Menge über die Spieleentwicklung selbst geschrieben. Gerade als Informatiker vergisst man allerdings gerne, dass das nur der erste Schritt zum Veröffentlichlichen eines Spieles ist. Erst durch eine ganze Menge Ausprobieren und Feinschliff kann das Spiel richtig gut werden. Und selbst wenn wir dann ein Spiel haben, was gut aussieht, coole Sounds und Musik hat, Spaß macht und Wiederspielwert hat, bringt uns das alles nicht viel, wenn keiner von dem Spiel weiß. Deshalb veröffentlichen wir nebenbei schön fleißig wöchentliche Blogposts auf acardshooter.tumblr.com, twittern, und schreiben Artikel für lokale Studentenzeitungen. Aber natürlich ist es damit noch nicht getan: Marketingstrategien und Zielgruppen wollen noch identifiziert werden.

Trailer, Steam Page, Website und presskit() wollen erstellt werden. Gaming-Journalisten, Streamer und Freunde wollen angeschrieben werden. Nebenbei müssen wir uns auch noch überlegen, ob oder was wir eigentlich gründen wollen und was wir rechtlich beachten müssen. Und wo kann uns das HPI bei diesen Problemen helfen? Wir hoffen, dass wir bis zur nächsten Ausgabe des HPIimgzn Antworten auf all diese Fragen gefunden haben und euch davon berichten können. Bis dahin: stay tuned!

– Markus Brand, Erik Brendel,
Wilhelm Friedemann und Fabian Pottbäcker



Kaugummiautomat

Moritz Schneider

Sprachgeschichten

Als Informatiker beschäftigen wir uns mit Programmiersprachen, teilweise auch mit Erkennung oder Synthese natürlicher Sprache und wir setzen uns dabei mit Regeln und Spezialfällen auseinander. Aber wir nutzen diese auch täglich beim Lesen, Schreiben, Sprechen und Hören – und nehmen sie meist einfach als gegeben hin. Natürliche Sprache beinhaltet bei näherer Betrachtung viele interessante Phänomene, von denen in dieser Kolumne ein paar vorgestellt werden.

In dieser Ausgabe betrachten wir drei kleine Buchstaben, die häufig zusammen als Vorsilbe zu finden sind: *Ver* ist die häufigste Vorsilbe deutscher Verben mit nicht-trennbarem Präfix. Nicht-trennbar heißt, dass die Vorsilbe nicht vom Rest des Wortes getrennt werden kann. Möglich ist das zum Beispiel bei *vor*: Sowohl »das kann vorkommen« als auch »das kommt vor« sind korrekt. Bei Verben mit *ver* kann die Vorsilbe nicht getrennt werden.

Interessant ist bei näherer Betrachtung nun, dass viele Wörter mit der Vorsilbe *ver* ähnliche Bedeutungsfelder haben, also sprachgeschichtlich nicht einfach nur zufällig entstanden sind. Es lassen sich zwei große Gruppen von Wörtern finden: Veränderung beschreibende und negativ geprägte Wörter. Die meisten der *ver*-Verben lassen sich wenigstens einer dieser Gruppen zuordnen.

Zu den negativ geprägten Wörtern gehören unter anderem *verabscheuen*, *verbieten*, *verfälschen*, *verschlechtern* und *verzweifeln*. Wenn gleich es auch positiv geprägte Wörter mit der gleichen Vorsilbe gibt (zum Beispiel *verdeutlichen* oder *verzeihen*) überwiegt das Negative und Beeinträchtigende unter den Wörtern mit *ver*.

Die andere Gruppe besteht aus denjenigen Verben, die eine Zustandsänderung beschreiben. Das Wort *verändern* selbst, aber auch *vergrößern*, *verrücken* und *verzerrern* sind einige Beispiele. Es werden Sachen oder Personen in ihrer Beschaffenheit oder Position in einen anderen Zustand versetzt. Auch *vervielfältigen* ist möglich, so zum Beispiel *verdoppeln*.

Wenn die Zustandsänderung zur Folge hat, dass etwas später nicht mehr vorhanden oder nur schwer umkehrbar ist, passen die Wörter sogar in beide Gruppen: *verbrauchen*, *verenden*, *verlieren* und *verwüsten* sind einige Beispiele dafür.

Substantive auf *Ver* stammen übrigens in den meisten Fällen von Verben ab, die ebenfalls mit *ver* beginnen. Bei einem *Versprechen* *verspricht* man etwas, mit seinem *Verstand* vermag man in der Regel etwas zu *verstehen* und durch einen *Vertrag* einigen sich zwei Parteien auf eine Lösung, um sich in diesem Punkt zu *vertragen*.

Viele Wörter mit der Vorsilbe *ver* werden also untereinander nicht nur durch ihren gemeinsamen Wortbeginn, sondern auch durch ähnliche Bedeutungsfelder *verbunden*.

– Johannes Wolf

Verrückte Verse

So wie die Sahne auf der Torte
verzieren Silben manche Worte.

Und wenn es hinten nicht gefällt,
geht's auch dem Wort vorangestellt.

Unter den Wortanfangsgebunden
ist ver am häufigsten zu finden.

Setzt jemand vor ein Verb ein ver,
verändert sich der Sinn oft sehr.

Ob achten, passen oder stimmen,
Ein ver davor verkehrt ihr Sinnen.

Wo vorher noch viel Gutes war,
steht hinterher nur Schlechtes da.

So geht es auch bei wirken, kümmern,
ein ver versteht's es zu ver-schlimmern.

Doch nicht bei all'n Handlungsbegriffen
muss ver nur Negatives stiften.

So können loben oder trauen
auf positive Wirkung bauen.

Und auch bei bessern, stehen, senden
gewinnen positive Enden.

Ob Stimmung oder Besserung,
ein ver verschafft oft Änderung.

Lost and Found

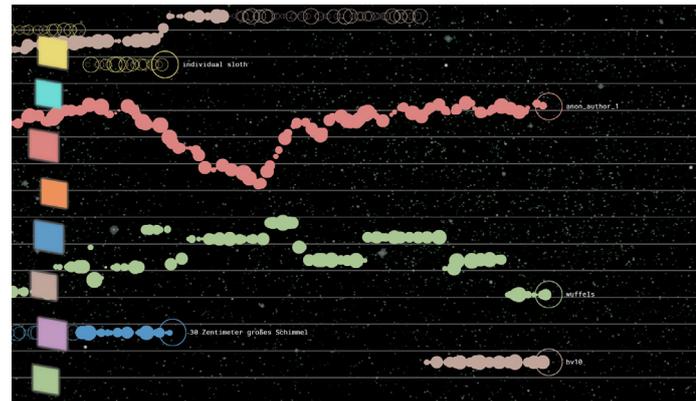
Bestimmt habt ihr schon mal eine nerdige Webpräsenz gefunden und mit Freunden geteilt oder als Lesezeichen für »später« aufgehoben. Leider geht ein solcher Fund dort viel zu schnell verloren und ist gleich darauf vergessen – deswegen halten wir hier vier der besten Web-Geheimtipps für euch fest. Dieses Mal dabei: Gruppen-Beats, nicht ganz so trockene Spieltheorie, die allgemeinen Geschäftsbedingungen und nukleare Verwüstung.

Mit Menschen musizieren

Eigentlich musiziert man entweder mit vielen Freunden in einer Band oder synthetisiert die Musik allein am Computer. Was ist aber, wenn man die modernsten Web-Technologien aus Socket-Verbindungen und Audio-API kombiniert und das Erlebnis »sozialisiert«?

Dinahmoe, eine Firma mit Expertise in interaktiven Softwareprodukten, hat dies in verschiedensten »Chrome-Experimenten« ausprobiert und in weiteren größeren Projektinszenierungen (»JAM with Chrome«, »This Exquisite Forest«) verfeinert.

Dabei kam unter anderem *Plink* heraus – ein Mehrspieler-Musikerlebnis mit intuitivem Nutzerinterface, in dem man nur mit der Maus oder per Touch in Echtzeit Musik kreiert. Man sucht sich eines von acht Instrumenten aus, welches man über 16

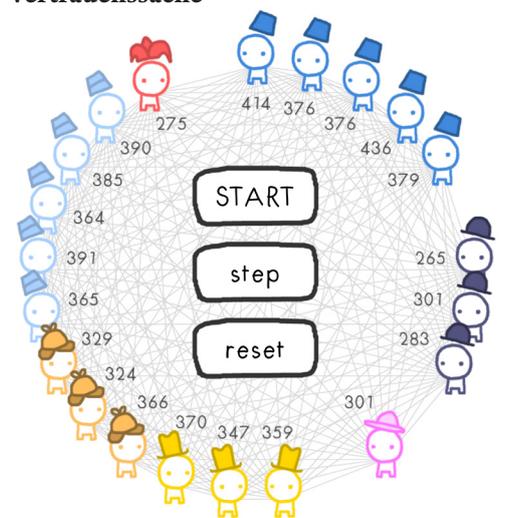


Höhenlagen auf einer pentatonischen Skala verteilt erklingen lässt. Kombiniert mit bis zu drei weiteren Mitspielern im virtuellen Musikzimmer (sowohl offen als auch geschlossen) klingt selbst zufälliges Geklimper harmonisch und macht süchtig!

labs.dinahmoe.com

plink.in

Vertrauenssache



ncase.me/trust/

Warum konnten Soldaten verschiedener Länder im Ersten Weltkrieg Fußball spielen, wenn heutzutage manche Menschen nicht mal ihren Nachbarn über den Weg trauen? Mit dieser Frage beschäftigt sich *The Evolution of Trust* von Nicky Case.

Hier ist Spieltheorie angesagt, doch statt trockener Definitionen bietet die Website eine interaktive Erklärung voller Charme. Sollte man lieber mit anderen kooperieren oder lohnt es sich eher, seine Mitmenschen auszunutzen? Lösungen bietet die Simulation zwar nicht, doch sie hilft, ein Verständnis für dieses (mehr oder weniger) moderne Problem zu entwickeln.

Nicht gelesen und trotzdem akzeptiert

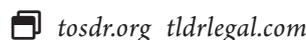
Jeder, der sich mal bei einem Online-Service angemeldet, eine App benutzt, oder etwas im Internet bestellt hat, kennt das: Man will gerade die Transaktion abschließen, doch dann wird man aufgefordert, die Nutzungsbedingungen zu lesen. Natürlich hat man gerade keine Lust und klickt auf »Weiter«. Doch wozu hat man eigentlich gerade zugestimmt? Auf welche Bilder, Informationen und Erstgeborene haben Facebook und Co jetzt Anspruch?



- Your account can be suspended and your data deleted any time for any reason
- This service provides archives of their terms of service so that changes can be viewed over time
- The service allows you to use pseudonyms
- GitHub requires cookies

Hier bietet *Terms of Service; Didn't Read* Abhilfe: Die Nutzungsbedingungen vieler verschiedener Dienste findet ihr dort einfach zusammengefasst und bewertet.

Bist du eher ein Fan von Open Source? Dann wirf einen Blick auf *tl;drLegal*, wo dir eine Menge verschiedener Lizenzen in menschenlesbarer Sprache zur Verfügung stehen, damit du wirklich weißt, was die MIT- oder GNU-Lizenz in deinem Git-Repo bedeutet.

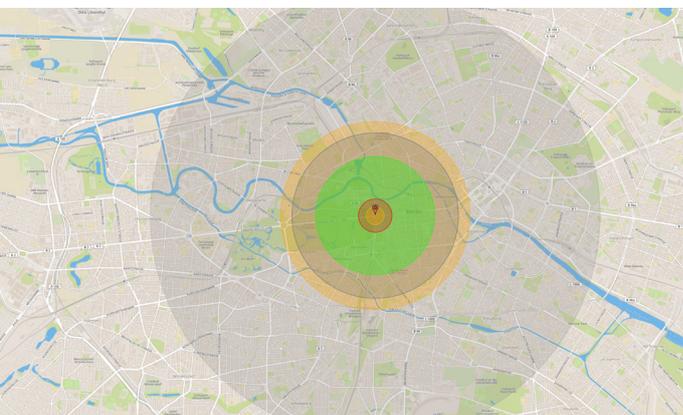


MIT License (Expat) 557742 👁️ 82 ❤️
Code License managed by kevin
5 Rules 1 Rules 2 Rules

Apache License 2.0 (Apache-2.0) 457516 👁️ 69 ❤️
Code License managed by kevin
7 Rules 2 Rules 4 Rules

Fallout-Simulator

160,5 Millionen Atombomben sind weltweit bereits detoniert – natürlich nur in einer Simulation.



Mit *Nukemap* hat der Atomwaffenhistoriker Alex Wellerstein 2012 eine Visualisierung der Dimensionen nuklearer Detonationen erstellt. Nutzer können selbst die Position und Größe

der Bombe wählen und bekommen anschließend ausgesprochen eindrucksvoll präsentiert, welche verheerende Wirkung eine entsprechende Detonation hätte.

Es ist durchaus interessant, sich etwas Zeit zu nehmen, um mit Hilfe von Presets Detonationswerte (in Kilotonnen) oder Explosionshöhe der Bombe anzugeben und die Gefahrenzonen auf der Karte zu sehen. Dabei entdeckt man erschreckende Radien von Druck, Ionenstrahlung, Thermalstrahlung, Feuerball, Krater oder Fallout, welche bei der größten Detonation selbst noch auf der globalen Weltansicht sichtbar sein können. Für besonders Interessierte bieten dazu die FAQ detaillierte Informationen zum Berechnungsmodell und zur Arbeit des Historikers.



Beiträge gesucht!

Natürlich gilt besonders auch hier: Wenn ihr selbst Vorschläge habt, so zögert nicht, diese an uns zu schicken (klub-zeitung@hpi.de) – und vielleicht hat dann die HPI-Gemeinde etwas zu schmunzeln!

Rätsel

Das schwierige Einfache

Es ist ein bewölkter Tag am Hasso-Plattner-Institut. Ein kühler aber ruhiger Wind erfasst mich, als ich auf dem Weg zum Klubraum bin. Der GDK – Geheimer Detektiv-Klub – hat es sich zur Aufgabe gemacht, im Verborgenen die Fälle und Mysterien, die am Institut auftauchen, zu lösen. Er besteht aus insgesamt vier Mitgliedern: K, M, die Chefin des GDK C, sowie seit Neuestem auch meiner Wenigkeit, F. Im letzten Semester konnte das Rätsel um das entführte Rasenmäherschaf gelüftet werden und hinter dieser Tat stand jemand, der sich selbst X nennt. Seine Identität sowie sein als nächstes geplantes Vorhaben bleiben ungeklärt. Doch da es seit dem letzten Vorfall zu keinem weiteren kriminellen Ereignis kam, muss ich den Ausdruck purer Langeweile erblicken, als ich den Klubraum öffne. M liest in einer Zeitschrift, K malt ein Bild und C hat sich in ihrem Bürostuhl zurückgelehnt, die Beine dabei auf ihrem Schreibtisch. In dieser Pose wirft sie Dartpfeile auf eine Zielscheibe, die sie selbst mit einem Filzstift an die gegenüberliegende Wand gemalt hat. Bei diesem Anblick frage ich mich ernsthaft, in welchem Klub ich da nur hineingeraten bin. Ich will versuchen die Stille zu brechen: »Ich habe zwei Flügel und kann nicht fliegen. Hab' einen Rücken und kann nicht liegen. Ich habe ein Bein und kann nicht stehen. Trag' eine Brille und kann nicht sehen. Was bin ich?« C seufzt schwer. Sie wirft unbeirrt weiterhin Dartpfeile auf die Wand und sagt: »Nase«. Ich will mich noch nicht geschlagen geben: »Was geht am Morgen auf zwei Beinen...« »Mensch!«, stoppt C mich erneut und zeigt mir mit ihrem Blick, dass sie genervt ist. Leicht enttäuscht schnappe ich mir eine Zeitschrift und setze mich neben M. Einige Stunden der Langeweile später ist plötzlich ein Klingeln im Raum zu vernehmen. Alle sehen zu M, er geht an sein

Handy: »Ja? ... Ja. ... Wirklich? ... Okay, ich melde mich wieder.« Er legt auf und holt kurz Luft, ehe er sich an uns wendet. »Das gerade war ein Freund von mir, der hier an einem Fachgebiet arbeitet. Heute Morgen geschah es, da wurde der Rechner eines anonymen Dozenten mit einer Ransomware infiziert. Das Besondere dabei ist, dass der Erpresser kein Geld fordert. Es soll aber möglich sein, durch Eingabe eines Schlüssels die Daten wieder zu dechiffrieren. Außerdem soll sich auf dem Schreibtisch des Dozenten ein Briefumschlag befunden haben, auf dem nur ein X steht.« C wird hellhörig, man merkt, dass sie es sich nicht ansehen lassen will, doch jeder erkennt das Leuchten in ihren Augen. Sie wirft den letzten Dartpfeil an die Wand, ohne dabei hinzusehen – Volltreffer. Sie nimmt die Beine vom Tisch, steht auf und zieht sich ihre Handschuhe und ihren Mantel über. Zielstrebig geht sie zur Tür und fragt in den Raum: »Worauf wartet ihr? Wir haben einen Fall zu lösen!« Alle springen auf und folgen ihr enthusiastisch zum besagten Fachgebiet. Dort im Büro des Dozenten angekommen, strahlt uns ein bedrohlich aussehender Screen entgegen, der mitteilt, dass der Rechner gehackt wurde und zum Entschlüsseln der passende Schlüssel eingegeben werden muss. Der Briefumschlag enthält einen klassischen Erpresserbrief, darin sind sieben Instruktionen, die jedoch wiederum verschlüsselt sind. Als C den Umschlag genauer untersucht und ihn umdreht, fallen kleine Schnipsel heraus. Auf diesen stehen verschiedene Verschlüsselungsmethoden und wohl zugehörige Schlüssel...

Hilf' dem GDK und versuche das Rätsel zu knacken! Sende uns den Schlüssel bis zum 31.03.2019 an gewinnspiel@hpimgzn.de und gewinne mit etwas Glück einen Gutschein für Ulfs Café.

– Florian Fregien

1. $GGVVA \ XAAGf \ FVDFg \ GGGfA \ FxfgA \ FAXGv \ VxAGf \ VxVfV \ XfXVA \ AVAXG \ AfGxV \ FfVxf \ GxVfV \ XxGvf \ XDGvV \ VxfFX$
 $FAlfD \ GGxVx \ XxGAX \ ADAfV \ VxXGG \ VgffX \ VfgAG \ AXAGG \ GGGVD \ GxxGx \ DAXXG \ DGAfG \ DfADA \ FGGAX \ XGDGG \ FGGfG \ FgXVg$
 $FDXDD \ GvDgX \ FvVdX \ FfGGv \ XxfVd \ FdgfX \ GxVdX \ XxAXx \ DxxAG \ AgfDx \ DfXXX \ FxxxV \ FADfX \ AXfVx \ GAAAX \ FGDGD \ GGG$

2. $ABC \ ABhM \ PmOvL \ DkWB \ JfRQB \ RBOB \ AFB \ PQRAB \ QBhIR \ YP \ YBF \ ABC \ BOPOB \ SUIBOPXJJIR \ F \ AFBPJ \ PBjBPQB$

3. $D7 \ AB \ 2 \ 95 \ DB \ 1 \ B \ 45 \ 3 \ C \ 7B \ 6 \ 55 \ 57 \ 2 \ 6D \ EE \ 4A \ 26 \ 52 \ B \ 93 \ 2 \ 3 \ 15 \ 95 \ 76 \ 34 \ CB \ DD \ 54 \ 42 \ 8D \ DE \ BE$
 $67 \ BE \ 64 \ CD \ 8 \ 1 \ EC \ AD \ 46 \ D \ 8B \ 85 \ 27 \ A8 \ 74 \ D8 \ F4 \ 16 \ 9 \ CA \ E8 \ 48 \ 3 \ E \ C6 \ 6E \ 8 \ 7C \ D6 \ A5 \ 8 \ 18 \ 4A \ 32 \ 9D \ 77$
 $BA \ 5 \ C \ 2 \ F \ 9C \ B8 \ 6E \ F \ 85$

4. $OWIDY \ gFISW \ LYRR \ JSFOY \ LDOYL \ IXKYB \ DMCPR \ PBPFSO \ gKI$

5. $110T8iv_{32} \ 1kAGIE_{32} \ 1TE508P_{32} \ 2GN7RFM_{32} \ 10NK4CN_{32} \ 1TE508P_{32} \ VLU2St_{32} \ 1TMRNDN_{32} \ 1OL6MH2_{32} \ 1TE508P_{32} \ IH3889_{32} \ 1kAGIE_{32}$
 $1TE508P_{32} \ 17v690g_{32} \ 614HP3_{32} \ 1KJLTLR_{32} \ 1TMRNDN_{32} \ 1MgKEA8_{32} \ 1TE508P_{32} \ 20VSCQ9_{32} \ 18BGM30_{32} \ ATAFAS_{32} \ G3PN65_{32} \ 30AEDC_{32}$
 $1TE508P_{32} \ 1CU78B_{32} \ 6UFJQ1_{32} \ 2782AF0_{32} \ 1kAGIE_{32} \ 1TE508P_{32} \ 1TMRNDN_{32} \ 1TE508P_{32} \ 1MgKEA8_{32} \ 30AEDC_{32} \ 1TMRNDN_{32} \ 24DSOEL_{32}$
 $614HP3_{32} \ 1MgKEA8_{32} \ 1CU78B_{32} \ 1TE508P_{32} \ 30AEDC_{32} \ IH3889_{32} \ 1KJLTLR_{32} \ 1OL6MH2_{32} \ 30AEDC_{32} \ 6NS06k_{32} \ 18BGM30_{32} \ 30AEDC_{32}$
 $14G0GE7_{32} \ ATAFAS_{32} \ 1MgKEA8_{32} \ 10NK4CN_{32}$

6. $nBBs \ ke \ knegn \ d \ n \ e \ a \ m \ k \ o \ i \ t \ d \ u \ t \ e \ z \ n \ n \ l \ s \ e \ a \ n \ h \ e \ m \ w \ s \ o \ u \ v \ e \ d \ e \ s \ d \ a \ i$
 $n \ e \ m \ s \ u \ a \ e \ w \ a \ i \ v \ s \ s \ i \ s \ b \ e \ d \ e \ d \ e \ t \ e \ n \ o \ c \ z \ u \ n \ h \ e \ z \ u \ m \ k \ e \ f \ e \ z \ u \ b \ e \ n \ e$
 $e \ l \ o \ t \ e \ e \ e \ n \ g \ e \ p \ p \ n \ c \ a \ n \ s \ t \ p \ n \ a \ z \ u \ m \ n \ l \ d \ e \ t \ u \ t \ i \ n \ t \ . \ d \ e \ e \ m \ u \ n \ e \ d \ e \ s$

7. $TBAUR \ GHMFS \ OXVSJ \ GT \ DH \ FEHBJ \ GKQYV \ EWWJ \ NMUIF \ MBEBP \ WQYKW \ JGHWK \ WWDWG \ DS \ DJKHK$
 $HBTNA \ MSMPD \ G \ VOXKE \ DGXUB \ LIHDV \ FYJHZ \ B \ D \ D \ N \ RQXWX \ CLAF \ NDOXP \ D \ Q \ D \ J \ P \ GNFJU \ IHIL$

Das Produkt aller Zahlen zum Quadrat ist der gesuchte Schlüssel.

Verschlüsselung mit CBC und Nullen-Padding

AES

Enigma

$m = 310LID9_{32}, e = BV_{32}$

III-G, I-D, V-K, UKW:C,
AE BM CF DK GV HO
IU NQ, HPI

Caesar

R-6, A+14, E+0, T-1, S+2, E+9, L-5

Vigenere

Railfence

69 = <T,O> Cantor

RSA

ADFGVX

$\max\{x \mid 0 = -x^6 - 12x^5 + 432x^4 + 7286x^3 + 28137x^2 + 36198x + 14904\}$

$$1. S = \begin{pmatrix} 82 & 89 & 75 & 85 & 76 & 196 \\ 81 & 65 & 67 & 220 & 95 & 73 \\ 60 & 63 & 78 & 87 & 71 & 88 \\ 74 & 70 & 46 & 68 & 32 & 83 \\ 66 & 86 & 90 & 80 & 79 & 62 \\ 84 & 44 & 72 & 69 & 47 & 77 \end{pmatrix}$$

2. $T = \text{Spur}(S)$

Die Geschichte vom Weihnachtsschaf

Das Schaf auf dem Rasenmäher ist längst inoffizielles Maskottchen am HPI geworden. Und seit Mai wird es auch auf Telegram als Sticker in vielen verschiedenen Posen verwendet. Doch woher kommen diese Schafe eigentlich?

Es war einmal in einem sehr schafigen Land, da wurde zur Vorweihnachtszeit ein kleines Schaf geboren. Sein liebster Besitz war eine rote Mütze mit flauschigem, weißem Rand und einer eben solchen Bommel. Außerdem hatte die Mütze ein kleines Geweih. So konnte das kleine Schaf so tun, als wäre es ein prächtiges Rentier, das zu Weihnachten den großen Zauberschlitten von Stall zu Stall zog.

Eines Tages allerdings geriet unser kleines Schaf in ein Dilemma. Beim Spielen mit der Weihnachtsdekoration seiner Familie verhedderte es sich mit seinem prächtigen Geweih in einer recht bunten Lichterkette. Da erschrak es erst einmal sehr. Aber dann erinnerte es sich daran, dass seine Familie immer darauf achtete, schaf-sichere Elektronik zu kaufen. Das kleine Schaf war nicht in Gefahr, denn die Lichterkette würde sein schönes oranges Fell und seine Lieblingsmütze nicht versehentlich anzünden. Als der Schreck überwunden war, trappelte das kleine Schaf zum nahegelegenen Spiegel und sah zu seinem Erstaunen ein fröhlich buntes Getümmel auf seinem Kopf. Das gefiel ihm so gut, dass unser Schaf beschloss, von nun an auch die bunte Lichterkette so zu lieben, wie es auch seine Mütze liebte. Seine Familie war verzückt von der Fröhlichkeit, die das kleine Schaf versprühte. So beschlossen sie, es aufgrund seiner Lieblingsgegenstände und seines Traums vom Geschenkesschlitten von nun an »Weihnachtsschaf« zu nennen.

Und sie lebten glücklich viele, viele Weihnachtsfeste lang.

– Ende –

Naja, oder zumindest spielt es sich in etwa so in meinem Kopf ab. Bunt und flauschig. Das kleine Weihnachtsschaf habe ich im Dezember gemalt, sodass es der FSR für die Weihnachtsfeier verwenden konnte.

Und so entsteht ein neues Schaf in meiner kleinen flauschigen Stickerwelt:

Als erstes gehts in die Brainstorming-Phase. Welches Thema hat mein neues Schaf? Welche Accessoires und Posen könnten helfen, dieses Thema zu vermitteln?

Im Falle des Weihnachtsschafs war die erste Idee eine schöne rote Weihnachtsmütze. Natürlich mit Bommel. Da ich mir aber nicht sicher war, wie die Mütze aussehen soll, gab es erst mal sechs Entwürfe. Ich zeichne meine ersten Skizzen immer per Hand mit Bleistift oder einem gut radierbaren Buntstift, da ich analog flexibler und kreativer bin als digital.



Diese Zeichnungen wurden dann von meinen fleißigen Schafkritikern begutachtet – ich habe da eine Telegram-Gruppe für Sticker-Entwürfe. Die kleine kurze Mütze (Nummer 2) wurde auf den Ideenstapel für das Winterschaf zurückgelegt.

Als klar war, dass das neue Schaf eine lange Mütze bekommen sollte, ging es an die zweite Phase des Brainstormings. Ich experimentiere dann gerne mit unterschiedlichen Posen und weiteren Accessoires. So entstanden diese Skizzen:



Beliebteste Kandidaten hier waren das Rentier-Schaf und das Schaf mit der Lichterkette. Naja, gesagt, getan und schwups wurde ein kleines Rentier-Schaf gezeichnet, das eine Lichterkette im Geweih hatte.



Nach allgemeinem Knuffigkeits-Bekunden meiner Freunde habe ich die Zeichnung dann digitalisiert. Dafür benutze ich die kostenlose Malsoftware Krita und mein Surface-Tablet. Ich füge ein Foto meiner Skizze in einer neuen Ebene des digitalen Bildes ein und erhöhe die Transparenz, damit ich besser drübermalen kann.

Anschließend male ich digital die schwarzen Linien, auch Outlines genannt. Für das Weihnachtsschaf gab es mehrere Ebenen, da ich mit unterschiedlichen Geweihen experimentiert habe. Ebenen erleichtern die Arbeit enorm. Falls euch das Konzept nicht bekannt ist, stellt es euch wie einzelne Plastikfolien vor, auf denen ihr etwas malt. Diese legt ihr dann übereinander, um das Gesamtbild entstehen zu lassen. So könnt ihr Details leichter anpassen, ohne die Gesamtarbeit zu verändern und könnt schneller zwischen unterschiedlichen Ideen und Entwürfen hin und her wechseln.



Sind die Outlines von meiner lieben Schaf-Gruppe abgesehen, geht es weiter zum Kolorieren. Auch hier verwende ich natürlich wieder neue Ebenen. Ich lege zuerst die Grundfarben der Flächen fest, also ohne Schatten und belichtete Stellen, und schaue, dass alles soweit stimmig ist. Das geschieht allerdings nach Bauchgefühl und nicht nach den Regeln der Farbtheorie.



Für die Schattierung verwende ich neue Ebenen in einer sogenannten Clipping Group mit den Ebenen der dazugehörigen Grundfarben. Also habe ich dann zum Beispiel eine Gruppe »Fell« mit den Ebenen »Grundfarbe«, »Schatten« und »Highlights«. Die Clipping Group ermöglicht es mir, nicht über die Linien zu malen; ich muss also nicht darauf achten, dass die Farben perfekt an den Outlines enden und ich kann nicht aus Versehen andere Flächen übermalen. Das resultierte dann in diesem Bild:



Nach dem Einfärben der Glühbirnen ging es noch an die Lichteffekte, um die Lichterkette spannender zu gestalten. Dafür habe ich einfach ganz viele Gradiente genutzt, um einen Schein um die einzelnen Birnen herum zu legen. Das Ergebnis seht ihr rechts.

Um daraus nun für euch einen Sticker bei Telegram zu machen, füge ich immer noch einen weißen Rand außen hinzu und skaliere dann das Bild herunter auf die 512x512 Pixel, die verlangt sind. Und dann ist es nach schnellem Hochladen auch schon geschafft. Ein neuer Sticker ist im Pack angekommen und kann von allen Telegram-Nutzern verwendet werden.

– Lea Gerneth



Der Hahn und der Wolf

Ein eitler Hahn begrüßte wie jeden Morgen krähend den Tag und freute sich, dass alle Tiere und Menschen auf seinen Ruf hin mit der Arbeit begannen. Er stolzierte auf seinem Misthaufen auf und ab und dachte: »Wenn ich es schaffen würde, von weiter oben zu krähen, könnte man mich noch weiter hören. Dann könnte ich noch mehr Leuten mitteilen, wann sie mit der Arbeit beginnen müssen.« Das würde seine Hennen bestimmt beeindrucken.

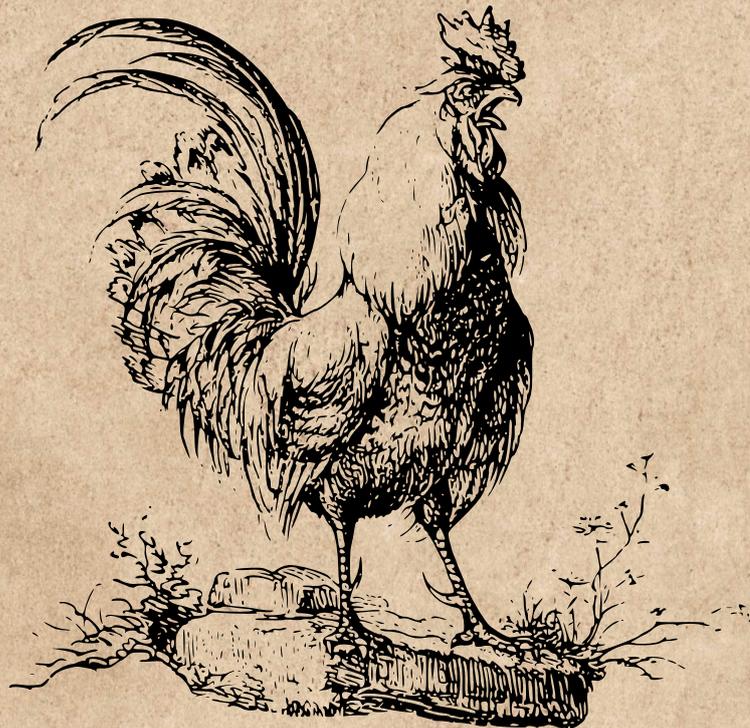
Er machte sich gleich ans Werk. Seine Hennen ließ er ununterbrochen neues Material herantragen. Von früh bis spät arbeitete der Hahn an seinem Misthaufen und vergrößerte ihn Tag für Tag. So sehr freute er sich auf seinen neuen Haufen, dass er alles um sich herum vergaß. Die anderen Tiere auf dem Hof wunderten sich sehr über den Hahn.

Doch es konnte dem Hahn nicht schnell genug gehen. So sehr sich seine Hennen auch anstrebten, immer ging es ihm zu langsam voran. Deshalb ließ er nach dem Wolf schicken, der sicher viel größere Lasten tragen könne.

Der gierige Wolf erkannte seine Chance und war nur zu gerne bereit, den Hahn zu unterstützen: »Ich helfe dir bei deiner Arbeit und überwache auch, dass alle ordentlich mitarbeiten, damit wir schnell vorankommen. Erlaube mir daher bitte, nachts im Hühnerstall zu schlafen, damit ich morgens gleich vor Ort bin.«

Der Hahn dachte sich nichts Böses und versprach dem Wolf, ihm einen Schlafplatz einrichten zu lassen.

In den nächsten Tagen ging die Arbeit gut voran. Der Hahn war stolz auf den



Fortschritt und ganz auf seine Arbeit fokussiert. In jeder Nacht jedoch verliebte sich der Wolf eine der Hennen ein, um seinen ewig währenden Hunger zu stillen.

Als der Hahn eines Tages bemerkte, dass nur noch wenige Hennen zugegen waren, fragte er den Wolf, warum die anderen nicht zur Arbeit erschienen seien. Der Wolf log: »Du hast deinen Hennen zu viel zugemutet. Sie sind erschöpft von der Arbeit und müssen sich etwas ausruhen, werden aber bestimmt bald wieder bei Kräften sein. Lass uns so lange weiterbauen.« Der Hahn war beruhigt und konzentrierte sich wieder auf seinen Misthaufen.

Nach mehreren Wochen war der Hahn endlich zufrieden. Voller Stolz krächte er am Morgen von seinem neuen großen Haufen. Er krächte einmal, er krächte zweimal, doch keine

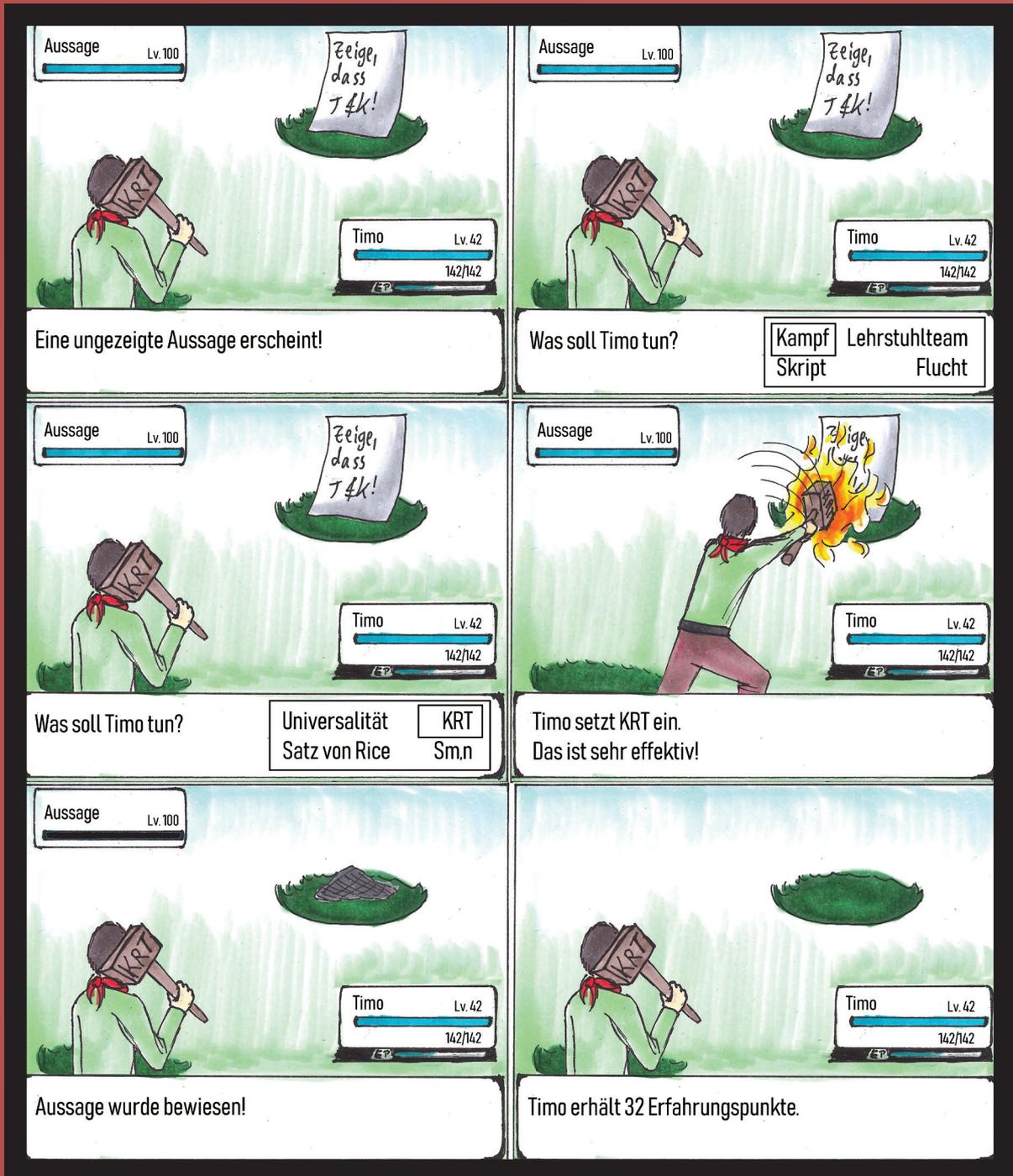
einzigste Henne ließ sich auf dem Hof blicken. Auch der Wolf war nirgends zu sehen. Der Hahn stieg vom Misthaufen hinab und begab sich in den Hühnerstall. Dort entdeckte er, was der Wolf angerichtet hatte. Keine Henne war verschont geblieben, nur ein paar Federn hatte der Wolf zurückgelassen.

Der böse Wolf in seinem Bau aber lachte über den Hahn, der nun einen großen Misthaufen hatte, aber keine Hennen mehr, die er damit beeindrucken konnte.

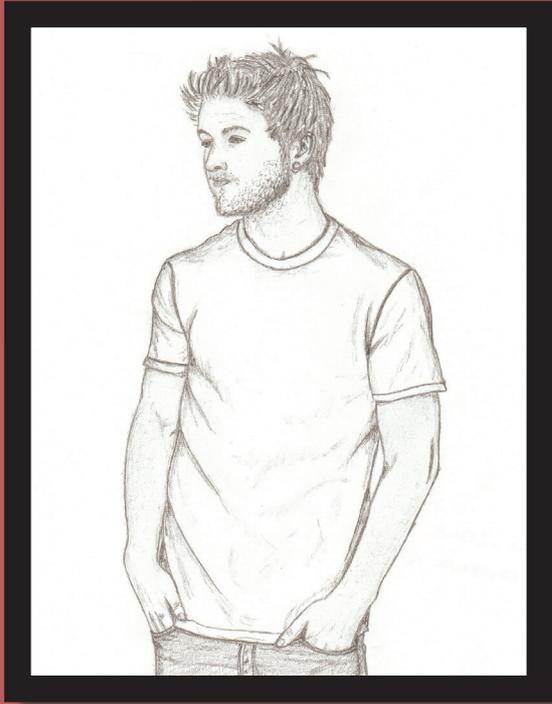
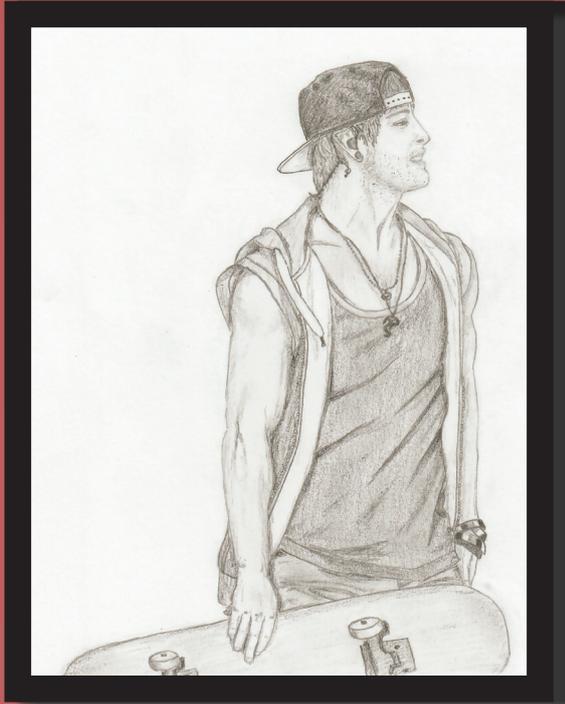
Durch Hast und fehlende Umsicht in der Planung können die eigentlichen Ziele schnell aus den Augen verloren werden.

– David Hahn und Johannes Wolf



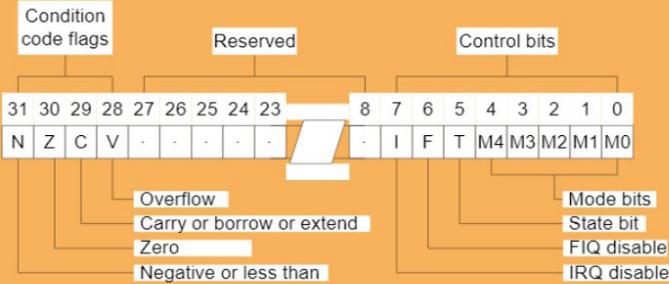


Timo und der KRT-Hammer
 Andrea Nathansen



baby, du musst ein supervisorbit sein, denn bei dir verliere ich einfach die kontrolle

to:
from:

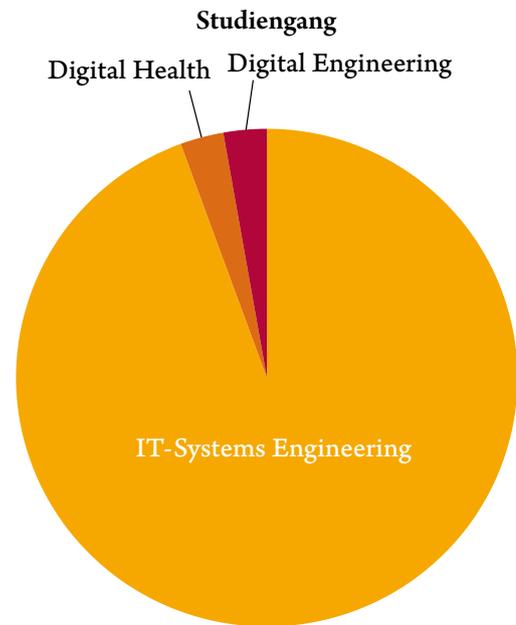
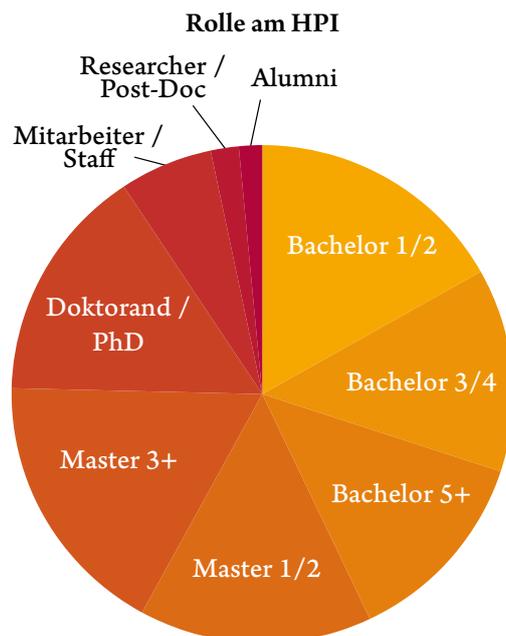


Portraits und Karte
Anonyme Künstler

Windows, Python, IntelliJ IDEA

Vor fünf Jahren, in Ausgabe 15 – unsere langjährige Leserschaft wird sich sicherlich noch daran erinnern –, haben wir die Vorlieben der HPI-Studierenden zu den Themen Betriebssysteme, Programmiersprachen und Software untersucht. Nun ist es wieder soweit: Wir haben gefragt und über 300 von euch haben geantwortet.

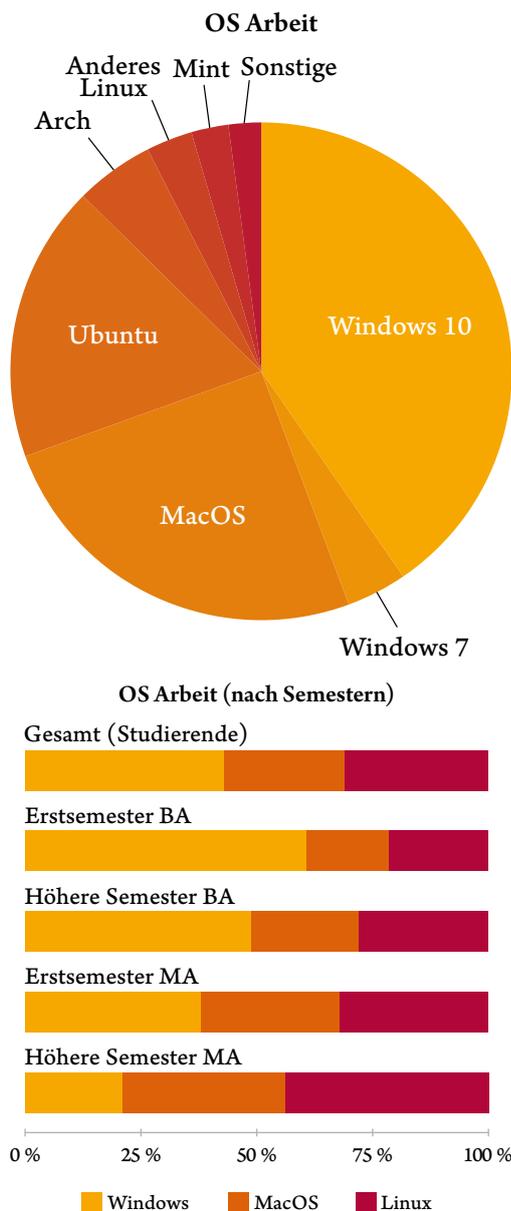
Zuerst ein riesiges Dankeschön an alle, die sich die Zeit genommen haben, an der Umfrage teilzunehmen – ohne euch wäre dieser Artikel nicht zustande gekommen. Dabei stellt sich die Frage: Wer seid ihr eigentlich? Wie setzen sich die 334 Antworten zusammen? Fangen wir mit den Bachelor-Erstsemestern an: 56 Stimmen ist schon eine beachtliche Quote. Doch auch die älteren Jahrgänge sind gut repräsentiert. Sowohl die Drittsemester als auch Studierende ab dem fünften Semester haben jeweils über 40 Mal abgestimmt. Ähnlich engagiert zeigen sich auch die Studierenden im Master: Es gibt 50 Antworten von Master-Erstsemestern und sogar 58 Antworten aus älteren Jahrgängen. Werden diese 251 Stimmen nach Studiengängen aufgeteilt, entfallen übrigens 237 auf IT-Systems Engineering und jeweils 7 auf Digital Health und Data Engineering.



Doch nicht nur die Bachelor- und Masterstudierenden haben fleißig abgestimmt, auch die Doktoranden sind mit über 50 Antworten vertreten. Ein ganz besonderer Dank gilt weiterhin den 20 Mitarbeitern, die sich, obwohl die Umfrage teilweise wenig Rücksicht auf sie genommen hat, tapfer durchgeklickt haben. Und nicht zuletzt seien an dieser Stelle die sechs Researcher und Post-Docs, sowie die fünf Alumni erwähnt, die teilgenommen haben. Lange Rede, kurzer Sinn: Vielen Dank an alle, die geantwortet haben!

Betriebssysteme

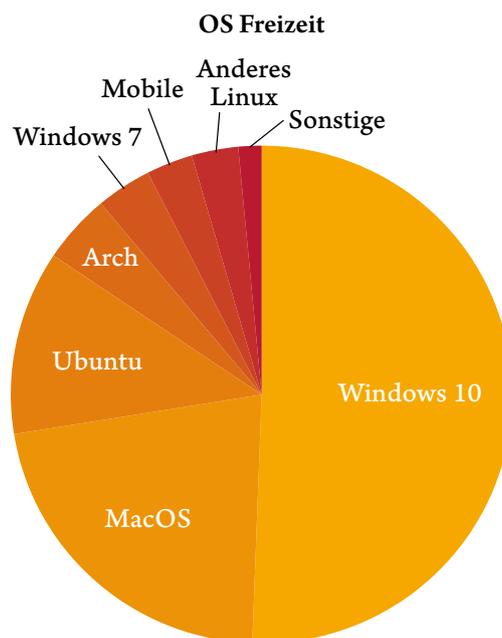
Ein ewiger Streitpunkt sind Betriebssysteme: Bietet Windows eine breitere Auswahl an Tools? Ist MacOS besser zum Programmieren geeignet? Nutzen wahre Programmierer nur Linux? Es hat sicherlich jeder einen Grund, sich für ein Betriebssystem zu entscheiden,

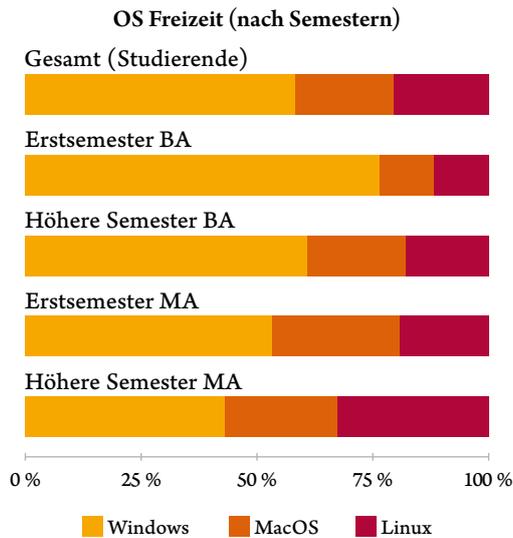


doch an dieser Stelle sollen keine Vor- und Nachteile diskutiert, sondern lediglich die harten Fakten ausgewertet werden. So ist Windows mit knapp 45 Prozent weiterhin das vorherrschende Betriebssystem vor Linux mit 29 Prozent und MacOS mit 25 Prozent. Hier fällt auf, dass Windows und MacOS seit 2013 an Marktanteil verloren haben – damals waren es noch 55 bzw. 30 Prozent. Linux hat dagegen seinen Anteil von ehemals 15 Prozent fast verdoppeln können und ist damit vor MacOS auf den zweiten Platz gerückt.

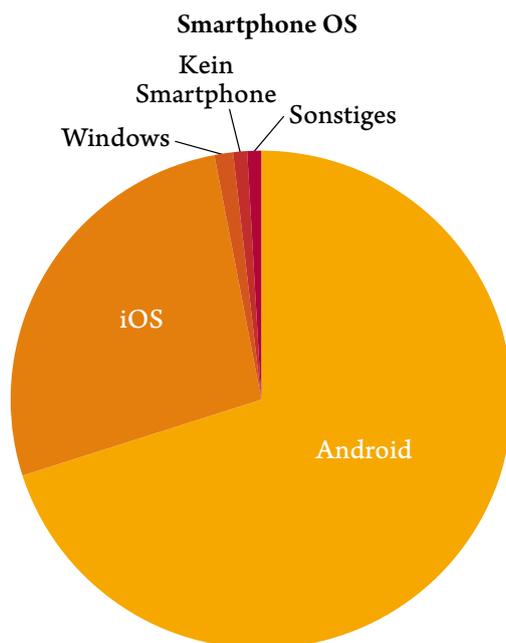
Wie schon 2013 zeichnet sich hier jedoch ein deutlicher Trend ab, wenn man die Nutzung der Betriebssysteme in Abhängigkeit vom Jahrgang betrachtet. Während Windows unter den den Bachelor-Erstsemestern noch mit über 60 Prozent dominiert, sinkt der Anteil in älteren Jahrgängen immer weiter, bis schließlich nur noch 21 Prozent der älteren Masterstudierenden Windows zur Arbeit nutzen. MacOS und Linux gewinnen somit immer weiter an Nutzern (von 18 auf 35 Prozent bzw. von 21 auf 44 Prozent), wobei das Verhältnis zwischen den Beiden interessanterweise relativ konstant bleibt.

Einen deutlich größeren Vorsprung hat Windows jedoch, wenn es um die Freizeitnutzung geht. Hier kommen alle Windows-Versionen zusammen auf 55 Prozent. Während das Verhältnis zwischen MacOS und Linux weiterhin relativ ausgeglichen bleibt, hat MacOS mit 22 gegen 19 Prozent knapp die Nase vorn. Hier lässt sich übrigens ein ähnliches Muster wie bei den Arbeits-Betriebssystemen beobachten: 76 Prozent der Erstsemester nutzen Windows, jedoch nur 43 Prozent der Studierenden aus höheren Mastersemestern.





Deutlich besser sieht es für Linux (in Form von Android) jedoch im Kampf der Smartphone-Betriebssysteme aus. Mit einem Marktanteil von 70 Prozent übertrumpft Android iOS deutlich, das iPhone-Betriebssystem kommt auf 27 Prozent. Mit nur einem Prozent, genauer gesagt vier Stimmen, liegt Windows abgeschlagen auf dem dritten Platz – mit nur einer Stimme mehr als »kein Smartphone«.

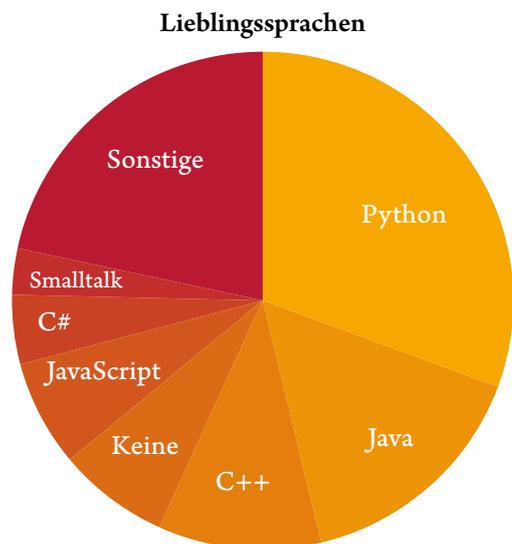


Programmiersprachen

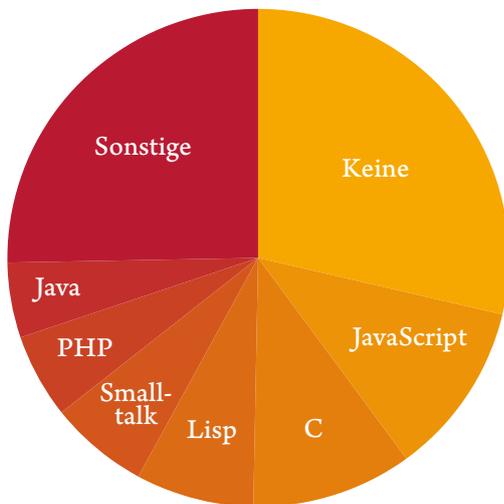
Wie schon die Betriebssysteme bieten auch Programmiersprachen einiges Diskussionspotential – nur eben mit mehr Parteien. Doch auch hier gibt es einen klaren Sieger: 31 Prozent der Stimmen für die Lieblingsprogrammiersprache gehen an Python, gefolgt von Java mit 16 und C++ mit 11 Prozent, während sich sieben Prozent nicht auf eine Sprache festlegen konnten. JavaScript, C# und Smalltalk konnten jeweils zumindest zehn oder mehr Stimmen erreichen.

Das sind einige deutliche Änderungen gegenüber 2013: Damals führte noch Java vor Python und C++ (25, 21 bzw. 14 Prozent). Die Plätze vier bis sieben belegten Ruby, JavaScript, C# und Smalltalk. Statt ehemals acht Prozent hat Ruby dieses Mal nur drei Prozent der Stimmen erhalten und ist somit auf den siebten Platz zurückgefallen.

Bei den unbeliebtesten Programmiersprachen geht der erste Platz statt an eine Sprache an die Toleranz der HPI-ler: 29 Prozent können jeder Sprache etwas abgewinnen. Die nächsten Plätze sind kaum überraschend: Wer noch nie über unintuitives Verhalten von JavaScript geflucht, sich mit C selber ins



Unbeliebte Sprachen

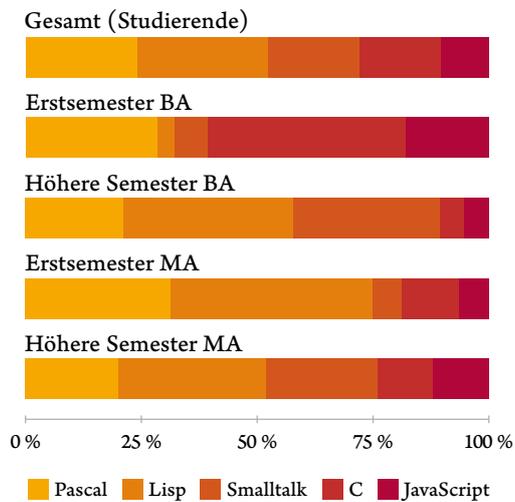


eigene Knie geschossen oder in Lisp die Übersicht über die Klammern verloren hat, kann sich glücklich schätzen. Im Vergleich mit den beliebtesten Sprachen fällt auf, dass die Meinungen zu JavaScript anscheinend stark auseinandergehen. Auch Java und Smalltalk sind in beiden Ranglisten auf relativ hohen Positionen zu finden. Ein interessantes Detail: Während C zu den unbeliebtesten Sprachen gehört, ist C++ sehr beliebt.

Seit 2013 hat sich auch hier einiges geändert: Die Antwort »Keine« lag damals mit 10 Prozent Stimmenanteil nur auf dem vierten Platz. C, PHP und Java waren besonders unbeliebt (21, 18, bzw. 17 Prozent). JavaScript wurde aufgrund geringer Stimmenanzahl gar nicht gesondert aufgeführt, sondern in »Sonstiges« einsortiert.

Während die beliebtesten Programmiersprachen bei Betrachtung der einzelnen Jahrgänge keine auffälligen Muster aufweisen, sieht das bei den unbeliebten Sprachen anders aus. Ein Hinweis vorweg: Aufgrund der großen Vielfalt der Antwortmöglichkeiten und dem hohen Anteil von »Keine« sind die hier betrachteten Stimmenzahlen relativ klein – auf die im Folgenden betrachteten Sprachen entfallen insgesamt 107 Stimmen aus allen

Unbeliebte Sprache (nach Semestern)

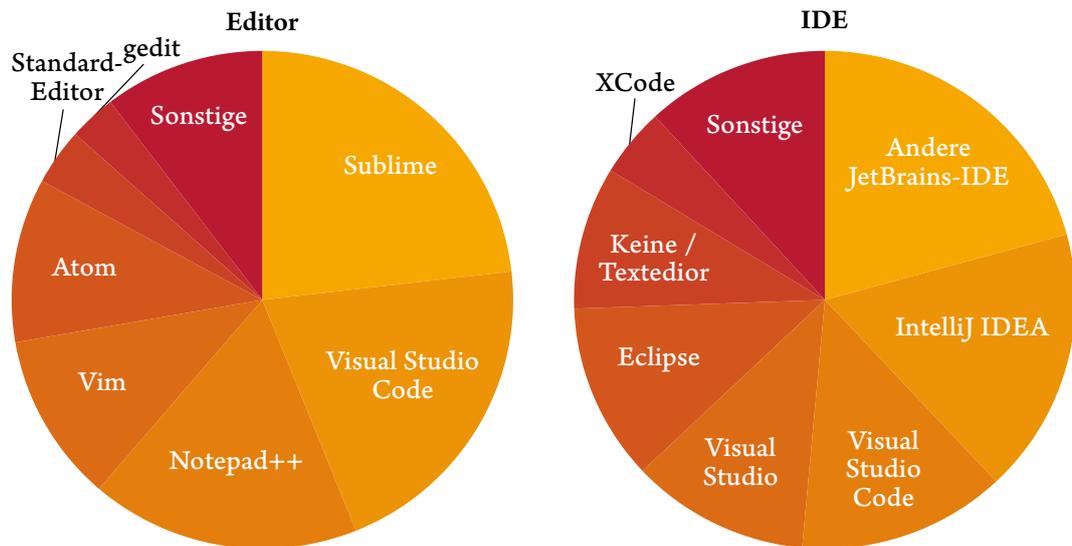


Bachelor- und Masterjahrgängen zusammen. Doch das hält mich nicht davon ab, zumindest Vermutungen aufzustellen.

JavaScript ist in allen Jahrgängen etwa gleichmäßig unbeliebt – immerhin eine Eigenschaft, auf die man sich bei JavaScript verlassen kann. Die Ergebnisse der anderen Sprachen weichen jedoch besonders im Bachelor stark vom Durchschnitt ab. Zum Zeitpunkt der Umfrage wurde in Programmiertechnik I anscheinend gerade Scheme behandelt, was den hohen Stimmenanteil von Lisp begründen würde. Die niedrigen Werte für C und Smalltalk hängen vermutlich damit zusammen, dass es noch keinen Kontakt damit gab – diese werden erst später behandelt. Der überdurchschnittliche Anteil von Pascal beruht sehr wahrscheinlich auf der häufigen Verwendung von Pascal im Informatikunterricht.

Software

Hier stellt sich die Frage: Wird Lukas auch für diese Einleitung eine abgedroschene Phrase finden, um zu beschreiben, dass es unterschiedliche Meinungen gibt? Ja! Der genutzte Texteditor ist schon fast mit einer Religion gleichzusetzen: Einige schwören auf



Sublime, für Andere kommen nur Kommandozeilentools in Frage. Doch was sagen die Zahlen? Wenig überraschend liegt Sublime mit 23 Prozent der Stimmen auf dem ersten Platz – jedoch dicht gefolgt von Visual Studio Code mit 21 Prozent. Auch Notepad++ ist mit 17 Prozent nicht allzu weit entfernt. Vim sichert sich sehr knapp den vierten Platz, mit einer einzigen Stimme Vorsprung vor Atom.

Die auffälligste Änderung gegenüber 2013 sind zwei Neuankömmlinge: Atom und VS Code sind erst 2014 bzw. 2016 erschienen – umso eindrucksvoller ist es, dass VS Code den ersten Platz nur knapp verpasst hat. Sublime und Notepad++ (ehemals 36 bzw. 26 Prozent) haben entsprechend stark an Marktanteil verloren.

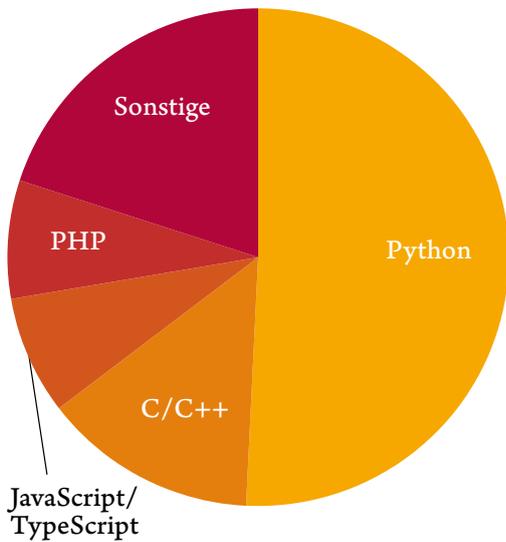
Bei den Entwicklungsumgebungen kann man sehen, was passiert, wenn man voreilig optimiert: Statt alle JetBrains-IDEs einzeln aufzuführen, habe ich nur IntelliJ IDEA herausgegriffen und alle anderen als »Andere JetBrains-IDEs« zusammengefasst. Wie sich herausstellt, war das ein Fehler: Diese Gruppe belegt mit 21 Prozent den ersten Platz, gefolgt von IntelliJ IDEA mit 17 Prozent. Somit ist hier leider einiges an Informationen verloren gegangen. Um das Verhältnis zwi-

schen den JetBrains-Tools zumindest erahnen zu können, bietet es sich jedoch an, die zugehörigen Antworten zur Lieblingssprache zu betrachten. Die beiden am häufigsten vertretenen Sprachen sind hier Python und C/C++ (51 bzw. 14 Prozent der Antworten, die »Andere JetBrains-IDEs« als IDE angeben). Somit liegt die Vermutung nahe, dass PyCharm und CLion noch relativ stark verbreitet sind.

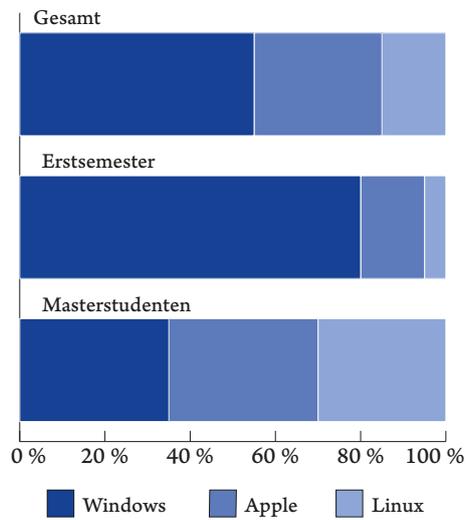
Des Weiteren legt Visual Studio Code auch bei den IDEs einen beachtlichen Auftritt hin: Mit 13 Prozent der Stimmen liegt es sogar vor dem »großen Bruder« Visual Studio – dieser teilt sich mit zwölf Prozent einen Platz mit Eclipse. Neun Prozent der HPI-ler nutzen übrigens keinerlei IDEs, sondern nur einen Texteditor zur Entwicklung.

Im Vergleich zu 2013 hat sich bei den IDEs einiges getan: Während Eclipse damals noch mit 35 Prozent auf dem ersten Platz lag, wurde es mittlerweile von IntelliJ IDEA überholt. Auch QtCreator hat deutlich an Marktanteil verloren, von ehemals zehn auf ein Prozent. Zu den Gewinnern zählt einerseits VS Code, allerdings kann auch PyCharm einen Anstieg von ehemals drei auf nun (vermutlich) knapp 11 Prozent der Stimmen verzeichnen.

Lieblingssprache (Andere JetBrains-IDE)



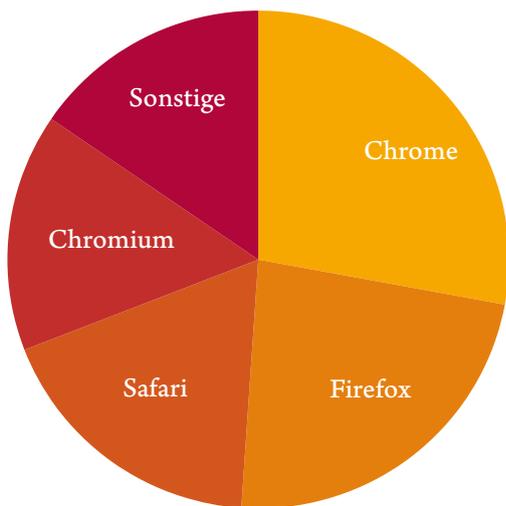
Genutzte Betriebssysteme



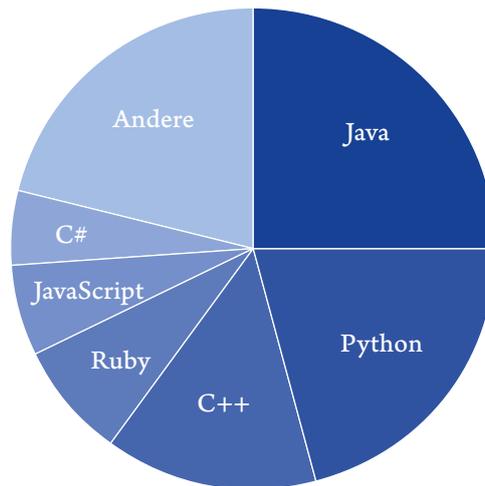
Ein Blick in die Vergangenheit

Linux konnte zulegen, Python überholt Java, Eclipse und Visual Studio müssen einstecken – und Diagramme haben nun ein passenderes Farbschema.

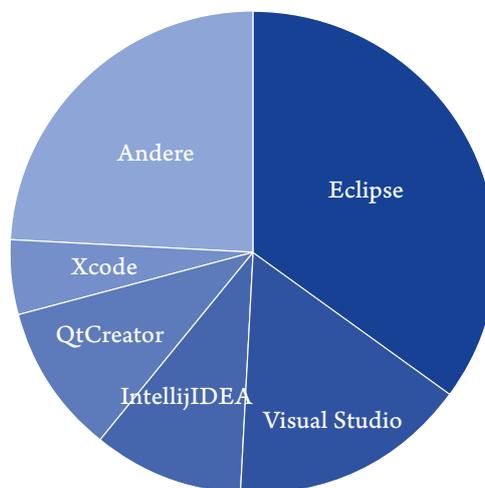
Browser



Beliebteste Programmiersprachen



Genutzte IDEs



Bei den Browsern gab es ein Kopf-an-Kopf-Rennen, jedoch konnte sich Chrome mit 41 Prozent der Stimmen knapp vor Firefox (40 Prozent) durchsetzen. Der dritte Platz geht an Safari (11 Prozent). Im Vergleich zu 2013 hat Chrome etwas eingebüßt, während Firefox zulegen konnte (ehemals 47 bzw. 34 Prozent). Während Opera damals noch sechs Prozent der Stimmen verzeichnen konnte, hat es diesmal nicht einmal für zwei Prozent gereicht.

– Lukas Wagner

Fruchtradar

Rumlaufen, Früchte finden und essen

Menschen, so wie Du einer bist, essen gerne Früchte. Im Studentendorf beim HPI gibt es Früchte, die man essen kann, von denen die meisten aber nicht wissen, dass es für den Körper auch gesund ist. Dazu gehören Cornelkirschen, Felsenbirnen und Weißdorn. Was man kennt, aber nicht erkennt, sind zum Beispiel die Walderdbeeren an der Tischtennisplatte oder die Esskastanie hinter dem Haus B.

Mundraub.org ist eine interaktive Karte, auf der ich solche Fundorte gerne eintrage. Diese Web-App braucht für jeden Fundort und jede Interaktion eine Internetverbindung. Allerdings wollte ich auch bei einem Besuch der österreichischen Berge, wo es kein Internet gibt, nach Früchten suchen und neue Fundorte eintragen. Aus der Idee, die Fruchtstandorte offline aufzunehmen, ist die App »Fruchtradar« entstanden. Inzwischen besitzt sie auch eine Karte mit Fruchtfundorten, die offline funktioniert und das Smartphone vibrieren lässt, wenn man sich einer Frucht nähert.

Freie Software

Die App ist eine freie Software, unter der GPL lizenziert und der Code auf GitHub verfügbar. So etwas tue ich, weil ich nicht dafür verantwortlich sein möchte, dass Mitglieder der Community, so wie ich, sich jedes Mal eine neue App aus den Fingern saugen müssen, wenn der Entwickler das Interesse verliert.

Was mir das bei der Entwicklung noch nutzt, beschreibe ich weiter unten.

Plattformen zum Teilen, die nicht geteilt werden dürfen

Als ich die App entwickelt habe, musste ich die API von *Mundraub* reverse-engineeren. Diese Plattform hat proprietären Code: Ich kann ihn nicht einsehen. Es gibt keine JSON-API. Es gibt Drupal-Formfelder, die man sich

als HTML runterladen, parsen und dann die undokumentierten Felder ausgefüllt als eine POST-Request an den Server senden muss. Wenn man was falsch macht, kommt »200 OK« zurück. Die Anbindung an *Mundraub.org* zu realisieren ginge wahrscheinlich dreimal schneller, wenn sie dokumentiert oder der Code offen wäre.

Nach einiger Zeit stellte ich fest, dass es auch andere Länder auf der Welt gibt, in denen Menschen dieselbe Idee hatten wie *Mundraub*. In Tschechien gibt es *Na-Ovoce.cz*, in der Slowakei *Fruitmap.org*. Es gibt weitere, englischsprachige Plattformen. *Na-Ovoce* ist open-source. Die API ist dokumentiert. Es dauerte wenige Tage, bis ich hier eine Anbindung an die App fertig hatte, sodass man sich nun entscheiden kann, wo man die Fruchtfundorte teilt: *Mundraub.org* oder *Na-Ovoce.cz*.

Der Gedanke des Teilens ist bei vielen Gemeinschaftsplattformen eine Grundlage für ihre Existenz: Damit sie funktionieren, tragen Leute einfach so etwas ein und haben Freude daran. Alle werden dadurch bereichert. Diese Plattformen bauen wahrscheinlich sogar auf offener Software auf und ich hoffe, dass auch sie später in der Lage sein werden, ihren eigenen Code zu teilen, nicht nur die Daten. Diese App soll eine Grundlage schaffen, indem sie Fruchtplattformen aus aller Welt vereinigt, was für eine proprietäre App schwer möglich wäre.

Lizenzen und Community

Ich lizenziere den Code unter *GPL*. Das ist eine *Share-Alike-Lizenz*. Grundsätzlich unterscheide ich da zwei Arten: »Mach, was du willst« (*MIT*, *Apache*, *CC-BY*) und »Share-Alike«, heißt »Weitergabe unter gleichen Bedingungen« (*GPL*, *MPL*, *CC-BY-SA*).



Peter Hintjens beschreibt in »How to make money from Open-Source« (Infobox links), dass MIT von Universitäten verwendet wird, damit subventionierter Code zu Billigpreisen auf den Markt kommt. In »Softwareanwender dieser Welt, vereinigt euch!« (Infobox mitte) zeigt Prof. Dr. Dirk Riehle, dass große Unternehmen Lizenzen im MIT-Stil dazu nutzen, eine kostengünstige Alternative auf den Markt zu bringen, um Monopole zu brechen.

Ich bin kein Marktführer. Für mich als kleinen Entwickler oder als Community hat die Lizenzwahl eine andere Motivation: MIT erlaubt eine weitere Lizenzierung zu irgendwelchen Bedingungen, kann also auch geheim gehalten werden. Wenn ich aber eine Abänderung meiner App sehe, möchte ich immer noch den Code einsehen und herausfinden, was die Änderungen waren. Deswegen wähle ich eine Share-Alike-Lizenz aus: Ich darf das Werk immer noch so benutzen, wenn es jemand anderes verändert hat. Auch Code-Beiträge sind dann GPL-lizenziert und ich darf sie ohne zusätzlichen Vertrag verwenden.

Governance

Der Quelltext ist auf GitHub gehostet. Damit ist er öffentlich einsehbar. Es gibt ca. 60 offene Issues, von denen 50 mit »Nicco macht es nicht« markiert sind. Zwölf sind »good first issues«. Personen, die mit GitHub, Android oder speziell meiner App anfangen wollen, zu entwickeln, können die »good first issues« verwenden, um einzusteigen. Diese sind sehr genau beschrieben. Es dauert länger, ein solches Issue zu erstellen, als es selber zu lösen.

Sie bilden einen zeitsparenden Einsprungspunkt für neue Entwickler und eine Möglichkeit, um zu sehen, ob der Aufnahmeprozess funktioniert, wenn sie von anderen gelöst werden. Wenn man auf GitHub nach »good first issue« sucht, dann findet man viele solcher Issues, mit denen man praktisch an Projekten in allen Sprachen in die Entwicklung einsteigen kann.

Wenn man eine neue App erstellt, hat Android schon Übersetzungen eingebaut. Viele Übersetzungsplattformen sind für open-source Apps kostenlos. So benutzt das Fruchtradar die Plattform »Transifex«, um die Übersetzungen zu crowd-sourcen. Eine Person aus Frankreich hat geholfen, große Teile der App ins Französische zu übersetzen. Könnte ich nicht.

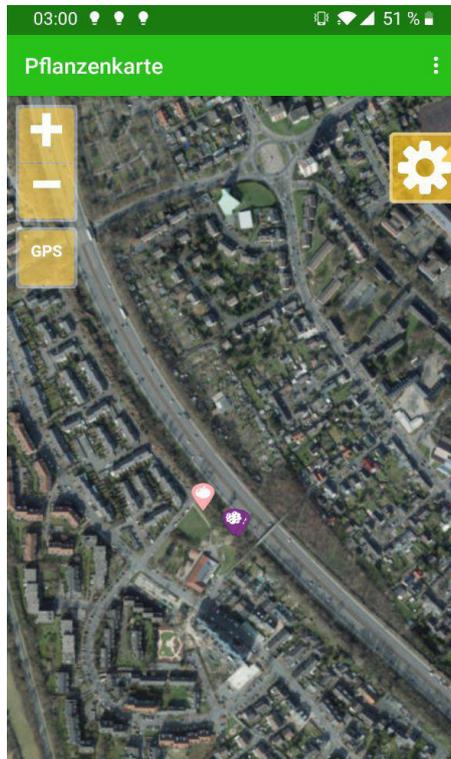
Publikation

Das Fruchtradar habe ich auf F-Droid (Infobox rechts) veröffentlicht. F-Droid ist dabei so eingestellt, dass es alle Git-Tags als neue Version baut. Immer wenn eine neue Version der App im master-Branch entsteht, wird diese automatisch publiziert. Das verkürzt die Feedback-Schleife für Beitragende und stellt mich ihnen nicht in den Weg. Es reicht ein Knopfdruck, um Änderungen live gehen zu lassen, was motiviert, etwas beizutragen.

Es gibt tatsächlich Leute, die diese App benutzen und auch Feedback geben. Von zweien möchte ich hier erzählen. Circa einen Monat nachdem die App veröffentlicht wurde, schrieb ein Designer ein Issue, dass er gern ein Logo entwerfen würde.

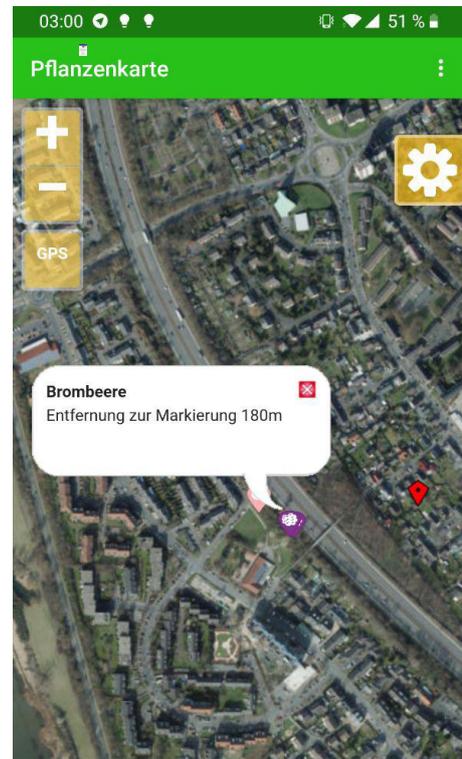
Kartenansicht (links)

Die Kartenansicht hilft in der Umgebung eingetragene Pflanzen zu entdecken



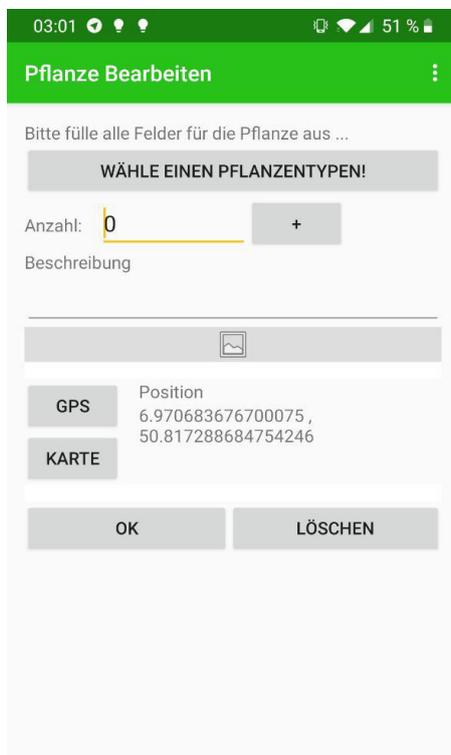
Details einer Brombeerpflanze (rechts)

Ein Klick auf die jeweilige Pflanze verrät einem wie weit entfernt man von dieser ist.



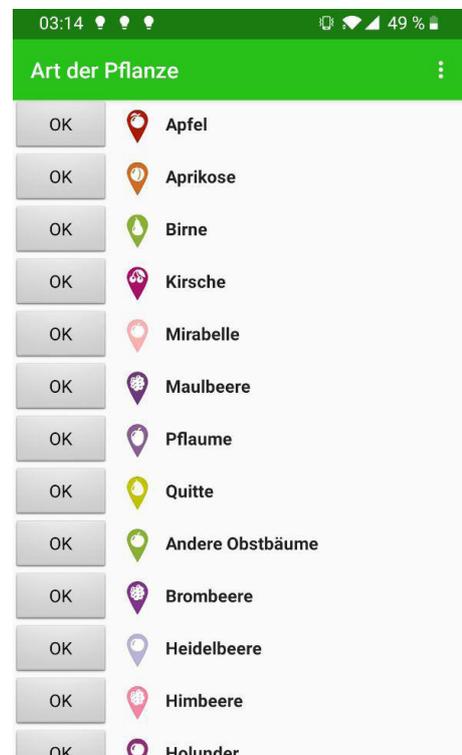
Datenaktualisierung (links)

Sollte man einer neuen oder fehlerhaft verzeichneten Pflanze begegnen, so kann man die Daten einfach direkt anpassen.



Pflanzentyp bitte! (rechts)

Ein wichtiges anpassbares Attribut ist die Art der Pflanze, sie ermöglicht erst die gezielte Suche auf der Kartenansicht.



Nach einiger Zeit hatte die App also ein neues Logo, statt des grünen Android-Männchens. Die Folge war, dass nun mehr Leute die App benutzten, Feedback gaben und beitrugen. Eine andere Person öffnete ein Issue und bot an, eine Datenschutzerklärung zu schreiben, weil es Spaß macht. Nun hat die App auch eine ausführliche Datenschutzerklärung.

Fazit

Ich schreibe schon lange open-source Software, um die Welt zu einem besseren Ort zu machen (z.B. indem ich den Code noch habe, wenn mein Rechner kaputt geht). Ich trage auch gern zu offener Software bei, wenn ich das für sinnvoll erachte. Durch diese App hatte ich die Möglichkeit, zu sehen, dass open-source die Schwelle zum Beitragen senkt und tatsächlich jemand auf die Idee kommt, mir zu helfen.

Gleichzeitig habe ich den F-Droid-App-Store kennengelernt. Diese offenen Apps erlauben es mir, Bugs zu fixen und die Android-Version auf die meines Handys herabzusetzen.

– Nicco Kunzmann

Der Autor empfiehlt zur Vertiefung:

»**How to make money from Open-Source**«

ein Blogbeitrag von Nutzer »pieterh«

hintjens.com/blog:27

»**Softwareanwender dieser Welt,**

vereinigt euch!«

eine Ringvorlesung der FU-Berlin

ot4os.imp.fu-berlin.de/

»**F-Droid**«

eine FOSS Distributionsplattform

f-droid.org/de/about/



Nachhaltigkeit am HPI

Ein Interview mit Lilith Diringer

Kaum am HPI angekommen und schon werden die ersten Schritte in die Wege geleitet, um die Klublandschaft am HPI ein wenig zu verwirbeln. Denn »zwar sind die ganzen Klubs am HPI schon super gut«, wie die Klubsprecherin und Initiatorin des Nachhaltigkeitsklubs Lilith sagt, »aber es fehlte eben noch ein Klub, der nachhaltige Themen in den Fokus seiner Klubarbeit stellt«. Der Prozess konnte also starten. Mitglieder finden, ein Konzept ausarbeiten, Mails für die Formalien herumschicken... Am 30. November fand dann genau einen Monat nach der ersten Klubaktivität die Gründungsfeier statt.

Zuerst die Frage, die allen wohl am meisten brennt: Seid ihr die Veganer-Polizei?

→ *Lilith*: Nein, sind wir nicht. Tatsächlich hat unser Klub bislang kaum Mitglieder, die sich rein vegan ernähren – auch wenn diese bei uns natürlich herzlich willkommen sind. Zum Einen sind wir einfach ein Zusammenschluss an Studierenden, die sich für nachhaltige Themen interessieren. In Bezug auf die Ernährung bedeutet das, dass uns bewusst ist, was wir essen. Da zählt die Reduktion tierischer Produkte genauso wie das Bevorzugen regionaler und biologischer Ware bis hin dazu, dass wir darauf achten, wie das, was wir kaufen, verpackt ist. Auch heißt das nicht, dass wir nicht auch ab und an Chips im

Supermarkt kaufen, oder Pizza essen – aber wir wollen allgemein darauf achten und den Konsum von nicht nachhaltigen Lebensmitteln im Gesamten reduzieren. Viel wichtiger jedoch ist, dass wir uns ganz und gar nicht auf das Thema Ernährung festnageln. Wir möchten noch ganz andere Bereiche zum Oberbegriff Nachhaltigkeit beleuchten. Auch erheben wir in keinem anderen Themenfeld den Zeigefinger und wollen in keinem Fall »Umweltsünder« bloßstellen. Viel wichtiger ist es, zum Nachdenken aufzufordern und zur offenen Diskussion anzuregen, indem wir auf Produktionsbedingungen und Auswirkungen unseres Konsums aufmerksam machen. Zum Beispiel informieren wir andere Studierende darüber, wie viel Wasser, CO₂ und andere Ressourcen für die Produktion einer Colaflasche verbraucht und wie viele Plastikabfälle produziert werden. Oder wir zeigen in unseren Blogartikeln wie Weihnachtsbäume in von Chemikalien verseuchten Monokulturen angebaut werden. Dabei beabsichtigen wir nicht, dass man sich bei jedem Schluck Cola schämt, oder dass man den Weihnachtsabend nicht mehr genießen kann, sondern wir zeigen leckere bzw. kreative Alternativen auf, durch die ihr als Studierende vielleicht erkennt, dass ein lebender Tannenbaum im Topf, eine Lichterholzinstallation, oder ein kreativer Filzbaum sogar noch schöner sein können.

Milch ohne Kuh

Der Nachhaltigkeitsklub regt unter anderem an, umweltfreundliche Alternativen für verschiedene Produkte auszuprobieren, etwa Hafer- statt Kuhmilch.



Was genau macht eigentlich der Nachhaltigkeitsklub?

→ *Lilith*: Allem voran bieten wir eine Plattform. Alle Interessierten bekommen über unseren Mailverteiler und die Telegram-Gruppe mit, wenn es in der Nähe Aktionen gibt. So waren wir z. B. schon auf der Konferenz Bits&Bäume vertreten und haben uns für die Teilnahme an einer Critical Mass verabredet. Darüber hinaus analysieren wir, wie nachhaltig das HPI ist. Wir schauen uns an, wo beispielsweise noch Papier eingespart werden kann und bewerben die Nutzung von Recyclingpapier. Wir recherchieren, welche Alternativen es gibt, unter anderem, ob wir eine nachhaltigere Druckerei für Flyer und Broschüren finden können und ob es bessere Getränke für die FridayBeers sowie nachhaltigeres Grillgut für die Angebote des FSR bzw. Veranstaltungsklubs gibt. Als dritte Sparte organisieren wir Workshoptage. Wir laden Referenten ein oder bringen uns gegenseitig bei, wie man Spülmittel selbst herstellt, wie sich Innenkomposter bauen lassen, wie man aus altem Brot leckere Aufläufe zaubert. Wir wollen dazulernen, Wissen austauschen und Alternativen aufzeigen. Allen voran ist es gut, seine eingespielten Gewohnheiten zu hinterfragen und durch die Anderen immer wieder auf neue Ideen gebracht zu werden. Neben

Massenfahrradfahren

Bei einer Critical Mass handelt es sich um ein scheinbar unorganisiertes Treffen von Radfahrern, die gemeinsam durch die Innenstädte fahren und so allein durch ihre Menge auf ihre Rechte gegenüber dem Autoverkehr aufmerksam machen. Man kann jederzeit zu einer organisierten Critical Mass hinzustoßen, oder eine eigene organisieren.



Frisch am HPI ...

... und schon krepelt sie das Institut um: Lilith, die Klubsprecherin des neuen Nachhaltigkeitsklubs.

den Workshoptagen haben wir laufende Projekte. Zum Beispiel stellen wir am laufenden Band Kambucha und Wasserkefir her – richtig leckere Getränke, die unsere Klubmitglieder dann auch konsumieren können. Darüber hinaus laden wir in speziell gewählte Hörsaalkinos und in Vorträge, zu denen wir Referenten einladen, alle Studierenden ein, mitzuhören und mitzudiskutieren. Unsere Themen sind dabei sehr weit gefasst und reichen vom Sozialen über Umwelt hin zu Datensicherheit. Beispielsweise haben wir auch den Flüchtlingsklub mit integriert, sodass er jetzt eine Untergruppe bei uns bildet. Und auch zu Aktionstagen wie dem Unterschriftenmarathon von Amnesty International planen wir eine Aktion.

Wie bist du auf den Gedanken gekommen, den Klub zu gründen?

→ *Lilith*: Ich beschäftige mich schon sehr lange mit dem Thema Umweltschutz. Seit ich 2015 an einem einjährigen WWF-Projekt

teilgenommen habe, bin ich noch mehr in das Thema vertieft. In der Schule habe ich mich bereits in der fairtrade AG und der Amnesty International AG engagiert und außerhalb in Netzwerken wie UNICEF und Plan International. In meinem letzten Jahr am Gymnasium habe ich zudem ein Großprojekt durchgeführt, in dem ich die gesamte Schule analysiert und konkrete Projekte durchgeführt habe, um die Schule »grüner« zu machen: gläserne Mülltonnen, um die Lebensmittelabfälle zu reduzieren, mehr vegetarische Alternativen am Kiosk, Ausbau der Fahrradwege. In dem Themengebiet kannte ich mich also schon aus und mir ist aufgefallen, dass es immer und überall noch etwas zu verbessern gibt – und der Weg dahin richtig Spaß machen kann. Deshalb dachte ich, es wäre super, mit motivierten Mitstudierenden das HPI zu einem Vorreiter zu machen.

Müssen wir demnächst damit rechnen schräg angeschaut zu werden, wenn wir nicht richtig Müll trennen?

→ *Lilith*: Nein, auch das nicht. Mülltrennung ist super wichtig und eigentlich auch gar nicht so aufwendig. Wir werden sicher auf das Thema zu sprechen kommen, aber wie gesagt: Wir bieten Informationen und Angebote für die Studentenschaft und hoffen, dass sie angenommen werden. Wir möchten niemanden zwingen, sondern wollen erreichen, dass jeder selbst für sich entscheidet, was ihm wichtig ist und was für ihn verantwortungsvolles Handeln bedeutet bzw. wie er dieses in seinem Alltag umsetzen möchte.

Versucht ihr auch, Essensreformen in der Mensa umzusetzen?

→ *Lilith*: Die Mensa ist etwas weiter hinten angestellt. Sie wird über das Studierendenwerk koordiniert. Zunächst möchten wir uns auf das HPI beschränken – und wenn hier dann alles zufriedenstellend läuft, weiten wir unseren Blick auf Themen, die die Uni Potsdam betreffen. Konkret beim Thema Essen

»Unsere Themen [...] reichen vom Sozialen über Umwelt hin zu Datensicherheit.«

wollen wir es erst einmal schaffen, Kooperationen mit umliegenden Lebensmittelerzeugern, sowie Geschäften und Restaurants aufzubauen, von denen wir z. B. Biogemüseboxen für die Studierenden beziehen können, oder mit denen wir foodsharing (Lebensmittelreste, die vor der Tonne gerettet werden) vielleicht sogar kostenlos der Studentenschaft anbieten können. Das alles befindet sich momentan aber noch stark in der Entwicklung.

Wie können Studierende mit eurer Unterstützung nachhaltiger leben?

→ *Lilith*: Studierende finden in unseren Workshops Ideen, wie sie alltägliche Handlungen nachhaltiger gestalten können. Die bei den Workshops gebauten Dinge – z. B. der Innenkompostierer – können von den Teilnehmern mitgenommen werden. Und wenn man schon einmal selbst Mandelmilch hergestellt hat, weiß man, wie gut es schmeckt und hat auch die ersten Handgriffe bereits geübt. In Zukunft ist die Hürde, »irgendwann einmal auszuprobieren, wie man eigentlich Mandelmilch herstellt«, nicht mehr so groß

und es geht auf einmal am Morgen in zwei Minuten. Darüber hinaus wollen wir nachhaltiges Leben attraktiv machen, indem wir beispielsweise anstreben, gesunde und nachhaltige Lebensmittel kostengünstig an euch zu verteilen. Auch machen wir die Studierenden auf Projekte und Produkte aufmerksam, an denen sie Freude haben und die gleichzeitig nachhaltig sind – auf die sie ohne uns aber gar nicht gekommen wären. Zudem machen wir es den Klubs einfacher, nachhaltig zu sein. Unter Anderem möchten wir mit dem Kochklub kooperieren und z.B. einen Kochtag unter dem Motto »saisonal Kochen« anbieten. Oder aber indem wir den Filmklub dabei unterstützen, nachhaltiges und kostenloses Catering über foodsharing zu bekommen. Oder indem wir eine Liste an Lieferservices zusammenstellen, die die Klubs anstelle der typischen Pizza nutzen können. Wir nehmen also einiges an Arbeiten ab, sodass der Umstieg zu nachhaltigen Alternativen nicht mehr schwer fallen sollte.

Wie viele Änderungen glaubt ihr, am HPI erreichen zu können?

→ *Lilith*: Viele! Dabei muss man aber auch sagen, dass das HPI in vielen Bereichen jetzt schon ziemlich gut abschneidet. Und auch dem FSR muss ein Lob ausgesprochen werden, denn er hat bereits vor unserer Klubgründung bei vielen getroffenen Entscheidungen auch den Nachhaltigkeitsgedanken in seine Überlegungen mit einbezogen. Die Aspekte, die es noch zu verbessern gibt, wollen wir aber in jedem Fall anpacken.

Wie weit werden sich eure Klubaktivitäten auf das Studentenleben der anderen Studierenden auswirken?

→ *Lilith*: Das kommt ganz darauf an, wie der individuelle Studierende unsere Aktivitäten aufnehmen möchte. In einigen Bereichen

wird er ganz automatisch betroffen sein, zum Beispiel, wenn wir das Abtrockensystem in den Toiletten verändern würden. Von anderen kann er sich beeinflussen lassen, wenn er es möchte: Indem er unserer Einladung folgt, Neues auszuprobieren, an unseren Diskussionen teilzunehmen, oder unsere Workshops zu besuchen.

Abschließend möchte ich noch sagen, dass wir besonders als junger Klub sehr offen für Ideen und Anregungen sind. Wir freuen uns über neue Mitglieder. Aber auch wenn ihr keine Zeit habt, euch bei uns regelmäßig zu engagieren, und einfach nur auf einen Missstand aufmerksam machen wollt, könnt ihr uns jederzeit kontaktieren und wir werden versuchen, eure Anregungen in unsere Klubaktivitäten einzubinden. Und auch wenn ihr Vorschläge für Seminare oder Referenten habt, die zu unserem Klubprofil passen und die ihr schon immer einmal einladen wolltet könnt ihr uns gerne einfach anschreiben: *lilith.diringer@student.hpi.de*
malte.barth@student.hpi.de

Vielen Dank für das Interview!

*Das Interview führten Leonard Geier
und Jonas Bounama*

Von Apps zur Alkoholkontrolle hin zu Herzpatienten

Seit 2017 befindet sich das HPI in Transformation. Ab diesem Jahr beschäftigt sich Prof. Dr. Erwin Böttinger mit der Konzeption eines neuen Studiengangs – weg vom reinen ITSE, hin zu einem immer gefragteren Anwendungsgebiet: der Gesundheitsversorgung.

Um den Studiengang mit aufzubauen, sind zum Start der ersten Masterstudierenden dieses Faches zum 1. Oktober 2018 die beiden Gesundheitsforscher Prof. Bert Arnrich sowie Prof. Christoph Lippert an die Digital-Engineering Fakultät berufen worden. Sie haben uns verraten, welche Konzepte und Inhalte sie in diesem Studiengang umsetzen möchten und welche Potentiale die Schnittstelle zwischen Technik und Gesundheit bereits heute sowie in Zukunft bietet.

Sie beide haben bereits viel im Bereich der Gesundheit geforscht. Wie unterscheidet sich Machine Learning in der Medizin von der Anwendung in technischen Bereichen?

→ *Lippert*: Das kommt ganz auf den konkreten Einsatz an. Es gibt Bereiche, die auch im Gesundheitssystem ganz klassisch funktionieren. z. B. die Bildanalyse beim MRT. Hier können Algorithmen den Ärzten beim Erkennen auffälliger Strukturen helfen. Dies ist nicht großartig von der klassischen Bilderkennung zu unterscheiden. Daneben gibt es aber auch weniger klassische Bereiche, z. B. wenn es um die rigorose Anwendung von Statistik geht. Allgemein kommt in der Diagnostik und Forschung mehr Engineering vor als in anderen, rein informatischen Themenbereichen. Sehr häufig müssen kausale Zusammenhänge der Naturwissenschaften verstanden werden. Es geht darum, der Ursache auf den Grund zu gehen, aktuell gibt es noch viele Hypothesen, die auch häufig bereits veröffentlicht werden, sodass wenigstens die ersten Anhaltspunkte festgehalten werden.

→ *Arnrich*: In der medizinischen Anwendung ist die Erklärbarkeit besonders wichtig. Eine Blackbox ist hierbei sehr schwierig. Die verwendeten Verfahren müssen vermittelbar sein, weil der Mediziner die Verantwortung trägt und das Funktionieren solch komplexer Systeme sicherstellen muss.

→ *Lippert*: Oder als Alternative der Einsatz von Statistik, die sonst im technischen Bereich nicht so häufig angewandt wird.

Ist es schwierig, an verwendbare Daten heranzukommen bzw. mit diesen unter Datenschutzaspekten richtig umzugehen?

→ *Lippert*: Dies ist stark von den Anwendungsbereichen abhängig. Werden Techniken in Autos oder Flugzeugen eingesetzt, müssen diese natürlich auch gut validiert werden.

→ *Arnrich*: Wie man mit den Gesundheitsdaten richtig umgeht, wird sehr kontrovers diskutiert. Es kann niemand vorhersagen, wofür sie in Zukunft genutzt werden. Beispielsweise ist es ganz natürlich, dass wir auf der Grundlage unseres Fahrverhaltens in unterschiedliche Versicherungsklassen bei der PKW Nutzung eingeteilt werden. Es könnte also auch passieren, dass wir anhand unseres Lebensstils bald verschiedene Krankenkassenbeiträge zahlen müssen. Besonders gesund lebende Personen könnten somit »belohnt« werden, ähnlich wie jahrelang sichere Autofahrer.

→ *Lippert*: Es wird versucht, viel zu pseudonymisieren und anonymisieren. Ehrlich gesagt ist dies aber ab einem gewissen Grad unmöglich. Eine Genomsequenz ist viel eindeutiger als ein Fingerabdruck – da ist es selbstverständlich möglich, auf die Person zurückzuschließen. In offenen Gesundheits-

systemen wie der UK jedoch, ist es viel selbstverständlicher, Gesundheitsdaten preiszugeben. Schließlich können wir ohne das Teilen dieser Daten auch keine neuen Erkenntnisse gewinnen. In einem öffentlichen System macht es durchaus Sinn, Informationen über sich selbst preiszugeben, damit diese letztendlich der Allgemeinheit zugute kommen.

Gehen wir einmal näher auf den neuen Studiengang Digital Health ein. Welche Lehrveranstaltungen planen Sie im Konkreten? Welchen Eindruck haben Sie vom ersten Jahrgang, der am 1. Oktober begonnen hat?

→ *Arnrich*: Wir beide sind erst seit Anfang Oktober hier und konnten daher zur Vorarbeit gar nicht viel beitragen. Bis jetzt haben wir vor allem Mitarbeitergespräche geführt. Bei den 30 Studierenden sind wir von der Mischung sehr positiv gestimmt: Etwa die Hälfte haben einen medizinischen, die andere Hälfte einen IT-Hintergrund. Auch was das Verhältnis von deutschen und ausländischen Studierenden sowie die Geschlechter betrifft, ist die Gruppe sehr ausgeglichen. Zurzeit ist es besonders wichtig, den Wissensstand aller anzugleichen. Hierfür gibt es sogenannte bridge modules, bei denen die Gruppen jeweils aus dem Bereich mehr lernen, aus dem sie gerade nicht kommen. (Schmunzelnd) Grundlagen in Mathe brauchen aber alle noch einmal. Zudem wird es einige Veranstaltungen geben, die frei wählbar, und tatsächlich auch offen für andere Studierende an der Digital Engineering-Fakultät sind. Bislang aber halten wir beide noch keine Vorlesungen. Im Sommersemester werde ich ein Modul zu Connected Health Care anbieten, sowie einen Statistikkurs Mobile Computing. Hierbei bekommen die Studierenden einen Methodenkoffer, mit dem sie eigene Ideen entwickeln und Hypothesen entwickeln können, um Experimente mit dem eigenen Smartphone durchzuführen.

Dieser Kurs kam bislang immer gut an, z. B. haben einige Studierende eine App gebaut, mit der das Smartphone herausfinden kann, ob sein Träger betrunken ist.

→ *Lippert*: Im Sommersemester werde auch ich mit zwei Vorlesungen beginnen. Die erste ist eine Statistikvorlesung zum Verstehen von Krankheiten und Risikofaktoren sowie für die Erstellung von Modellen und die Durchführung von Analysen. Der zweite Kurs wird sich aufs Machine Learning for Precision Health Care (medizinische Bildanalyse besonders für die Anwendungen im Gesundheitsbereich) beziehen. Es wird unter anderem um Clusteringverfahren gehen.

→ *Arnrich*: Insgesamt befindet sich der Studiengang noch in der Entwicklung. Fest geplant ist der Ablauf nur bis zum kommenden Sommersemester.

Welche Lehrenden werden hierfür eingesetzt werden? Wie genau ist also Ihr Team aufgebaut?

→ *Lippert*: In den Vorlesungen werden unterschiedliche Personen zum Zug kommen, die dann z. B. zwei Veranstaltungen halten. Für einen beginnenden Studiengang ist dies auch durchaus sinnvoll. Erst wenn es in die Tiefe geht, ist es wichtig und nützlich, wenn es einen festen Professor für jedes Modul gibt. Bei unserem ersten Gesamttreffen waren wir mit allen Externen 30 Involvierte.

Gibt es vergleichbare Studiengänge an anderen Standorten in Deutschland?

→ *Arnrich*: Es gibt in Deutschland einige Standorte, an denen es Studiengänge in ähnlicher Richtung gibt, von deren Erfahrung wir sicherlich lernen können. Daher lud das HPI auch am 6. und 7. Dezember zum Digital Health Forum ein. Internationale Experten aus Wissenschaft, Wirtschaft und Politik traten dort in den Austausch.

Was könnten mögliche Hürden beim Aufbau des Studiengangs sein?

→ *Arnrich*: Das HPI war bislang sehr fokussiert. Nun soll die Anzahl der Professuren von 14 auf 24 erweitert werden. Wir sind eine Art Versuchskaninchen und müssen noch herausfinden, wie wir uns in der Technikdominanz am besten finden. Momentan ist z. B. für Doktoranden noch nicht ganz klar, welche Zusatzkurse belegt werden müssen, um in unseren Fachbereichen promovieren zu können. Wie passen die interdisziplinären Projekte zusammen? Wie können die Personen aus den verschiedenen Fächern am besten zusammengebracht werden?

Auf welche Berufsfelder bereiten Sie die Studierenden vor? Wie zukunftsfähig sind die erlernten Methoden und Fähigkeiten?

Prof. Christoph Lippert

studierte bis 2008 in München Bioinformatik mit abschließender Diplomarbeit bei Siemens im Bereich *Machine Learning*. Anschließend folgte eine Promotion am Max-Planck-Institut für Intelligente Systeme in Tübingen (Genomweite Assoziationsstudien). Ein Praktikum mit anschließender Anstellung bei Microsoft Research führte ihn ab 2012 für drei Jahre nach Los Angeles. Es folgte ein Wechsel zu einem kalifornischen Start-up im Bereich der Genomsequenzierung. Dort führte er unter anderem Projekte zu *Machine Learning* durch und arbeitete mit verschiedenen Technologieunternehmen zusammen. 2017 begann er damit, seine akademische Karriere voranzutreiben, die er in Berlin am Max-Delbrück-Institut als Juniorgruppenleiter begann, bevor er dem Ruf ans HPI folgte.

→ *Arnrich*: Grob haben wir vier Berufsfelder definiert, darunter natürlich die akademische Karriere, aber auch den Einsatz im Medical Data Engineering. Berlin z. B. hat sich zum Ziel gesetzt, bis 2030 ein Health Hub zu werden. Hierbei werden medizinische und universitäre Zentren zusammengebracht.

→ *Lippert*: Generell müssen sich unsere Absolventen keine Sorgen darum machen, später eine Anstellung zu finden. Momentan werden Daten meist weggesperrt, damit bloß keine Gesetze verletzt werden. Ein sehr interessantes und nützliches Berufsfeld wäre es daher, sich darum zu kümmern, Daten bestmöglichst zu nutzen und auszuwerten, ohne Gesetze zu brechen.

Zum Abschluss: Wie lassen sich Ihrer Meinung nach Medizin und Technik gut kombinieren?

→ *Arnrich*: Wichtig ist, dass die Ärzte offen sind. Als ich meinen Weg im Gesundheitsbereich gefunden habe, hatte ich das Glück,



dass meine Kooperationspartner die IT nicht verteufelt haben, sondern erkannten, dass sie davon profitieren können. Die Bereitschaft, sich mit neuen Technologien auseinanderzusetzen ist die Voraussetzung für eine gelungene Zusammenarbeit. Schwierig bleibt es dennoch, sich zu verständigen. Es benötigt etwas Zeit, bis man eine gemeinsame Sprache spricht. Doch die Mühe lohnt sich. Wir haben bereits versucht, einige medizinische Studien ohne medizinisches Fachpersonal durchzuführen – und sind stets gescheitert. Auch wenn es zunächst etwas mühsam erscheint, kann man durch Zusammenarbeit Projekte umsetzen, die in den einzelnen Gruppen nie möglich wären.

→ *Lippert*: Die Technologien, die durch die Schnittstelle zwischen Informatik und Medizin entstehen, sind in jedem Fall zukunftsweisend. Eine Genomsequenzierung ist inzwischen kaum mehr mit Aufwand verbunden und auch *Designer Babies* und *Genome Editing* sind bereits lange keine Zukunfts-



musik mehr. Um diese Projekte umzusetzen, ist der Studiengang Digital Health von zentraler Bedeutung.

Vielen Dank für das Interview!

*Das Interview führten Tobias Markus und
Lilith Diringer.*

Prof. Bert Arnrich

wollte nach dem Abitur etwas mit Naturwissenschaften machen und bezeichnet Informatik als sein damaliges Hobby. Über eine Beratung wurde er auf den Studiengang Naturwissenschaftliche Informatik an der Universität Bielefeld aufmerksam. Mit der Diplomarbeit in Kooperation mit einem Herzzentrum spezialisierte er sich auf das Themengebiet der medizinischen Informatik. Hier erfuhr er zum ersten Mal, wie es möglich ist, aus den vielen gesammelten Daten zu lernen, und errichtete ein *Data Warehouse*. Auf diesen Daten konnte er Data Mining betreiben, im Fokus stand das Erkennen verlässlicher Muster. Seine Motivation dabei war die Vision, bei der Gesundheitsversorgung zu helfen und sie zu verbessern. Für seinen Postdoc verschlug es ihn an die ETH Zürich, um dort alltägliche Gesundheitsdaten zu untersuchen. Gefolgt wurde sein Postdoc von einem vierjährigen Aufenthalt in Istanbul, unterstützt durch eine EU Förderung. Im Zuge seiner Rückkehr nach Deutschland stieg er in München in die Unternehmensberatung ein. Die Ausschreibung des HPI entsprach schließlich genau seinem Interessengebiet, der Kombination aus Informatik und Medizin. Beim Anwendungsfeld Connected Health Care werden zum Wohle der Menschheit gesundheitsrelevante Daten aus dem Alltag erfasst und ausgewertet.



Halloweenkürbisse
Kunstklub



Paradox

Ronja Wagner

Warum wir alle ins kalte Wasser springen sollten

Wir alle wissen, wie wohltuend eine warme Dusche sein kann. Aber tatsächlich ist eine kalte mindestens genauso angenehm. Ihre Effekte sind in jedem Fall viel langfristiger.

Zugeben, wenn man im Winter am Kamin sitzt (oder sich mangels Kamin seiner mehr oder weniger funktionierenden Heizung erfreut), schießt einem nicht sofort der Gedanke durch den Kopf, eine kalte Dusche zu nehmen. Und ich gebe auch zu, dass es jedes Mal eine gewisse Überwindung ist, den Duschhahn bis zum Anschlag in Richtung »kalt« zu drehen. Aber trotzdem ist es den Versuch wert und es gibt mehr als genug Personen, bei denen der Tag nicht vor dem ersten Kontakt mit dem (eisig) kalten Nass beginnen kann.

Aber warum solltest du dich in den Kreis dieser Personen eingliedern? Der Vorteil an der kalten Dusche – oder noch besser: dem Eisbad – ist, dass sie eben keine trendige »Bio-Hacking-Methode« ist, bei der leicht dubiose Unternehmen versuchen, dir anhand von mittelmäßigen Studien übertriebene Produkte zu verkaufen, oder bei der absolut vertrauenswürdige Start-ups nur deine Daten brauchen und dann versprechen, dich durch

deren Analyse zu einem besseren Menschen zu machen. Nein, die kalte Dusche spart nicht nur Warmwasser, Zeit (versuch mal, genauso lange kalt wie warm zu duschen) und stärkt das Immunsystem, sondern gewährt schon direkt nach dem Verlassen der Dusche erste Erfolge. Denn spätestens am Ende dieser ist jede Müdigkeit verfliegen und du fühlst dich glücklich, da der Körper das Glückshormon Dopamin ausschüttet. Dopamin ist übrigens auch der Grund, weshalb sich der Mensch zu einigen Drogen hingezogen fühlt – unter anderem auch diese, die es jeden Freitag zwischen 16:00 und 17:00 Uhr kostenlos am HPI gibt. Man könnte also die kalte Dusche auch als gesunde Droge bezeichnen.

Das Beste ist aber, dass sie besonders leicht (oder auch weniger leicht) gestressten Studenten dabei hilft. Einerseits, indem sie wie schon erwähnt die Zeit verringert, die man zum Aufstehen und ganz besonders im Bad braucht, falls man kein Eisbär ist. Außerdem verringert ein kalte Dusche das Stresslevel

Partnerlook

Für Menschen wie
Schafe kann kaltes
Wasser wohltuend sein.



und man gewöhnt sich an unkomfortable Situationen, sodass man generell besser mit solchen Situationen umgehen kann.

Nun bist du natürlich hochmotiviert, lässt deine Mate und alles andere stehen und willst direkt unter die Dusche springen. Aber bevor du den Hahn auf ganz links (oder im Wohnheim eben nur am rechten Hahn) drehst, dich komplett unter den Wasserstrahl stellst und dann mit einem spitzen Aufschrei sofort wieder vergisst, diesen Artikel jemals gelesen zu haben und du deinen Leichtsinn bereust, auf irgendeinen komischen Typen gehört zu haben, lass mich dir noch ein paar Tipps geben, wie du diese Situation vermeiden kannst. Bedenke zuerst, dass man alles in Maßen genießen sollte und es vielleicht kein gutes Zeichen ist, wenn dein Körper auf einmal anfängt zu hyperventilieren. Fang also damit an, dich am Ende einer warmen Dusche einige Sekunden kälter als sonst laukalt abzduschen, beginnend mit Armen und Beinen, und das Ganze dann langsam zu steigern. Außerdem hilft es, wenn du dich auf den warmen Punkt in der Magenregion konzentrierst. Falls du nicht weißt, wo der sein soll: Macht nichts, konzentriere dich auf ihn.

– Armin Wells

»Man könnte also die kalte Dusche auch als gesunde Droge bezeichnen.«

Armin Wells

Armin empfiehlt



Grün trifft Gold - wie kann ich mein Geld nachhaltig anlegen?

Wir Studierende klagen immer über Geldmangel. Ganz egal, ob wir uns durch BAföG-Anträge hindurch kämpfen, lange Diskussionen mit unseren Eltern führen, oder uns mit Minijobs über Wasser halten – es reicht nie aus!

Denn der Schritt vom Schüler zum Studierenden ist gerade der Moment, in dem man sich zum ersten Mal gezwungenermaßen mit seinen Finanzen beschäftigen muss. Die Eigenverantwortung im Bereich der Geldverwaltung nimmt schlagartig zu. Der Zeitpunkt ist erreicht, ab dem man nicht mehr bequem das Taschengeld bei den Eltern abholen kann und das Essen abends auf dem Tisch steht – und man beginnt auf einmal damit, sich näher in den Angebotsdschungel von Konten und Anlagemodellen hineinzufuchsen.

Die boomenden Banken, auf die man bei den ersten Recherchen stößt, sind in Deutschland die Deutsche Bank mit über 100.000 Mitarbeitern gefolgt von der Deutschen Zentralbank mit knapp 30.000 Angestellten – und auch die KfW sowie die Commerzbank spielen ganz vorne mit.

Gibt es keine ernst zu nehmenden Alternativen neben diesen riesigen Geldinstituten?

Doch, die gibt es - und das nicht erst seit Kurzem. Trotzdem haben fast drei Viertel der Menschen, die sich laut einer Umfrage der Zeb 2012 für sozial-ökologische Bankangebote interessieren, noch nie etwas von deren Existenz erfahren.

Um hierbei Abhilfe zu schaffen, habe ich die populärsten Alternativ-Modelle genauer unter die Lupe genommen: GLS Bank, Triodos, Ethikbank, Umweltbank, DKB.

Auf der Homepage Informationen nachlesen kann jeder. Daher habe ich einen anderen Weg versucht: Mich ernsthaft beraten lassen und durch Telefon und E-Mail direkten Kontakt zu den jeweiligen Banken aufnehmen. DKB und Ethikbank konnten mir leider »aufgrund Zeitmangels«, wie sie in der



Antwort auf meine Anfrage geschrieben, keine persönlichen Antworten geben – dort musste ich tatsächlich auf die jeweiligen informationsgefüllten Internetauftritte zurückgreifen. Die restlichen drei vorgestellten Geldinstitute haben mir ausführliche Fragebögen beantwortet oder Interviews mit mir geführt.

Bevor es nun aber mit der Vorstellung der Banken losgehen kann noch ein Hinweis: dieser Artikel soll euch einen Überblick über die verschiedenen Angebote vermitteln. Der Text dient nicht als Empfehlung für bestimmte Banken oder dem Aufstellen eines Rankings. Am Ende seid es immer noch ihr, die eure Ansprüche und Kriterien am besten kennen und danach ein passendes Geldinstitut auswählen können.

Triodos Bank

Entstehung und Entwicklung

Die Triodos Bank in Deutschland wurde 2009 als Tochter der niederländischen Triodos Bank aufgebaut, die seit 1980 existiert. Tatsächlich geht die Entstehungsgeschichte bis in die 1960er und 1970er Jahre zurück. Damals formierte sich eine kleine Gruppe sozial engagierter Menschen aus der Wirtschafts- und Finanzwelt in den Niederlanden. Ihr Ziel: herausfinden, wie Geld nachhaltig verwaltet werden kann.

In Deutschland gehört Triodos laut eigener Aussagen aktuell zu den am stärksten wachsenden Banken. Belegt werden diese Daten z. B. durch einen Kundenzuwachs von 17% im ersten Halbjahr 2017. Triodos führt diesen Zuwachs auf den steigenden Wunsch der Menschen zurück, Gewissheit darüber zu haben, was mit ihrem Geld passiert. Ebenso wird die Finanzkrise als wichtiges Ereignis angesehen, das die Kundenzahlen potenzierte. Hinzu kommt ein neues Gesetz, das im September 2016 in Kraft trat. Durch dieses kön-

nen Bankkonten deutlich leichter gewechselt werden.

Angebote und Kundengruppe

Die Triodos Bank richtet sich an alle Menschen, die ihr Geld nachhaltig einsetzen möchten. Besondere Voraussetzungen gibt es nicht. Ein Konto kann eröffnen, wer mindestens 18 Jahre alt ist.

Es handelt sich bei Triodos um eine Direktbank, die ähnliche Konditionen bietet wie gewöhnliche Hausbanken. Unter anderem können Giro- oder Tagesgeldkonten eröffnet werden - und es gibt die Möglichkeiten, sein Geld als Festzins oder in einem Sparplan anzulegen. Ist man an nachhaltigen Fonds und aktienähnlichen Rechten interessiert, ist man bei der Triodos Bank ebenso an der richtigen Adresse. Auch Geschäftskonten werden bei Triodos bedient, Kredite jedoch werden nicht vergeben.

Nimmt man die Dienste der Triodos Bank in Anspruch, so muss man für das Girokonto mit Kontoführungsgebühren von 3,50 Euro im Monat rechnen.

Service

Zur Beratung und zur Verfolgung der Transaktionen bietet Triodos einen umfangreichen Online Service sowie telefonische Erreichbarkeit. Der Kontowechselservice macht es potentiellen Kunden besonders einfach, ihr Geld zur Triodos Bank zu verlegen.

Persönliche Beratungen bietet die Triodos Bank an ihren Niederlassungen in Frankfurt am Main und Berlin. Als Direktbank weist sie jedoch keine klassischen Bankfilialen auf.

Hat man schließlich ein Konto, kann man mit der Triodos Kreditkarte weltweit an jedem Geldautomaten mit Mastercard-Logo gebührenfrei Geld abheben – zum Beispiel bei allen Großbanken, Volksbanken, der Reisebank und vielen mehr.

Projekte, Kunden und Sonstiges

Die Triodos Bank setzt sehr auf Transparenz. Jeder Kredit, der an ein nachhaltiges Unternehmen vergeben wird, wird auf der Website veröffentlicht. So kann jeder Kunde direkt sehen, wie sein Geld wirkt. In Deutschland hat die Triodos Bank rund 15.000 Kundinnen und Kunden, in ganz Europa sind es über 670.000. Die Triodos Bank ist eine der einzigen dieser Banken, die auch außerhalb Deutschlands agiert. Von ihrem Ursprungsland Niederlande hat sie sich inzwischen in Belgien, Großbritannien und Spanien etabliert. Zusätzlich sind in nächster Zeit Erweiterungen in Europa, wie z.B. nach Frankreich, geplant.

Umweltbank

Entstehung und Entwicklung

Die Umweltbank wurde 1997 von Horst P. Propp gegründet. Als ehemaliger Vorstand bei der Ökobank baute er gemeinsam mit seiner Frau die Umweltbank auf. Besonders wichtig waren ihm dabei eine höhere Seriosität und Professionalität sowie bessere Beratungstätigkeiten, die er mit ökologischen Gesichtspunkten in Einklang bringen wollte.

Angebote und Kundengruppe

Besonders zum zwanzigjährigen Jubiläum kann die Umweltbank wie Triodos auf ein stetiges Wachstum seit Gründung zurückblicken. Eine weitere Parallele zu Triodos: Auch Oliver Patzsch, zuständig für die Kommunikation der Umweltbank, stellt den besonderen Zuwachs durch die Finanzkrise heraus: »Es wurde offensichtlich, dass die meisten bis dahin populären Bänker das Geld sehr risikoreich anlegten. Viele suchten daraufhin nach Alternativen«.

Die Kundengruppen sind überwiegend Privatsparer. Denn genau auf diesen Bereich

hat sich die Umweltbank spezialisiert: Sparen. Während Girokonto, Wertpapierangebot und Kreditvergabe eher im Hintergrund liegen, sind die Angebote und dementsprechend auch die Anleger bei dieser Bank umso breiter gestreut. Von der Großmutter, die 25€ im Monat für ihr Enkelkind zurücklegen lässt, bis hin zum Arbeiter, der auf ein Haus spart. Je nach festgelegter Laufzeit des Kontos variieren auch die erhaltenen Zinsen. Angefangen bei Tagesgeldkonten, die einen Zinssatz von etwa 0,025% bieten, bis hin zu langfristigen Anlagen mit Zinsen von 1,25%. Kontoführungsgebühren fallen keine an. Mit diesen Zahlen kann die Umweltbank laut eigener Angaben mit den »großen« Geldinstituten durchaus mithalten. Einige konventionelle Banken würden sogar überhaupt keine Zinsen vergeben. Höhere Zinsen anderer Geldinstitute hingegen wären meist mit einem höherem Risiko verbunden.

Service

Auch bei der Umweltbank handelt es sich um eine Direktbank und dementsprechend weist auch diese Bank kaum Filialen auf. Beratungsgespräche erfolgen daher ausschließlich in Nürnberg persönlich, ansonsten wird auf die telefonische Beratung verwiesen.

Da es keine Möglichkeit der Einrichtung eines Girokontos gibt, ist auch das Abholen von Geld an Automaten nicht möglich. Transaktionen von anderen Konten können jedoch problemlos durchgeführt werden, was erneut den Sparaspekt betont.

Projekte, Kunden und Sonstiges

Zwei Drittel der Anlagen werden in Projekte zur Förderung erneuerbarer Energien investiert. Das restliche Drittel dient der Unterstützung nachhaltiger Bauprojekte von, z.B. für uns besonders interessant, Studentenwohnheimen mit geringem Energieverbrauch, oder



Geldautomaten

Auch bei nachhaltigen Banken kann man vielerorts Geld abheben.

Schulen. All diese Projekte müssen innerhalb Deutschlands ausgeführt werden. Dadurch hat die Umweltbank zum einen staatliche Sicherheit, zum anderen können Projekte auch persönlich besucht und betreut werden.

Welche Projekte letztendlich mitfinanziert werden, ist durch Positiv- und Ausschlusskriterien eindeutig geregelt und wird im Einzelfall von den Mitarbeitern der Kreditabteilung beschlossen. Hierbei wird stets darauf geachtet, dass die Projekte auch tatsächlich dem Umweltschutz dienen, sich gleichzeitig aber auch auf ökonomischer Ebene lohnen. Dabei wird eine allgemeine Projektübersicht mit einem hohen Anspruch an Transparenz geboten. Seit der Gründung konnten so 22.000 Projekte unterstützt werden. 114.000 Kunden verzeichnet die Umweltbank seit Beginn ihrer Arbeit. Über die aktuellen Kundenzahlen ließ sich jedoch nichts Genaues herausfinden. Was die Umweltbank im Speziellen auszeichnet ist zum einen die Tatsache, dass sie die einzige Bank ist, deren umweltfördernder Einsatz bereits in der Satzung festgehalten wurde. Zudem unterstützt die Umweltbank ausschließlich Projekte und keine Unternehmen.

GLS Bank

Entstehung und Entwicklung

Auch die GLS Bank hat eine ganz eigene Entstehungsgeschichte. Sie geht zurück auf eine

Gruppe Eltern, die eine Schule in Bochum bauen lassen wollten. Die nützliche Idee, wie das Vorhaben trotz Geldmangels und noch vor der Etablierung des heutzutage populären Crowdfundings umgesetzt werden könnte, kam von Wilhelm Ernst Barkhoff. Er schlug vor, dass sich die Eltern zusammenschließen sollten, um einen Kredit zu erhalten. Aus diesem Impuls entstanden weitere Projekte. 1974 kam es dann durch Barkhoff und einige Mitstreiter zur Gründung der ersten sozial-ökologischen Bank der Welt.

Wie auch die anderen Banken weist auch die GLS ein stetiges Wachstum auf. Im vergangenen Jahr ist laut eigener Angaben das Kundengeschäftsvolumen um 14 Prozent gewachsen, was der durchschnittlichen Größe einer Volksbank entspricht. Die GLS ist sehr zuversichtlich und ist davon überzeugt, dass sich die Gesellschaft momentan in einem Umdenkprozess befindet: »Energiewende, Bio-Lebensmittel, Elektroautos – überall ist Bewegung. Dann fragen die Menschen sich auch irgendwann, was mit ihrem Geld passiert«, erhalte ich schriftlich als eindeutiges Statement.

Angebote und Kundengruppe

Die GLS Bank vergleicht ihre Angebote mit einer Hausbank. Girokonten, GLS Bank Anteile, Baufinanzierungen, Beteiligungen und Fonds stellen wichtige Komponenten ihrer Arbeit dar.



Soziale Projekte

Viele der Banken wählen nach strengen Kriterien aus, wem sie Geld geben.

Um die Attraktivität für junge Menschen zu erhöhen, bietet die GLS zusätzliche Angebote für die junge Zielgruppe: Das Starterkonto (14-17 Jahre) und das Junge Konto (18-27) mit besonders attraktiven Konditionen.

Service

Wer ein Konto bei der GLS hat, kann sich an über 16.000 Geldautomaten der Volks- und Raiffeisenbanken versorgen.

Mit der GLS BankCard kann zudem kostenfrei an über 18.500 Geldautomaten der Banken abgehoben und eingezahlt werden, die am BankCard ServiceNetz (BSN) teilnehmen. Auch eine Online-Automatensuche steht zur Verfügung.

Die GLS Bank versucht, über provisionsfreie Beratungen einen besonders ehrlichen und bedürfnisorientierten Service anzubieten. Sieben Filialen sowie ein telefonischer Kundenservice und Informationsmöglichkeiten in den sozialen Medien (u.a. Facebook, Twitter, Instagram) stehen hierfür zur Verfügung. Darüber hinaus gibt es einen Chat, einen

Kontowechselservice, Video-Ident und eine Mobile Banking App.

Projekte, Kunden und Sonstiges

Auch hier kann ich nicht direkt bestimmen, dass mein Geld genau dem Bauern von nebenan zugutekommt, die GLS Bank veröffentlicht aber, wie sie das Geld ihrer Kunden verwendet, und stellt Informationen im Internet sowie in ihrem Kundenmagazin zur Verfügung. Auch einzelne Unternehmen und geförderte Projekte werden detailliert vorgestellt und es wird regelmäßig über die positive Wirkung des Geldes berichtet. Zusätzlich kann man die Auswahl der Projekte, die unterstützt werden sollen, einschränken. Der Kunde kann bestimmen, in welchem Bereich sein Geld wirken soll, z.B. ökologische Ernährung oder erneuerbare Energien. Die Zahl der unterstützten Projekte stimmt interessanterweise mit der der Umweltbank überein: 22.000 Projekte werden aktuell unterstützt.

Darüber hinaus hat jeder Anleger genau eine Stimme in der genossenschaftlichen Ver-

einigung. Unabhängig vom Geld kann also jeder über die Arbeitsprozesse und Förderungsziele der Bank mitbestimmen. Hierzu wird regelmäßig zu Informationsveranstaltungen und Sitzungen eingeladen. Rund 215.000 Kunden kann die GLS vorweisen. Alleine im vergangenen Jahr konnten 29.000 Neukunden hinzugewonnen werden. Die GLS setzt sich nicht nur für Umweltprojekte, sondern auch für soziale Aspekte und den bewussten Umgang mit Geld ein.

EthikBank

Entstehung & Entwicklung

Die EthikBank wurde im Jahr 2002 als Zweigniederlassung der Volksbank Eisenberg eG von Sylke Schröder und Klaus Eisenberg gegründet. Bereits 1996 wurde die Volksbank Eisenberg DIREKT aus der Wiege gehoben. Seit Januar 2001 fördert diese aus Kundenspenden und Eigenmitteln je ein Projekt aus dem Bereich der Frauenförderung, ein Ethik- und ein Umweltprojekt. Die Konzentration liegt bei der neuen Bank somit auf dem ethisch-ökologischen Gesichtspunkt.

Auch heute noch wird ein hybrides Geschäftsmodell betrieben. Dabei ist die Volksbank Eisenberg als Filialbank für Kunden in der Region und die EthikBank als Direktbank für Kunden in ganz Deutschland vorgesehen. Die EthikBank kann auf erfolgreiche Jahre zurückblicken. Nicht nur vervielfachten sich die Kundenzahlen, auch konnte sie zudem 2013 den Erfolg verzeichnen, als zweitplatzierte Bank auf der Liste der beliebtesten Banken der Zeitschrift *€uro* gelistet zu werden. Aus der Finanzkrise ging auch sie gestärkt hervor.

Angebote und Kundengruppe

Die EthikBank weist ein breites Angebot auf. So können z.B. gleichermaßen Kredite beantragt oder ein Girokonto eröffnet werden. Hier

finden sich z.B. der Ökokredit, eine nachhaltige Baufinanzierung und der FAIRbraucherkredit. Als Geldanlagen hat man die Wahl zwischen einem Zinskonto, Festgeld, Mäusekonto, Spargeld, Wachstumszertifikat, ...

Die Zinsen und Ansprüche sind nach eigenen Angaben marktgerecht und immer wieder betont die EthikBank ihre Überzeugung, dass sich nachhaltiges Wirtschaften auch in Zukunft auszahlen wird.

Die Gebühren werden dank des Verzichts auf Filialen und dadurch niedrigere Verwaltungskosten gering gehalten. Auch die Größe der Bank möchten sie bei einem überschaubaren Rahmen belassen, da die EthikBank die Qualitäten einer Bank nicht in ihrer Quantität widerspiegelt sieht.

Neben Privatkunden finden sich 20 Prozent Geschäftskunden bei der EthikBank.

Service

Auch die EthikBank arbeitet in der Organisationsform einer Direktbank. Der Service erfolgt daher größtenteils telefonisch, per E-Mail oder postalisch. Zusätzlich wird damit geworben, bei einem Anruf im Kundenzentrum nicht von einem anonymen Callcenteragent bedient zu werden, sondern von menschlichen Mitarbeitern.

Projekte, Kunden und Sonstiges

Die EthikBank gibt an, das Geld nach streng ethisch-ökologischen Gesichtspunkten anzulegen. Die Ausschlusskriterien stellen sie auf ihrer Internetseite folgendermaßen dar:

»Wir kaufen keine Wertpapiere oder geben Firmen keine Kredite, die Militärwaffen herstellen oder vertreiben, Atomkraftwerke besitzen, Pflanzen und Saatgut gentechnisch verändern, die Tierversuche (Kosmetika) machen oder Kinder zur Arbeit ausnutzen. Auch die Spekulation mit Nahrungsmitteln und Rohstoffen ist ein Tabu.«

Stattdessen gehe das Geld der Anleger an verantwortungsbewusste Firmen, umweltschonende Unternehmen, Pfandbriefe zur Immobilienfinanzierung, ökologische Baustoffe und erneuerbare Energien. Die Auswahl der Projekte und unterstützten Firmen erfolgt durch ein umfangreiches Ethik-Research.

Bei Staatspapieren vertrauen sie auf das Nachhaltigkeitsrating der Zürcher Kantonalbank, in Hinblick auf Menschenrechtsverletzungen auf die amerikanischen NGOs freedom house und bei Korruption hilft ihnen der Korruptionsindex von Transparency International in ihrer Auswahl.

Sehr wichtig ist ihnen das Konzept der »gläsernen Bank«. Kunden können zwar nicht über Investitionen direkt bestimmen, es ist ihnen aber möglich, alle Aktionen und Partner auf der Homepage nachzulesen. Möchte man aber doch ein Projekt im Speziellen fördern, so kann man dies durch Direktspenden, oder durch das Anlegen eines Förderkontos umsetzen.

Besonders ist auch, dass sich die EthikBank auf drei Projekte beschränkt, die sie nur selten wechseln und die alle im Ausland liegen. Nach neuesten Zahlen vom Dezember 2016 weist die EthikBank eine Kundenzahl von etwa 241.000 auf. Als Besonderheiten erwähnt die EthikBank die von den Kunden als positiv rückgemeldete persönliche Kommunikation trotz Direktbankstruktur. Die Transparenz sei - selbst im Vergleich zu anderen Alternativbanken - besonders hoch. Als Beleg führt die EthikBank an, die erste Bank in Deutschland gewesen zu sein, die ihre Kapitalmarktgeschäfte offengelegt hat. Darüber hinaus ist es bisher noch die einzige Bank, die unabhängigen Wirtschaftsprüfern regelmäßig einen besonderen Prüfungsauftrag erteilt und die entstehenden Prüfungsberichte veröffentlicht.

DKB

Entstehung und Entwicklung

Die DKB wurde nach der Wende als Aktiengesellschaft und erste private Bank der DDR gegründet. Seit dem vollständigen Kauf 1994 handelt es sich bei der DKB um eine Tochtergesellschaft der Bayrischen Landesbank. Der Privatkundenzuwachs von 117% aus dem Jahr 2008 spricht bei der Frage der Finanzkrisenauswirkungen für sich. Auch in den folgenden Jahren lag der Zuwachs zwischen 8 und 20%.

Angebote und Kundengruppe

Bei der DKB wird man gut in der Breite fündig. Ein Girokonto kann problemlos eingerichtet werden, Ersparnisse werden angelegt, Wertpapiere stehen zum Angebot und auch Kredite werden von dieser Bank vergeben. Die DKB bietet zudem besondere Konditionen für Minderjährige, wie z.B. Das DKB-Cash u18 an. Die DKB konzentriert sich zudem auf ausgewählte Kundengruppen in den Bereichen Privatkunden, Firmenkunden und öffentliche Kunden.

Service

Die DKB fügt sich in unser bisheriges Bild der vorgestellten Banken gut ein, denn auch diese Bank beschränkt sich auf den Betrieb sehr weniger Filialen.

Von den Angeboten sticht das Girokonto DKB Cash als beliebtestes hervor. Darüber hinaus werden Sparprodukte, Ratenkredite und Immobilienfinanzierungen in besonderem Maße genutzt. Mit einer DKB Visacard kann laut Internetauftritt Geld an über 1 Mio. Schaltern weltweit kostenlos abgehoben werden, solange der Automatenbetreiber selbst keine Gebühren verlangt.

Projekte, Kunden und Sonstiges

Besonders viele Projekte, die von der DKB unterstützt werden, liegen im Sektor der erneuerbaren Energien, wie z.B. der Bau von Windparks oder Solaranlagen, den sie seit 1996 mitfinanziert. Insgesamt geht ein Großteil der 80% der Bilanzsumme, die »umweltfreundlich & sozial« eingesetzt wird, an mittelständische und kommunale Unternehmen sowie Privatkunden zur Schaffung von Wohnräumen und den Bau von Kitas und Schulen. Einmal jährlich erfolgt ein »umfangreiches Reporting« über die unterstützten Projekte.

Neben diesen umweltfreundlichen Anlagen wirbt die DKB zusätzlich damit, bereits im eigenen Betrieb ressourcen- und energieschonend zu Arbeiten. Laut aktuellsten Zahlen aus dem Juni 2017 kann die DKB auf eine Kundschaft von etwa 3,6 Mio. Anlegern blicken. Ein besonderer Fokus der DKB liegt seit 2016 auf den Green Bonds, mit dem Wind-Photovoltaik- und Biogasanlagen unterstützt werden. Die Basis dafür bildet das Kreditportfolio der DKB im Bereich erneuerbarer Energien - mit einem Volumen von rund 10 Mrd. € gehört es laut sustainability bond rating von oekom zu den größten in Deutschland.

Der erste Green Bond der DKB ist derzeit die am besten bewertete grüne Anleihe weltweit.

Fazit

Ich hoffe abschließend, dass ich euch eine übersichtliche Darstellung der verschiedenen Banken und Angebote mit an die Hand geben konnte und es euch nun leichter fällt, euch für ein spezielles Konzept zu entscheiden. Wenn ihr immer noch Fragen habt: Am besten informieren und beraten euch die Angestellten der einzelnen Banken selbst. Wenn ihr deren Empfehlungen mit der notwendigen Reflexion betrachtet und die einzelnen Angebote für eure spezielle Lage und eure Ziele vergleicht, findet ihr die individuell beste Anlagemöglichkeit.

Ihr geht kommendes Semester ins Ausland? Für diesen Fall kann ich euch die Global alliance on banking for values empfehlen, ein Zusammenschluss von Geldinstituten, die weltweit nachhaltige Kapitalanlagen anbieten.

Alternativangebote bei konventionellen Banken?

Jetzt habe ich ökologische Banken untereinander verglichen. Aber was ist mit Alternativangeboten bei konventionellen Banken, die es sich zum Ziel gesetzt haben, den Trend abzugreifen und grüne Investitionsmöglichkeiten zu schaffen? Die von Grund auf »grünen« Banken versuchen sich hier bewusst von den Angeboten abzugrenzen. Ein Investmentfond einer herkömmlichen Bank sei zwar besser als gar kein grünes Anlageprodukt, aber der Unterschied liege im Geschäftsmodell, argumentiert beispielsweise die EthikBank.

Alternative Banken investieren alle Kundeneinlagen auf Basis ihrer ethischen und ökologischen Grundhaltung, an der sie ihre Anlagepolitik ausrichten. Diese Anlagepolitik umspanne die gesamte Bank, während bei konventionellen Banken maximal ein kleiner Bruchteil der Kundeneinlagen in die sozial-umweltfreundlichen Bereiche investiert wird und die Gewinnmaximierung im Vordergrund steht.

Natürlich gibt es auch Gegenstimmen, die gegenüber der »Gutmenschanken« eine gewisse Skepsis hegen. Wird das Geld wirklich ökologisch und sicher angelegt? Kann ich mein Geld auch an genügend Automaten abholen?

Das Thema Geld ist also kein leichtes und man sollte sich genauestens informieren, bevor man eine Anlageentscheidung fällt.

Habt ihr bereits Erfahrungen mit einer der erwähnten Banken gemacht? Oder kennt ihr andere »alternative« Angebote? Dann teilt eure Meinung auf...

– *Lilith Diringer*

Monaden sind wie Burritos

Wer schon mal von der funktionalen Programmiersprache Haskell gehört hat, wird sicher genauso fasziniert wie von dem Wort Monaden abgeschreckt sein. Dabei sind Monaden gar nicht so zentral und für den Anfänger wichtig, wie man vielleicht denkt.

Schon im Jahr 2003 sagt deshalb Simon Peyton Jones, ein Mitglied der Arbeitsgruppe, die Haskell ursprünglich 1990 definiert hat, deren größter Fehler sei es gewesen, Monaden nicht »warm fuzzy things« (warme flauschige Dinge) zu nennen. Doch was geschehen ist, ist geschehen und so ist das Ziel eines jeden frischgebackenen Haskell-Programmierers, endlich *Monaden* zu verstehen. Nicht wenige sind von ihrem Erfolg, es endlich verstanden zu haben, dann so begeistert, dass sie erstmal ein Monaden-Tutorial schreiben. Ein fataler Fehler, meint Brent Yorgey (Informatikprofessor am Hendrix College), und schreibt in einem vielzitierten Blogbeitrag 2009 von der *monad tutorial fallacy*: Yorgey argumentiert, dass Menschen nur von konkreten Beispielen ausgehend abstrakte Definitionen verstehen können.

Als Gegenbeispiel erzählt Yorgey die fiktive Geschichte des Joe Haskell. Nachdem Joe sich eine Woche lang bemüht hat, Monaden zu verstehen, hat er endlich den Aha-Moment. Es macht klick und Joe hat endlich alles verstanden: *Monaden sind wie Burritos!*

Burritos

Lecker, aber auch gesund?

[W]hen presented with a mathematical definition for the first time, most people (me included) don't »get it« immediately: it is only after examining some specific instances of the definition, and working through the implications of the definition in detail, that one begins to appreciate the definition and gain an understanding of what it »really says.«

Brent Yorgey



Wenn nur jemand Joe früher gesagt hätte, dass Burritos der Schlüssel zum Verständnis sind! Deshalb macht Joe sich an die Arbeit, er schreibt ein Tutorial mit dem Titel »Monaden sind wie Burritos«, in dem er anderen Anfängern den Schlüssel zum Erfolg, nämlich Burritos, aufzeigt. Doch hier liegt laut Yorgey der fatale Fehler: Denn die Woche des Grübelns und Experimentierens war für Joes Lernerfolg notwendig. In dieser Woche hat sich Joe das nur für ihn passende Gedankenbild der Burrito-Monaden gebastelt. Dieser Lernprozess ist für jede Person anders, und wenn Joes Leser versuchen, Monaden als Burritos zu verstehen, werden sie wahrscheinlich frustriert scheitern. Der Trugschluss, dass andere Menschen von der eigenen Gedankenstütze profitieren, ist Kern der *monad tutorial fallacy* und laut Yorgey nicht nur beim Erklären von Monaden relevant: Ein guter Lehrer versuche nicht, anderen forciert seine eigene Intuition eines Konzept beizubringen, sondern hilft dem Schüler, sein eigenes Verständnis zu erlangen.

In diesem Artikel möchte ich versuchen, Euch Monaden zu erklären. Aber wie ihr gerade selbst gelesen habt, ist uns die trockene Definition erstmal egal. Wir wollen *bottom up* mit Beispielen anfangen, die Gemeinsamkeiten der Beispiele erkennen und letztendlich unsere Erkenntnisse zu einer Definition zusammentragen, die dann vielleicht auch etwas mit Monaden zu tun hat.

Übrigens müsst ihr jetzt das HPImgzn nicht zur Seite legen, nur weil Euch Haskell nicht interessiert, denn Haskell hat kein Patent auf Monaden. Mit anderen Worten: Das Konzept taucht überall auf und bereichert auch imperative Programmierer sowie JavaScript-Liebhaber (auch in JavaScript lassen sich Monaden darstellen).

Haskell Syntax Primer

Um die nachfolgenden Beispiele besser zu verstehen, kommt an dieser Stelle eine Kurzeinführung in die Haskell-Syntax.

- Funktionsanwendung wird – im Gegensatz zu den meisten anderen Sprachen – ohne Klammern oder Kommas geschrieben. Wenn man also die Funktion f mit den Werten a , b und c aufrufen möchte, schreibt man $f\ a\ b\ c$. Die Definition von Funktionen erfolgt ähnlich: Um die Funktion add mit den Parametern a und b zu definieren, schreibt man $add\ a\ b = a + b$. Alternativ kann man auch ein anonyme Funktion mittels einer Lambda-Abstraktion definieren. So könnte man add äquivalent so schreiben: $add = \lambda a\ b \rightarrow a + b$

- In Haskell können die Typen aller Werte und Funktionen mit einer `type annotation` genau definiert werden: Mit $\pi = 3.14159$ `:: Double` legt man fest, dass der Wert π den Typ `Double` haben soll. Bei Funktion kommt die Annotation vor die Definition, am Beispiel add

```
add :: Int -> Int -> Int
add a b = a + b
```

Wir haben hier allerdings eine engere Typdefinition für add gewählt als eigentlich nötig: Denn die Addition ($+$) ist für alle Typen definiert, die *Instanz* der *type class* `Num` sind (`Num` besteht aus einigen einfachen arithmetischen Operationen). *Type classes* geben eine Reihe von Funktionen vor, die von *Instanzen* der *Type class* implementiert werden müssen. So sind z. B. `Double`, `Int`, `Char` Instanzen von `Num`. Wir können add eine weiter gefasste Definition geben, indem wir angeben, dass add auf einem *beliebigen* Typ, der durch eine *Typvariable* a repräsentiert ist, arbeitet:

```
add :: Num a => a -> a -> a
add a b = a + b
```

- Haskell unterstützt sog. Algebraische Datentypen (ADTs). Dabei handelt es sich um eine Kombination sog. *Product types* (entsprechen z.B. structs aus C/C++), also Typen, die mehrere Felder aus verschiedenen Typen haben können, und sog. *Sum types* (entsprechen z.B. enums aus C/C++), bei denen ein Typ verschiedene Werte (in Haskell: Konstruktoren) annehmen kann. Das Ergebnis sind *Sum-of-product types*, also Typen, die verschiedene Konstruktoren haben können, die jeweils mehrere Felder aus verschiedenen Typen haben. Beispiele:

```
data Complex = Complex Double Double
data Bool = False | True
data Maybe a = Nothing | Just a
```

Während die Typen `Complex` (*Product type*) und `Bool` (*Sum type*) auch in fast allen anderen Sprachen realisierbar sind, kann `Maybe` nicht direkt dargestellt werden (C++17 hat ein Äquivalent namens `std::optional<T>`, das aber nicht *first class* ist). Der Typ `Maybe` ist entweder `Nothing`, d.h. der Wert ist nicht vorhanden, oder `Just a`, also vorhanden.

Funktoren: Die Tortilla des Burritos

Zunächst einmal müssen wir uns mit einem grundlegenden und wichtigem Konzept beschäftigen: Die *Funktoren* (haben nichts mit dem gleichnamigem C++-Begriff zu tun). Als Beispiel nehmen wir den Datentyp `Liste`:

```
data List a = Empty | Cons a (List a)
```

Oha, in der Definition des Konstruktors `Cons` (für *construct*) taucht der Datentyp `List a` selbst wieder auf – wir haben es also mit einem *rekursiven Datentyp* zu tun. Meistens werden Listen in Haskell aber nicht mit der obigen Definition geschrieben, sondern mit der Kurzschreibweise `[1, 2, 3]` für eine Liste mit den Elementen 1, 2 und 3. Diese entspricht der Definition in der ausführlichen Form

```
Cons 1 (Cons 2 (Cons 3 (Empty)))
```

Listen sind durch ihre rekursive Definition eine ideale Datenstruktur für funktionale Programmiersprachen, da wir in diesen Funktionen häufig rekursiv definieren. Eine typische Funktion ist dabei das Anwenden einer Funktion auf jedes Element der Liste, `map :: (a -> b) -> [a] -> [b]` (oder `map :: (a -> b) -> List a -> List b`). Übrigens: Diese Funktion lässt sich mittlerweile auch in vielen imperativen Programmiersprachen wiederfinden, so z. B. `std::transform` in C++. Doch wie genau ist `map` definiert? Wir müssen zwei Fälle unterscheiden, entweder ist die Eingabeliste `Empty` oder sie ist `Cons a (List a)`. Im ersten Fall ist die Definition einfach: Wenn wir keinen Wert haben, auf den wir die Funktion `(a -> b)` anwenden könnten, können wir auch nicht viel tun! Damit können wir den ersten Fall schon mal definieren:

```
map :: (a -> b) -> List a -> List b
map _ Empty = Empty
```

Dabei steht `_` für einen Parameter, dessen Wert uns egal ist (wir brauchen die Funktion in diesem Fall schließlich gar nicht!). Im zweiten Fall kriegen wir den Wert `Cons a (List a)`. Da wir die Struktur der Liste erhalten wollen, müssen wir auch wieder einen `Cons`-Wert zurückgeben. Auf das erste Feld von `Cons`, also `a`, wollen wir unsere Funktion anwenden. Und für das zweite Feld, `List a`, schreiben wir ja gerade die Funktion, die genau das macht, was wir hier brauchen: An dieser Stelle rufen wir einfach wieder `map` auf!

```
map f (Cons x xs) = Cons (f x) (map f xs)
```

Damit ist unsere Definition von `map` fertig.

Nun können wir endlich den Begriff *Funktor* einführen: Jeder Datentyp `f a`, für den die Funktion `fmap` definiert ist, die den »inneren« Wert `a` mit einer beliebigen anderen Funktion

`g :: a -> b` verändert, ist ein Funktor. In Haskell können wir das Konzept Funktor ähnlich Num als typ class festhalten:

```
class Functor f where
  fmap :: (a -> b) -> f a -> f b
  pure :: a -> f a
```

Dabei ist `pure` eine Funktion, die einen beliebigen Wert in den Typ `f` liftet (»in eine höhere Ebene hebt«). Vielleicht kann man an dieser Stelle schon sehen, worauf das hinausläuft: Für Listen entspricht `fmap` einfach `map` und `pure` ist eine Funktion, die aus einem Wert eine einelementige Liste mit diesem Wert macht. Wir können daher auch schon die Instanz schreiben:

```
instance Functor List where
  fmap = map
  pure x = Cons x Empty
```

Wir kennen auch schon den Datentyp `Maybe`, der ebenfalls ein Funktor ist:

```
instance Functor Maybe where
  fmap Nothing = Nothing
  fmap f (Just x) = Just (f x)
```

Funktoren sind ein allgegenwärtiges Konzept. Die meisten bekannten Datenstrukturen sind Funktoren.

Maybe: Funktionen, die Scheitern dürfen

In der Syntaxeinführung haben wir `Maybe` bereits definiert, nun wollen wir uns den Datentyp mal genauer anschauen. `Maybe` wird in der Praxis meistens dazu eingesetzt, ein Ergebnis einer Berechnung darzustellen, das in manchen Fällen nicht definiert oder nicht vorhanden ist bzw. bei dem die Berechnung manchmal zu einem Fehler führt. Als einfaches Beispiel könnte man eine sichere

Warum Listen?

Der geneigte Algorithmiker wird sich an dieser Stelle vielleicht fragen, warum gerade Listen in Haskell so zentral sind. Denn wenn man sich die Struktur des Listentyps näher ansieht, erkennt man schnell, dass es sich um eine einfach verkettete Liste handelt – mit entsprechend ungünstiger linearer Laufzeit für wahlfreien Zugriff und Einfügen. In den meisten imperativen Programmiersprachen sind Arrays, die mit wahlfreiem Zugriff in konstanter Zeit Listen häufig überlegen sind, syntaktisch stark verwurzelt. Warum sind Listen dann so zentral in die Sprache eingebaut?

Der Grund: Listen werden in Haskell nicht primär als Datenstruktur zur (langfristigen) Speicherung verwendet, sondern dienen zur effizienten Verkettung mehrerer Funktionen. Man könnte sagen, Listen stellen funktionale »Schleifen« dar: Genau wie es in einer imperativen

Schleife immer entweder einen weiteren Durchlauf oder einen Abbruch gibt, ist eine Liste entweder leer oder sie hat mindestens ein Element (zu verarbeiten) übrig. Diese Analogie wird besonders bei der besprochenen Implementierung von `map` offensichtlich.

Diese Tatsache macht sich der Haskell-Compiler zunutze, um die sogenannte *list fusion* durchzuführen, bei der mehrere hintereinander auszuführende Listen-Operationen zu einer einzigen zusammengefügt werden. Der Optimierer kann beispielsweise Sortiervorgänge verschieben, um weniger Elemente sortieren zu müssen, wenn vorhergehende Operationen Elemente aus der Liste entfernen (ein gängiger RDBMS-Query Optimization-Trick). In imperativen Sprachen ist das nicht möglich, da der Compiler die Äquivalenz nicht beweisen kann.

Division definieren, die bei einer Division durch 0 keine Divide-by-0-Exception auslöst:

```
safeDiv :: Num a => a -> a -> Maybe a
safeDiv _ 0 = Nothing
safeDiv x y = x / y
```

Damit werden wir `Maybe` aber nicht gerecht. Denn das Konzept eines eventuell nicht vorhandenen Ergebnisses ist beinahe überall zu finden. Besonders wenn wir IO durchführen, indem wir z.B. eine REST-API ansprechen oder eine Datei öffnen, gibt es stets die Möglichkeit, dass ein Fehler beim Zugriff entsteht. Gerade in C hat daher IO-Code ein typisches Muster:

```
int fd = open("some/file", flags);
if (fd == -1) {
    perror("Cannot open file");
    exit(1);
}
ssize_t written = write(fd, buf, size);
if (written == -1) {
    perror("Cannot write to file");
    exit(1);
}
```

Dabei wird stets ein Systemaufruf durchgeführt, das Resultat auf einen Fehlerwert geprüft (meistens `0` oder `-1`) und im Falle eines Fehlers das Programm abgebrochen. Ein weiteres vergleichbares Muster ist die Nutzung des Nullpointers für Werte, die nicht vorhanden sind.

Zurück zu Haskell: Analog zu `safeDiv` kann man natürlich auch ein `safeLog` (und `safeSqrt`, `safeAsin` usw.) definieren:

```
safeLog :: (Floating a, Ord a) => a ->
Maybe a
safeLog x | x > 0 = Just (log x)
safeLog _ = Nothing
```

Hierbei steht die `type class Floating` für Gleitkommatypen (`Float` und `Double`) und

`Ord` für Totalordnungen (also Datentypen, deren Werte man vergleichen kann).

Wie können wir aber `safeLog` und `safeDiv` verbinden, sodass wir z.B. $f(x) = 1/\log(x)$ berechnen können? Die naive Variante ist etwas umständlich:

```
safeRecipLog :: (Floating a, Ord a) =>
a -> Maybe a
safeRecipLog x = case safeLog x of
    Nothing -> Nothing
    Just x' -> safeDiv 1 x'
```

Wir können zwei beliebige Funktion mit dem Kompositionsoperator `(.) :: (b -> c) -> (a -> b) -> (a -> c)` komponieren (`g . f` wendet erst `f` und dann `g` an), so können wir mit `recip . log` bereits die unsichere Variante von `safeRecipLog` definieren (`recip` ist die multiplikative Umkehrfunktion, d.h. $\text{recip } x = 1 / x$). Das bringt uns aber hier nichts, weil `safeLog` den Typ `Maybe a` zurückgibt, wir aber als Parameter von `safeDiv 1` den Typ `a` erwarten. Wir können also die beiden Funktionen nicht regulär komponieren. Tatsächlich gibt es aber Abhilfe: Mit der sog. Kleisli-Komposition (`<=<`) (nach dem Schweizer Mathematiker Heinrich Kleisli benannt) können wir `safeLog` und `safeDiv` komponieren:

```
safeRecipLog = (\x -> safeDiv 1 x)
               <=< safeLog
```

oder, vereinfacht:

```
safeRecipLog = safeDiv 1 <=< safeLog
```

Wie könnte `<=<` definiert sein? Versuchen wir, unsere erste Definition von `safeRecipLog` zu verallgemeinern:

```
(<=<) :: (b -> Maybe c) ->
      (a -> Maybe b) ->
      (a -> Maybe c)
g <=< f = \x -> case f x of
    Nothing -> Nothing
    Just x' -> g x'
```

Zustandsbehaftete Aktionen mit State

Wir gehen nun einen Schritt weiter und überlegen uns, wie wir zustandsbehaftete Berechnungen in Haskell darstellen können. Kann das überhaupt gehen, wenn Haskell einem seiteneffektfreie Funktionen verspricht, die keinen Zustand haben? Die Antwort lautet ja, denn wir reden hier nicht über einen allesumfassenden, latenten Zustand, der das Programm umgibt, sondern über einen zeitlich und räumlich begrenzten Zustand. Wie das geht, werden wir gleich sehen. Wir definieren zunächst einen neuen Datentyp:

```
data State s a = State { runState :: s
  -> (a, s) }
```

Wir sehen hier ein neues syntaktisches Konstrukt, die sog. *record syntax*, mit der man einzelnen Feldern eines Datentyps Zugriffsfunktionen hinzufügt. Wir können also mit der Funktion `runState :: State s a -> (s -> (a, s))` auf den Wert des Konstruktors `State` zugreifen (wobei der Wert wiederum eine Funktion mit der Signatur `s -> (a, s)` ist). Die Typvariable `s` steht für den *Zustandstyp*, die Typvariable `a` für den Rückgabebetyp. Wir können mit den Funktionen `get`, `put` und `modify` den Zustand manipulieren:

```
get :: State s s
get = State (\s -> (s, s))
put :: State s ()
put s = State (\_ -> ((), s))
modify :: (s -> s) -> State s ()
modify f = State (\s -> ((), f s))
```

`get` gibt konstant den aktuellen Zustand zurück, `put` setzt den Zustand auf einen neuen Wert und `modify` verändert den Zustand mit einer Funktion. Dabei sind `get`, `put` und `modify` *Aktionen*, die im Kontext eines Zustands ausgeführt werden. Jetzt erstmal ein einfaches Beispiel:

```
squareInt :: State Int ()
squareInt = do
  currentInt <- get
  put (currentInt ^ 2)
```

Die Funktion `squareInt` ist im Kontext eines Zustands mit dem Zustandstyp `Int` definiert. Sie liest erst den aktuellen Zustand aus und schreibt dann das Quadrat des Zustands zurück. Dazu haben wir hier die sogenannte *do-Notation* genutzt, die *syntactic sugar* für eine etwas ausführlichere Version ist, die wir nachher kennenlernen werden. Die *do-Notation* erlaubt es, *Aktionen* im imperativen Stil auszuführen (eigentlich werden sie nur auf eine besondere Weise verkettet, wie wir ebenfalls nachher sehen werden).

```
square1to100 :: Int
square1to100 = snd $ runState 0 do
  mapM_ (\x -> modify (+ x)) [1..100]
  squareInt
```

Was haben wir hier gemacht? In der Funktion `square1to100` berechnen wir zunächst die Summe der Zahlen zwischen 1 und 100, indem wir auf die Zahlen von 1 bis 100 (`[1..100]`) die anonyme Funktion `\x -> modify (+ x)` abbilden, die den Parameter zum Zustand hinzuaddiert. Die Funktionsweise von `mapM_` kann uns im Moment egal sein, zum Verständnis reicht es zu wissen, dass `mapM_` vergleichbar mit `map/fmap` ist. Anschließend quadrieren wir den Zustand mit `squareInt`. Wie kann man jetzt aus dem Typ `State Int ()` das eigentliche Ergebnis herausfinden? Wir müssen unseren Automaten »laufen lassen«, indem wir ihm einen Anfangszustand geben: `runState 0 do ...` `runState` liefert uns allerdings ein Tupel `(((), Int)`, aus dem wir noch mittels `snd $` das zweite Element extrahieren (`snd` gibt das zweite Element eines Zweiertupels zurück, `$` ist hier syntaktisch notwendig, aber für uns nicht weiter interessant). Damit haben wir erfolgreich

eine zustandsbehaftete Funktion ausgeführt, ohne Seiteneffekte nach außen dringen zu lassen.

Aber was passiert hinter den Kulissen – welche Umwandlung macht `do` für uns im Hintergrund? Wir können `squareInt` auch ohne `do` schreiben:

```
squareInt = get >>= (\currentInt -> put
  (currentInt ^ 2))
```

Dabei nutzen wir den sogenannten Bind-Operator (`>>=`) :: `State s a -> (a -> State s b) -> State s b`, der den Wert vom Typ `a` in `State s` als *ungelifteten* Parameter (*ohne* `State s`) einer anderen Funktion zur Verfügung stellt, die dann wiederum ein Ergebnis in `State s` produziert. In Verbindung mit der Version in `do`-Notation erschließt sich auch der Name `Bind`: Der Operator `>>=` bindet das Ergebnis einer Berechnung an einen konkreten Wert, den wir als Eingabe einer anderen Funktion nutzen können.

An dieser Stelle wird das Geheimnis gelüftet: Der Bind-Operator ist Teil der Definition von Monaden, und `State` sowie `Maybe` sind Monaden. Genauer gesagt: Jeder Datentyp, der Funktor ist, für den (`>>=`) definiert ist und für den gewisse Regeln (die uns aber hier nicht weiter interessieren) gelten, ist Monade. Das wollen wir erstmal festhalten:

```
class Functor m => Monad m where
  (>>=) :: m a -> (a -> m b) -> m b
instance Monad Maybe where
  Nothing >>= _ = Nothing
  Just x >>= f = f x
instance Monad (State s) where
  m >>= f = State $ \s ->
    let (a, s') = (runState m) s
    in (runState (f a)) s'
```

Die Instanz für `Maybe` ist noch relativ einfach, aber was geht bei der Instanz für `State s` vor? Wir definieren unser Ergebnis wieder als

`State` mit einer anonymen Funktion, die den vorherigen Zustand entgegennimmt (Erinnerung: `State` ist als Funktion `s -> (a, s)` definiert!) Dabei extrahieren wir zunächst mittels `runState m` diese Funktion aus der vorhergehenden Aktion `m` im `State`-Monad (also vom Typ `State s a`) und führen dessen »Zustandsfunktion« auf dem vorhergehenden Zustand `s` aus, womit wir als Ergebnis das Tupel `(a, s')` kriegen. Das `let ... in ...`-Konstrukt erlaubt es uns, diesem Zwischenergebnis einen Namen zu geben. Der eigentliche Wert des Ausdrucks steht dabei nach dem `in`. In diesem Fall extrahieren wir dabei wiederum die Zustandsfunktion aus dem `f a`, also der Anwendung der Funktion `f` (mit Typ `a -> State s b`) auf das Ergebnis `a` der vorhergehenden Aktion `m`. Dieser Zustandsfunktion übergeben wir den Folgezustand `s'` (der Zustand, der nach der Aktion `m` besteht).

Was genau dieses *Binden* bedeutet, kann man übrigens nicht allgemein sagen: Es hängt vom Datentyp ab. Bei `State` entspricht `Bind` einer sequentiellen Komposition zweier zustandsbehafteter Operationen. Bei `Maybe` propagiert `Bind` ein Ergebnis bzw. einen Fehler bei der Berechnung transparent weiter, ohne dass sich einzelne Funktionen Gedanken darüber machen müssen, was passiert, wenn die Eingabeparameter nicht vorhanden sind. Wie wir aber gleich sehen werden, kann `Bind` noch eine ganz andere Semantik annehmen.

Nichtdeterminismus durch Listen

Als letztes Beispiel dienen uns Listen. Genauer gesagt arbeiten wir im Folgenden mit einer vielleicht unerwarteten Interpretation von Listen: Wir verstehen die Funktion `a -> [b]` nicht mehr als eine Funktion, die auf eine Eingabe vom Typ `a` eine Liste mit Werten vom Typ `b` zurückgibt, sondern eine Funktion, die sich bei einer Eingabe vom Typ `a` »nicht entscheiden kann«, welchen Wert sie zurückgibt.

**Folgezustände:
Menge oder Liste?**

Eigentlich bräuchten wir hier eine Menge von Folgezuständen. Haskell kennt auch den Datentyp `Set`, für den man die folgenden Betrachtungen äquivalent führen kann, darauf wurde aber der Einfachheit halber verzichtet.

Diese Idee ist eng verwandt mit den Nichtdeterministischen endlichen Automaten (NFAs), wie man sie aus der Theoretischen Informatik kennt: Der NFA kann bei einer Eingabe frei aus mehreren möglichen Übergängen wählen. Er akzeptiert ein Eingabewort genau dann, wenn es eine Folge von Zustandsübergängen gibt, die zu einem akzeptierenden Zustand führen. Das kann man auf zwei verschiedene Arten verstehen: Entweder geht man davon aus, dass der Automat alle möglichen Wege gleichzeitig nimmt, oder man lässt den Automaten vom akzeptierenden Zustand aus immer den richtigen Übergang nehmen. Wenn Q die Menge der Zustände und Σ das Eingabealphabet ist, dann können wir die Übergangsfunktion δ direkt in Haskell modellieren:

```
 $\delta :: \Sigma \rightarrow Q \rightarrow [Q]$ 
```

(Die Reihenfolge vom ersten und zweiten Parameter ist hier gegenüber der gängigen Definition vertauscht.) Die Übergangsfunktion ordnet also jeder Eingabe und jedem Zustand eine Liste von Folgezuständen zu. Mit dieser Definition können wir direkt auch einen NFA basteln. So kriegen wir zum Beispiel mit `[s] >>= \delta 'N' >>= \delta 'F' >>= \delta 'A'` die Zustände, die ein NFA mit Startzustand `s` und Eingabe "NFA" hat. Der Bind-Operator hat bei Listen den Typ `(>>=) :: [a] -> (a -> [b]) -> [b]`, konkret führt er also auf eine Ausgangsliste eine expandierende Funktion, die jedem Wert eine Liste von Ausgabewerten zuweist, aus und gibt schlussendlich eine Liste von Ausgabewerten zurück. Die expandierende Funktion ist hier die Zustandsübergangsfunktion, könnte aber auch ganz anders aussehen.

Aber mit Listen sind wir natürlich nicht auf die Implementierung von NFAs beschränkt. Daher nun ein konkretes Beispiel:

```
cartesianProduct :: [a] -> [b] -> [(a, b)]
cartesianProduct xs ys = do
  x <- xs
```

```
y <- ys
pure (x, y)
```

Wie der Name verrät, haben wir uns hier das kartesische Produkt gebastelt – aber warum funktioniert das so? Für eine Antwort müssen wir etwas tiefer gehen und uns die äquivalente Form mit dem Bind-Operator ansehen:

```
cartesianProduct xs ys = xs >>= (\x ->
ys >>= (\y -> pure (x, y)))
```

Erklärung: Wir expandieren die Liste `xs` mit der Funktion, die wiederum `ys` mit einer Funktion, die einelementige Listen mit `(x, y)` zurückgibt, expandiert. Oder vereinfacht gesagt: Der Bind-Operator fungiert hier als Allquantor, der den rechtsseitigen Ausdruck für jedes Element der Liste evaluiert und die Ergebnisse sammelt.

Es gibt noch eine dritte Form, in die wir `cartesianProduct` bringen können:

```
cartesianProduct xs ys = [ (x,y) | x <- xs, y <- ys ]
```

Diese Syntax ist vergleichbar mit den *set comprehensions* aus der Mengenlehre, die jeder schon mal gesehen haben dürfte: $\{ (x,y) \mid x \in xs, y \in ys \}$. Tatsächlich heißt dieses Syntaxkonstrukt in Haskell *list comprehension* und unterstützt auch einschränkende Bedingungen. Du brauchst mal eben die Liste aller Primzahlen? Kein Grund in den Tisch zu beißen, die Lösung mit *list comprehensions* ist einfacher als man erwarten würde:

```
primes = [ p | p <- [2..], null [ f | f <- [2..(x-1)], p `mod` f == 0 ] ]
```

Diese Lösung entspricht auch der Lösung, die man intuitiv anhand der Definition von Primzahlen erwarten würde: Eine Zahl größer oder gleich 2 (`[2..]`), die keine Teiler außer sich selbst und 1 hat.

– Tobias Markus

Dr. Krohns Rechtstipps

Liebe HPI-ler,
in der neuen Ausgabe des HPImgzn eine Fortsetzung des Klassikers Rechtstipps mit einem Überblick über neueste Rechtsprechung mit Bezug zu Studieninhalten. Diese sind:

LG Köln, Urteil vom 3. April 2018 – 31 O 179/17

Das LG Köln hat entschieden, dass ein Domain-Treuhänder, der eine Domain für ein russisches Unternehmen registriert, für über den aufrufbaren Online-Shop begangene Markenrechtsverletzung haftet. Der für eine Anwendung deutschen Rechts notwendige Inlandsbezug folgt schon aus dem Umstand, dass es sich um eine .de-Domain handelt.

BGH, Urteil vom 12. Juli 2018 – III ZR 183/17

Der III. Zivilsenat des Bundesgerichtshofs hat entschieden, dass der Vertrag über ein Benutzerkonto bei einem sozialen Netzwerk grundsätzlich im Wege der Gesamtrechtsnachfolge auf die Erben des ursprünglichen Kontoberechtigten übergeht und diese einen Anspruch gegen den Netzwerkbetreiber auf Zugang zu dem Konto einschließlich der darin vorgehaltenen Kommunikationsinhalte haben.

Der Sachverhalt: Die Klägerin ist die Mutter der im Alter von 15 Jahren verstorbenen L. W. und neben dem Vater Mitglied der

Erbengemeinschaft nach ihrer Tochter. Die Beklagte betreibt ein soziales Netzwerk, über dessen Infrastruktur die Nutzer miteinander über das Internet kommunizieren und Inhalte austauschen können.

2011 registrierte sich die Tochter der Klägerin im Alter von 14 Jahren im Einverständnis ihrer Eltern bei dem sozialen Netzwerk der Beklagten und unterhielt dort ein Benutzerkonto. 2012 verstarb das Mädchen unter bisher ungeklärten Umständen infolge eines U-Bahnunglücks.

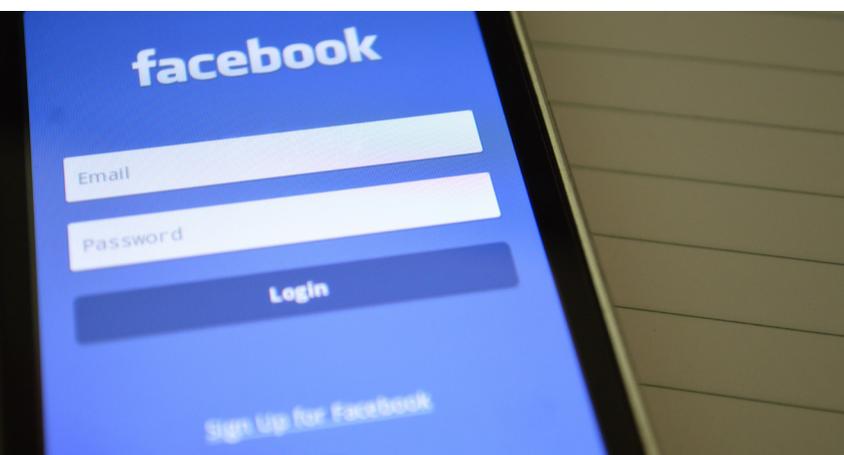
Die Klägerin versuchte hiernach, sich in das Benutzerkonto ihrer Tochter einzuloggen. Dies war ihr jedoch nicht möglich, weil die Beklagte es inzwischen in den sogenannten Gedenkzustand versetzt hatte, womit ein Zugang auch mit den Nutzerdaten nicht mehr möglich ist. Die Inhalte des Kontos bleiben jedoch weiter bestehen.

Die Klägerin beansprucht mit ihrer Klage von der Beklagten, den Erben Zugang zu dem vollständigen Benutzerkonto zu gewähren, insbesondere zu den darin vorgehaltenen Kommunikationsinhalten. Sie macht geltend, die Erbengemeinschaft benötige den Zugang zu dem Benutzerkonto, um Aufschluss darüber zu erhalten, ob ihre Tochter kurz vor ihrem Tod Suizidabsichten gehegt habe, und um Schadensersatzansprüche des U-Bahn-Fahrers abzuwehren.

Die Entscheidung des Bundesgerichtshofs: Der BGH entschied, dass die Erben gegen die Beklagte einen Anspruch haben, ihnen den Zugang zum Benutzerkonto der Erblasserin und den darin vorgehaltenen Kommunikationsinhalten zu gewähren. Dies ergibt sich aus dem Nutzungsvertrag zwischen der Tochter der Klägerin und der Beklagten, der im Wege der Gesamtrechtsnachfolge nach § 1922 Abs. 1 BGB auf die Erben übergegangen ist. Dessen Vererblichkeit ist nicht durch die

Vererbbar

Nutzungsverträge mit sozialen Netzwerken gehen auf Erben über.



vertraglichen Bestimmungen ausgeschlossen. Die Nutzungsbedingungen enthalten hierzu keine Regelung. Die Klauseln zum Gedenkzustand sind bereits nicht wirksam in den Vertrag einbezogen. Sie hielten überdies einer Inhaltskontrolle nach § 307 Abs. 1 und 2 BGB nicht stand und wären daher unwirksam.

Auch aus dem Wesen des Vertrags ergibt sich eine Unvererblichkeit des Vertragsverhältnisses nicht; insbesondere ist dieser nicht höchstpersönlicher Natur. Der höchstpersönliche Charakter folgt nicht aus im Nutzungsvertrag stillschweigend vorausgesetzten und damit immanenten Gründen des Schutzes der Persönlichkeitsrechte der Kommunikationspartner der Erblasserin. Zwar mag der Abschluss eines Nutzungsvertrags mit dem Betreiber eines sozialen Netzwerks in der Erwartung erfolgen, dass die Nachrichten zwischen den Teilnehmern des Netzwerks jedenfalls grundsätzlich vertraulich bleiben und nicht durch die Beklagte dritten Personen gegenüberoffengelegt werden. Die vertragliche Verpflichtung der Beklagten zur Übermittlung und Bereitstellung von Nachrichten und sonstigen Inhalten ist jedoch von vornherein kontobezogen. Sie hat nicht zum Inhalt, diese an eine bestimmte Person zu übermitteln, sondern an das angegebene Benutzerkonto. Der Absender einer Nachricht kann dementsprechend zwar darauf vertrauen, dass die Beklagte sie nur für das von ihm ausgewählte Benutzerkonto zur Verfügung stellt. Es besteht aber kein schutzwürdiges Vertrauen darauf, dass nur der Kontoinhaber und nicht Dritte von dem Kontoinhalt Kenntnis erlangen. Zu Lebzeiten muss mit einem Missbrauch des Zugangs durch Dritte oder mit der Zugangsgewährung seitens des Kontoberechtigten gerechnet werden und bei dessen Tod mit der Vererbung des Vertragsverhältnisses.

Nach der gesetzgeberischen Wertung gehen auch Rechtspositionen mit höchst-

persönlichen Inhalten auf die Erben über. So werden analoge Dokumente wie Tagebücher und persönliche Briefe vererbt, wie aus § 2047 Abs. 2 und § 2373 Satz 2 BGB zu schließen ist. Aus erbrechtlicher Sicht besteht kein Grund dafür, digitale Inhalte anders zu behandeln.

Einen Ausschluss der Vererblichkeit auf Grund des postmortalen Persönlichkeitsrechts der Erblasserin hat der III. Zivilsenat ebenfalls verneint.

Auch das Fernmeldegeheimnis steht dem Anspruch der Klägerin nicht entgegen. Der Erbe ist, da er vollständig in die Position des Erblassers einrückt, jedenfalls nicht »anderer« im Sinne von § 88 Abs. 3 TKG.

Landgericht Berlin, Urteil vom 24. Mai 2018 – 52 O 101/18

Grundsätzlich müssen Instagram-Nutzer ihre Posts nicht als Werbung kennzeichnen, wenn sie das präsentierte Produkt selbst erworben haben und in keiner Beziehung zu dem herstellenden Unternehmen stehen. Gilt ein Instagram-Nutzer hingegen als sogenannter Influencer mit einer erheblichen Anzahl von Followern und verlinkt auf seinen Posts den Hersteller oder sogar Shops, auf denen die präsentierten Produkte käuflich zu erwerben sind und fördert somit fremden Wettbewerb, kann der Beitrag dennoch als geschäftliche Handlung verstanden werden, wodurch eine kommerzielle Kennzeichnung erfolgen muss. Wird dieser werbliche Hinweis unterlassen, liegt ein Wettbewerbsverstoß vor.

EuGH, Urteil vom 7. August 2018 – C 161/17

Die Einstellung einer Fotografie, die mit Zustimmung des Urhebers auf einer Website frei zugänglich ist, auf eine andere Website bedarf einer neuen Zustimmung des Urhebers. Durch ein solches Einstellen wird die Fotografie nämlich einem neuen Publikum zugänglich gemacht.

BGH, Urteil vom 23. August 2018 – III ZR 192/17

Der Bundesgerichtshof hat in dieser Entscheidung eine pauschale Servicegebühr von 2,50 € für die elektronische Übermittlung einer Eintrittskarte zum Selbstaussdrucken für unzulässig erklärt.

Ticketkäufer haben bei Bestellungen von Eintrittskarten für Konzerte, Sportevents oder andere Veranstaltungen im Internet häufig und über eine Vielzahl von Anbietern hinweg eine *print@home*-Option zur Auswahl.

Hierbei werden die Tickets nicht per Brief zugeschickt, sondern nach elektronischer Übermittlung, z.B. per E-Mail, am heimischen Rechner ausgedruckt. Eventim, Marktführer in der Ticketvermittlung, verlangt bisher für diese *ticketdirect*-Option pauschal eine »Servicegebühr« in Höhe von bis zu 2,50 €, und das, obwohl für die Übermittlung weder Porto- noch Materialkosten anfallen. Der BGH hat nun in letzter Instanz zu Gunsten der Verbraucherzentrale entschieden und die Revision von Eventim zurückgewiesen.

Das Urteil hat grundsätzliche Bedeutung und verbietet generell für alle Anbieter im Internet die Forderung nach einer Gebühr für das Selbstaussdrucken.

Aus für die Servicegebühr

Für das Selbstaussdrucken von Tickets darf keine Gebühr erhoben werden.

LG Frankfurt a.M., Urteil vom 16. August 2018 – 2-03 O 32/17

Das LG Frankfurt hat entschieden, dass der Nutzer eines unter einer Creative-Commons-Lizenz stehenden Bildes die Nutzungsrechte verliert, wenn er die nach der Lizenz geforderten Angaben nicht tätigt (z. B. Kopie der Lizenzbedingungen, Nennung des Bildtitels). Ein Schadensersatzanspruch gegen den Verletzer könne auch bestehen, jedoch finde die MFM-Tabelle keine Anwendung für die Berechnung der Höhe einer angemessenen Lizenzgebühr. Für einen kostenpflichtigen Lizenzerwerb eines unter einer Creative-Commons-Lizenz stehenden Bildes könne nach den Grundsätzen der Lizenzanalogie lediglich dasjenige gefordert werden, was vernünftige Parteien für eine Befreiung von den Vorgaben der Creative-Commons-Lizenzen vereinbart hätten, nicht jedoch das, was für ein anderes Bild, das nicht auch kostenlos genutzt werden kann, anzusetzen gewesen wäre.

BGH, Urteil vom 10. Juli 2018 – VI ZR 225/17-LG Braunschweig

Die Verwendung von elektronischer Post für die Zwecke der Werbung ohne Einwilligung des Empfängers stellt grundsätzlich einen Eingriff in seine geschützte Privatsphäre und damit in sein allgemeines Persönlichkeitsrecht dar.

Eine Kundenzufriedenheitsbefragung in einer Mail fällt auch dann unter den Begriff der (Direkt-)Werbung, wenn mit der E-Mail die Übersendung einer Rechnung für ein zuvor gekauftes Produkt erfolgt.

Oberlandesgericht Frankfurt am Main, Urteil vom 06. September 2018, Az. 16 U 193/17

Das Oberlandesgericht Frankfurt am Main hat entschieden, dass es Google nicht generell untersagt werden darf, ältere negative Presse-



berichte über eine Person in der Trefferliste anzuzeigen, selbst wenn diese Gesundheitsdaten enthalten. Es komme auch nach Inkrafttreten der DS-GVO darauf an, ob das Interesse des Betroffenen im Einzelfall schwerer wiegt als das Öffentlichkeitsinteresse. Das durch die DS-GVO anerkannte »Recht auf Vergessen« überwiegt entgegen einer Entscheidung des EuGH zum früheren Recht nicht grundsätzlich das öffentliche Informationsinteresse.

EuGH, Urteil vom 4. Oktober 2018 C-105/17

Ob man bei Online-Verkäufen als Gewerbetreibender eingestuft wird, darf nicht allein an der Zahl der angebotenen Artikel festgemacht werden. Ausschlaggebend sei dagegen, ob die Verkäufe Teil einer »gewerblichen, handwerklichen oder beruflichen Tätigkeit« seien.

Ein Verbraucher hatte bei einer Bulgarin über eine Online-Plattform eine gebrauchte Armbanduhr erworben, war aber nicht zufrieden und wollte den Kaufvertrag widerrufen. Die Verkäuferin weigerte sich, die Uhr zurückzunehmen und den Kaufpreis zu erstatten, wozu sie nur als Privatperson berechtigt gewesen wäre. Die bulgarische Verbraucherschutz-Kommission stufte sie danach aber als gewerbliche Händlerin ein, weil sie feststellte, dass die Frau noch acht weitere Verkaufsanzeigen veröffentlicht hatte.

Als Gewerbetreibende hätte die Verkäuferin aber ihren Informationspflichten nachkommen müssen, stellte die Verbraucherschutz-Kommission fest. In den Anzeigen hätte sie dazu unter anderem weitere Angaben zu ihrer Person machen und auf das Widerrufs- und Gewährleistungsrecht für Verbraucher hinweisen müssen. Weil sie das unterließ, habe sie eine Ordnungswidrigkeit begangen und müsse eine Geldbuße bezahlen.

Die Frau zog dagegen vor Gericht - und das stellte beim EuGH die Frage, ob jemand als Gewerbetreibender im Sinne der Richtlinie

über unlautere Geschäftspraktiken eingestuft werden könne, weil er »eine vergleichsweise hohe Zahl« von Verkaufsanzeigen einstellt.

Der EuGH betonte, dass die Gerichte von Fall zu Fall entscheiden müssten, ob eine Person als gewerblicher Verkäufer handle. Anhaltspunkte dafür seien unter anderem, »ob der Verkauf planmäßig erfolgte, ob er eine gewisse Regelmäßigkeit hatte oder mit ihm ein Erwerbszweck verfolgt wurde, ob sich das Angebot auf eine begrenzte Anzahl von Waren konzentriert«. Zudem müssten die Rechtsform sowie die technischen Fähigkeiten des Verkäufers berücksichtigt werden.

Eine Verkäuferin, die eine Reihe von Anzeigen gleichzeitig auf einer Website veröffentlichte, könne nur dann als Gewerbetreibende eingestuft werden, wenn sie im Rahmen ihrer gewerblichen, handwerklichen oder beruflichen Tätigkeit handle.

Gibt es Rechtsthemen, die für die nächste Ausgabe des HPImgzn von Interesse sein könnten? Wenn ja, gerne mir mitteilen.

– *Timm Krohn*





Bruder muss
los zu
Website new:
"hpimgzn.de"

Impressum

Bildrechte

Cover:

Spieleautomat: Tobias Markus

Space-Invader: Lukas Wagner

Rückseite: Daniel-Amadeus J. Glöckner

Inhaltsverzeichnis: Sven Köhler

Editorial: Tobias Markus

HPIspiele

Reverse Engineering Anno 1602:

Screenshots: Christian Flach

Anno-Sprites: Ubisoft

VR Gaming:

Screenshots: Felix Thiel

Keep Talking and Nobody Explodes:

Steel Crate Games

I Expect You To Die: Schell Games

SUPERHOT VR: SUPERHOT Team

Hintergrund: *Foto ist gemeinfrei*

A Card Game:

Proposal-Board: GitHub

Andere Aufnahmen: Team

HPIkultur

Sprachgeschichten: *Bild ist gemeinfrei*

Lost and Found:

Screenshots, aufgenommen am 6. Dezember

Rätsel:

Florian Fregien

Bilder in Collage: *Gemeinfrei*

Weihnachtsschaf: Lea Gerneth

Der Hahn und der Wolf: *Alle Fotos gemeinfrei*

HPIintern

Umfrage:

Silvan Verhoeven

Lukas Wagner

Fruchtradar:

Screenshots: Lukas Wagner

Alle anderen Fotos sind gemeinfrei

Interview Nachhaltigkeitsklub:

Milch: Lilith Diringen

Porträt Lilith: Claudia Lange

Interview mit Prof. Arnrich und Prof. Lippert:

Sven Köhler

HPIwissen

Armin empfiehlt:

Joana Bergsiek

Schaf: CC by-nc-nd 4.0 Lea Gerneth

Grün trifft Gold:

Alle Fotos gemeinfrei

Monaden sind wie Burritos:

Alle Fotos gemeinfrei

Dr. Krohns Rechtstipps:

Foto von Dr. Krohn: Florian Schmidt

HPIimgzn.de – der Blog des HPIimgzn:

Alle Fotos gemeinfrei

Redaktion

Redaktionsschluss: 21. Dezember 2018

Auflage: 600 Stück

Redaktion dieser Ausgabe:

Joana Bergsiek, Jonas Bounama, Noel

Danz, Lilith Diringen, Grit Fessel, Christian

Flach, Florian Fregien, Marcel Garus,

Leonard Geier, Ulrike Herwig, Lisa Ihde,

Isabell Kraus, Jonathan Kreidler, Tobias Markus,

Moritz Schneider, Maximilian Stiede, Silvan

Verhoeven, Lukas Wagner, Ronja Wagner,

Johannes Wolf

V. i. S. d. P.

Lisa Ihde, Tobias Markus, Lukas Wagner

Kontakt

per E-Mail an info@hpimgzn.de

Danksagungen

Wir danken unseren zahlreichen externen Autoren, der Öffentlichkeitsarbeit für das schnelle Lektorat und Sandra Willoh für ihre freundliche Unterstützung der Redaktion.

