

# Das Notizbuch

Eine physikalische Krimi-App im Sihlwald



---

Maturitätsarbeit von Rosina Weiß

Betreuer: Fabio Torriani

Kantonsschule Wiedikon

Klasse 6e

Dezember 2018



### 1. Zusammenfassung

In meiner Maturitätsarbeit habe ich mich damit befasst, wie man das Erlernen von Physik spannend und kreativ gestalten kann. Dazu habe ich ausgewählte Themen aus der Physik spielerisch in einen Kriminalfall eingebaut. Eine App für iOS leitet im Rahmen eines zwei- bis dreistündigen Spaziergangs eine Spielergruppe durch den Wildnispark Zürich im Sihlwald. An verschiedenen Orten im Wald müssen einfache Experimente und Aufgaben gelöst werden, um den Kriminalfall aufzuklären. Dazu gibt es kleine Verstecke mit den entsprechenden Materialien, die man durch ein Navigationsgerät in der App finden kann. Insgesamt befinden sich neun Stationen auf dem Rundgang, die ein unterschiedliches physikalisches Thema behandeln. Bei jeder Station bekommt man weitere Informationen über den Kriminalfall, welche am Schluss von den Nutzerinnen und Nutzern aufgelöst wird. Die App „Das Notizbuch“ habe ich mit der Programmiersprache Swift und mit der Programmierumgebung Xcode entwickelt, welche ich im Rahmen meiner Maturitätsarbeit neu gelernt habe.

In meinem schriftlichen Kommentar stelle ich nach der Einleitung und der Darstellung der Grundlagen die zwei Schwerpunkte der Arbeit vor. Im Kapitel Endprodukt werden die App, der Kriminalfall und der Parcours im Sihlwald vorgestellt. Danach beschreibe ich die Vorgehensweise, wie ich zu diesem Endprodukt gekommen bin. Abschliessend ziehe ich in der Reflexion mein Fazit und gebe einen Ausblick auf die Zukunft meiner App.

Der Anhang enthält den gesamten Programmcode und die Screenshots der App, administrative Dokumente für die Organisation des Parcours im Wildnispark Zürich, sowie Dokumente aus der App und Auszüge aus den geführten Interviews, die ich zur Vorbereitung auf das Thema geführt habe.



## 2. Vorwort

Bei der Suche nach dem für mich perfekten Maturitätsarbeitsthema wusste ich lange nicht, was ich verwirklichen wollte. So habe ich am Anfang alles, was mich interessiert, in einen Topf geworfen und probiert, ein Thema zu finden, welches möglichst alle meine Interessen abdeckt. Hier standen vor allem die Musik, die Fotografie, mein Interesse für Physik und Informatik, die Natur und mein Spass Gesellschaftsspiele zu spielen im Vordergrund.

Für mich war auch wichtig, dass ich etwas Neues ausprobieren. Sonstige Projekte, bei denen ich mir etwas Grösseres vorgenommen hatte, sind häufig gescheitert oder ich habe irgendwann aus Zeitgründen aufgehört. Doch bei einer Maturitätsarbeit kann man nicht einfach irgendwo aufhören, sondern ist in einem gewissen Mass dazu gezwungen weiterzumachen. Somit wollte ich ein Projekt mit etwas Unbekanntem starten, bei dem ich gezwungen wäre, es abzuschliessen. Im Zeitalter der Digitalisierung bin ich auf die Idee gekommen, eine App zu programmieren. Dies hatte ich noch nie in meinem Leben gemacht, jedoch wollte ich es schon immer einmal ausprobieren.

Auch konnte ich mich schnell dazu entschliessen, dass die Natur und die Physik eine Rolle in meiner Arbeit spielen sollten. So bin ich zu der Idee mit einem Spaziergang in der Natur in Kombination mit physikalischen Aspekten gekommen. Selber habe ich auf Wanderungen immer gerne Geocaches (siehe Kapitel 5.2, S.14) gesucht oder irgendwelche Rätsel gelöst, welche zu einer gelungenen Abwechslung beim Wandern geführt haben. So etwas Ähnliches wollte ich gerne machen. Lange habe ich überlegt, welche Teile der Physik ich in meine Arbeit einbinden möchte. Zum Beispiel etwas, das mit der Astronomie oder vielleicht mit den physikalischen Aspekten des Fotografierens zu tun hat.

Zuhause spielen wir viele Gesellschaftsspiele und mir haben Exitgames und Rätselspiele immer sehr gut gefallen. So schien es mir geeignet, diese miteinander zu verknüpfen, vor allem, da ich mir vorstellte, dass das sicher viel Spass machen würde. Schliesslich habe ich mich entschlossen, einen Kriminalfall zu entwickeln und diesen dann durch physikalische Phänomene und mit Hilfe der App aufklären zu lassen.

Zum Schluss konnte ich praktisch alle meine Interessen in irgendeiner Art und Weise in meine Maturitätsarbeit einbinden, bis auf mein grösstes Interesse, die Musik. Am Anfang fand ich dies ein bisschen schade, doch ich habe schnell gemerkt, dass es gut ist, einmal in eine ganz andere Welt einzutauchen und etwas völlig Neues auszuprobieren.

An dieser Stelle möchte ich noch anfügen, dass ich aus Platzgründen und zur vereinfachten Lesbarkeit bei Funktionsbezeichnungen nur die männliche Form erwähne, obwohl die weibliche Form jeweils auch gemeint ist.



## 3. Inhaltsverzeichnis

1. Zusammenfassung.....	3
2. Vorwort.....	5
3. Inhaltsverzeichnis.....	7
4. Einleitung.....	9
5. Grundlagen.....	11
5.1. Krimi-Aktivitäten.....	11
5.1.1. Polizeimuseum Hamburg.....	11
5.1.2. Krimi-Spiele.....	12
5.2. Outdoor Aktivität.....	14
5.2.1. Geocaching.....	14
5.2.2. Foxtrail.....	15
6. Das Endprodukt.....	17
6.1. Konzept der App.....	17
6.1.1. Übersichtsdisplay.....	17
6.1.2. Navigation.....	18
6.1.3. Notizbuchfunktion.....	18
6.1.4. Die Stationen.....	19
6.2. Kriminalfall.....	19
6.2.1. Geschichte.....	19
6.2.2. Charaktere.....	23
6.3. Beschreibung der einzelnen Posten.....	24
6.3.1. Posten 1: Einführung.....	24
6.3.2. Posten 2: Optik.....	26
6.3.3. Posten 3: Akustik.....	27
6.3.4. Posten 4: Aerodynamik.....	28
6.3.5. Posten 5: Astronomie.....	29
6.3.6. Posten 6: Verschlüsselung.....	29
6.3.7. Posten 7: Druck.....	30
6.3.8. Posten 8: Mechanik.....	30
6.3.9. Posten 9: Aufklärung.....	31
7. Vorgehensweise.....	33
7.1. Konzept.....	33
7.1.1. Route.....	33
7.1.2. Physik-Elemente.....	35
7.1.3. Story.....	36
7.1.4. Namensgebung und App-Icon.....	36

---

7.2.	Organisation im Sihlwald .....	37
7.3.	Design der App .....	40
7.3.1.	Platzierung der Elemente .....	40
7.3.2.	Farben.....	41
7.3.3.	Fotos.....	41
7.4.	Programmierung .....	45
7.4.1.	iOS-App.....	45
7.4.2.	Kleine Einführung.....	46
7.4.3.	Erlernen von Swift .....	48
7.4.4.	Technischer Aufbau der App.....	49
7.5.	Interviews.....	52
7.5.1.	Interview mit Heike Riel.....	52
7.5.2.	Interview mit Jan-Peter Meyn.....	53
7.5.3.	Interview mit Fritz Gassmann.....	53
7.6.	Testen der App .....	53
8.	Schlusswort.....	55
8.1.	Reflektion .....	55
8.2.	Ausblick.....	56
9.	Danksagung.....	57
10.	Quellen- und Literaturverzeichnis.....	59
11.	Abbildungsverzeichnis.....	61
12.	Authentizitätserklärung.....	63
13.	Anhang.....	65
13.1.	Programmcode.....	65
13.2.	App UIViewController .....	180
13.3.	Dokumente der App .....	203
13.3.1.	Personalakten .....	203
13.3.2.	Kaufverträge (mit Fälschungen) .....	209
13.3.3.	Papierflieger Anleitungen.....	214
13.4.	Wildnispark Zürich .....	229
13.4.1.	Exposé .....	229
13.4.2.	Koordinationsblatt .....	233
13.5.	Interviews.....	240
13.5.1.	Interview mit Heike Riel .....	240
13.5.2.	Interview mit Jan-Peter Meyn .....	248
13.5.3.	Interview mit Fritz Gassmann .....	252
13.6.	Arbeitsjournal .....	255

## 4. Einleitung

In meiner Maturitätsarbeit beschäftige ich mich damit, wie man Menschen für die Physik begeistern kann. Mein Ziel ist es, dies durch eine selbstprogrammierte App zu erreichen, welche innerhalb eines zwei- bis dreistündigen Spaziergangs den Nutzer durch den Sihlwald leitet, während dieser einen Kriminalfall mit Hilfe physikalischer Aspekte aufklärt. Somit befindet sich meine Arbeit in den Gebieten der Physik, der Informatik, der Kriminologie und auch der Natur. Die Ausgangslage meiner Arbeit ist die pure Natur, auf welche meine App aufbaut.

Zudem findet der Spaziergang in einer kleinen Gruppe statt. Denn auf der einen Seite ist der Mensch sozial, während er auf der anderen Seite individuell sein möchte. Dies führt dazu, dass beide Seiten im Umgang in einer kleinen Gruppe erfüllt werden. Somit entsteht eine gute Interaktion miteinander, welche die Arbeitsfähigkeit erhöht und auch das Lernen steigert.[1] Das Lösen meines Kriminalfalls in einer Kleingruppe ist für das Lernen optimal, denn alle können von dem Wir-Gefühl profitieren. Es gibt eine gemeinsame Herausforderung und damit ein Gefühl der Zusammengehörigkeit. Wichtig ist mir nicht, dass physikalisches Wissen erworben wird, sondern dass die Neugier und der Wille am Forschen geweckt wird und man Spass hat.

Oft wird Physik als etwas sehr Komplexes eingeschätzt und man stellt sich Physiker in weissen Kitteln im Labor vor. Jedoch hat die Physik auch eine andere Seite, da sie nicht nur in Laboren sondern überall vorkommt, besonders in der Natur. Um herauszufinden, wie das Leben auf der Erde entstehen konnte und wie sich das Sein und Werden verändert, muss man mit offenen Augen durch die Natur gehen.[2] Zudem ist es nie schlecht, bei einem kleinen Spaziergang ein bisschen Luft zu schnappen.

Das Verständnis der Physik bedeutet im Ganzen mehr als die Aneignung von Detailwissen. Es ist wichtig, Strukturen zu erkennen und aufzubauen.[3] Auch dient die Wissenschaft Physik dazu, Gesetze und Zusammenhänge zu erstellen, welche die Erscheinungen der Natur vorhersagen oder erklären.[4]

Um das Lernen zu erleichtern, möchte ich eine spielerische Umsetzung nutzen. Spielen fördert nicht nur das soziale Lernen und stärkt die Wahrnehmungsfähigkeit, sondern fördert auch die Ausdrucksweise und die Kreativität eines Menschen.[5] Spielerische Aktivität zieht Aufmerksamkeit auf sich und fordert handlungsorientiert zu einem Auseinandersetzen mit physikalischen Zusammenhängen auf. Spiel ist dabei eine freiwillige Handlung und ermöglicht dabei eine entspannte Wahrnehmung. Die Realität wird dabei direkt erlebt und Alltägliches kann unter einer physikalischen Perspektive aufgenommen werden.[6] Diese Fähigkeiten erleichtern das Lernen massiv. Auch kann man viel besser Lernen, wenn man Spass hat.

Mit meiner App und dem Bezug zu einer möglichen Geschichte möchte ich einen Einblick in die Physik bieten und Motivation aufbauen, um mit physikalischen Inhalten in Kontakt zu treten.



## 5. Grundlagen

Am Anfang meiner Arbeit werde ich zuerst ein paar Grundlagen, auf denen meine Arbeit aufgebaut ist, erarbeiten. Dies betrifft vor allem verschiedene Erscheinungen von Krimis und Outdooraktivitäten. Die Grundlagen der Programmierung werden zum besseren Verständnis erst im Kapitel 7.4 auf der Seite 45 erläutert.

### 5.1. Krimi-Aktivitäten

#### 5.1.1. Polizeimuseum Hamburg

In meinen Sommerferien war ich im Polizeimuseum Hamburg. Dort habe ich mich mit vielen verschiedenen Themen zur Mordaufklärung befasst. Zum Beispiel gab es einen Film über die Sicherstellung von Spuren am Tatort. Es wurde erklärt, wie man einen Fussabdruck von einer glatten Oberfläche aufnehmen kann und dann nachkonstruieren kann. Dies hat mich dann auf die Idee gebracht, auch einen Fussabdruck in meine Maturitätsarbeit einzubauen. Zudem habe ich im Polizeimuseum wahrgenommen, wie Schriftproben geprüft werden können. Die Texte für Schriftproben finden sich in meiner Arbeit wieder.



Abbildung 1: Obergeschoss des Polizeimuseums Hamburg mit Kriminaltechnik



Abbildung 2: Rosina Weiß vor dem Polizeimuseum Hamburg

---

## 5.1.2. Krimi-Spiele

Zum Zweck der Orientierung stelle ich bekannte Krimi-Spiele vor. Ausserdem habe ich recherchiert, ob es bereits Krimi-Apps gibt.

### Krimidinner

Ein Krimidinner ist ein Rollenspiel, bei welchem das Ziel ist, einen Kriminalfall zu lösen. Es gibt mehrere Arten von Krimispielen, zum Beispiel sitzt man mit 6 bis 10 Spielern zwei bis drei Stunden am Tisch und es gibt dabei ein Dinner. Eine andere Version des Spiels kann mit bis zu 20 Leuten in mehreren Räumen stattfinden, bei dem man sich frei bewegen kann. Etwas ist bei allen gleich: Jeder verkörpert eine eigene Rolle, die vor dem Termin, an dem das Spiel stattfindet, verteilt wird und auf die man sich vorbereitet. Am Tag des eigentlichen Dinners verkleidet man sich passend zur Rolle. Es gibt neben der Hauptstory vom Mord auch noch unzählige Nebenstorys, mit denen die Charaktere verknüpft sind. Jeder Charakter hat eigene Ziele, die möglichst erfüllt werden sollten, dabei werden andere Spieler gegen diese arbeiten. Vielleicht werden Affären und Intrigen aufgedeckt, es könnte ein zweiter Mord passieren oder auch nur eine Hochzeit.

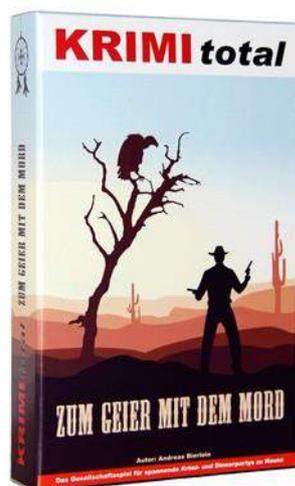


Abbildung 3: KRIMI total - Zum Geier mit dem Mord

### Escaperoom

Wenn man einen Escaperoom besucht, wird man für eine Zeitstunde in einen Raum eingeschlossen und muss dort drinnen gemeinsam Rätsel lösen, um wieder hinauszukommen. Meistens ist der Escaperoom auch mit einem bestimmten Thema verknüpft. Man muss vielleicht aus einem Safe ausbrechen oder den Weg aus einem Labyrinth finden. Es werden aber auch Mordfälle gelöst und nur, wenn man den richtigen Täter identifiziert hat, kommt man aus dem Raum. Diese Spiele werden oft in einer Gruppe von drei bis sechs Spielern durchgeführt. Die Kommunikation und das Teamwork sind dabei entscheidend. In der Schweiz existiert eine Website, auf der zentral die lokalen Escaperooms aufgelistet sind.[7]

### Exitgame

Ein Exitgame ist eine Art Escaperoom, den man aber bequem Zuhause spielen kann. Es ist ein Gesellschaftsspiel, das einen Escaperoom simuliert. Der Ablauf ist sehr ähnlich:

Man soll alle Rätsel lösen. Diese Exitgames werden am besten auch in der Gruppe gespielt. Meistens kann man sie nur einmal spielen, weil man dann die Rätsel bereits kennt und auch das Material während des Spiels zerschnitten und gefaltet wird. Es gibt auf dem Markt schon sehr viele verschiedene Versionen von Exitgames von verschiedenen Herstellern.

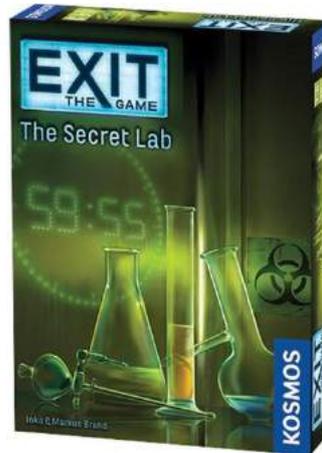


Abbildung 4: Exitgame - The Secret Lab von Kosmos

### Krimi-Rallye

Eine Krimi-Rallye findet meistens in einer Stadt statt und ist besonders für Touristen geeignet. Man macht dabei einen durch Informationen geführten Spaziergang durch die Stadt und muss einen Kriminalfall lösen. Ein entscheidendes Merkmal ist, dass man dabei mit Gegenständen und Orten in der Stadt interagiert und diese benötigt, um den Fall zu lösen.

Auf der Website von [krimispass.ch](http://krimispass.ch) kann man sich für ein Krimi-Spiel anmelden und vor Ort mit dem eigenen Smartphone nach Hinweisen suchen. Es gibt zwei Fälle, die an sieben Orten gespielt werden können. Man erhält eine elektronische Landkarte, die Spielanleitung und verschiedenen Informationen (Fotos, Texte, Audiodateien u. a.) per Mail oder QR-Codes von Schildern auf der Route und klärt den beschriebenen Fall auf.[8]



Abbildung 5: Screenshot der Website [krimispass.ch](http://krimispass.ch)

---

## 5.2. Outdoor Aktivität

Wichtig war mir, dass meine Nutzer mein Krimispiel nicht Zuhause auf dem Sofa oder am gemeinsamen Esstisch miteinander spielen, sondern dass sie sich dabei bewegen und an der frischen Luft in der Natur sind. Viele junge Menschen verbringen einen Grossteil des Tages im Sitzen und in geschlossenen Räumen und sind in ihrer Freizeit wenig unterwegs. Ausserdem wird zuhause viel im Internet gesurft und gegamet, sodass die Wahrnehmung und Entdeckung der Welt und das Sammeln von Lebenserfahrung nur virtuell stattfindet. Laut einer Befragung wünschen sich drei von vier Kindern mehr Spielen im Freien, setzen dies aber nicht um. Nur 50% der befragten Kinder schaffen es, sich mehr als einmal pro Woche mit Freunden überhaupt zu treffen.[9] Mit Hilfe meiner Krimi-App schaffe ich einen Anreiz, dass meine Nutzer einen Spaziergang in der Natur unternehmen und dabei sich aktiv bewegen. Ausserdem spielen sie gemeinsam, was die Teamfähigkeit und Kooperationskompetenzen stärkt.

In einer süddeutschen Studie, in der 130 Grundschulen beteiligt waren, konnte festgestellt werden, dass das Erleben der Natur extrem wichtig für die Entwicklung der Kinder und Jugendlichen ist. Zum Beispiel konnten nur noch 50% der Jugendlichen von einem eindrucksvollen Naturerlebnis berichten und 26% der Schüler fand keine Antwort auf die Frage, welche Aufgabe und Funktion der Wald hat. Die Studie hat ebenfalls herausgefunden, dass Kinder, die mit Erlebnis- und Naturpädagogik in Berührung kamen, deutlich besser abgeschnitten haben. Es wurde festgestellt, dass Menschen durchschnittlich 95% des Lebens in Kunstbiotopen verbringen, das heisst nicht in der Natur, sondern in Räumen.[10] Diese Entwicklungen erschrecken mich. Deswegen möchte ich einen Beitrag zur Änderung leisten und die Nutzer meiner Krimi-App zu einer walddpädagogischen Aktivität veranlassen und damit eine emotional-sensibilisierende Wirkung initiieren.

### 5.2.1. Geocaching

Geocaches sind Verstecke auf der ganzen Welt, die mittels Navigation innerhalb einer App geortet werden können. Für die Teilnahme ist erforderlich sich anzumelden. Anschliessend können auf einer digitalen Landkarte Verstecke lokalisiert und mit Hilfe der App aufgefunden werden. Die Nutzer navigieren in die unmittelbare Umgebung des Versteckes und suchen nach einem mehr oder weniger grossen Gefäss. Wenn man fündig geworden ist, trägt man sich analog mit seinem Anmelde-Namen in ein kleines Logbuch ein und kann soweit vorhanden, Gegenstände aus dem versteckten Behälter entnehmen und selber welche hineinlegen. Digital loggt man den Fundort ebenfalls. Manchmal findet man sogar einen Travel-Bug. Dies ist ein Trackable, der von Versteck zu Versteck reist und so Geschichten sammelt. Wenn man Travel-Bugs kauft und platziert, kann man selber welche auf Reisen schicken und innerhalb der App oder im Internet auf der Website der Groundspeak Inc. verfolgen, wo der Travel-Bug sich gerade auf der Welt befindet. An Treffen von Geocaching-Fans kann man sich untereinander austauschen und Fundstücke sowie Travel-Bugs vorzeigen, einloggen und bewundern.

Seit meiner Kindheit haben meine Schwestern und ich auf Wanderungen solche Geocaches gesucht und gefunden. Für mich war diese Aktion jeweils spannend und

aufregend und ich möchte diese Erlebnisse in meine Krimi-App integrieren. Ich werde Verstecke im Sihlwald kreieren.



Abbildung 6: Screenshot von der Website geocaching.com

### 5.2.2. Foxtrail

In Zürich gibt es Stadt-Rallyes, welche sich Foxtrail nennen. Es werden Stadtspaziergänge angeboten, wobei man die Spur eines Fuchses verfolgt. Die Stadt verwandelt sich zur Spielwiese: Man knackt Codes, findet Verstecke und agiert mit unterschiedlich angebrachten Installationen, zum Beispiel in Schaufenstern von Detaillisten oder in Telefonzellen und Eingangshallen. Öffentliche Gebäude sind meist einbezogen. Man wird aufgefordert, bestimmte Gebäude oder Sehenswürdigkeiten zu finden, zum Beispiel eine Statue in einem Park oder eine Texttafel zu einem historischen Gebäude. Man lernt dabei viele sehenswerte Orte kennen.

Zusammen mit meiner Familie haben wir während meiner Maturitätsarbeit einen solchen Foxtrail durchgeführt. Das Auffinden der Orte und die Orientierung war verbesserungswürdig. Das möchte ich unbedingt in meiner Krimi-App anders regeln.



Abbildung 7: Logo Foxtrail



## 6. Das Endprodukt

In diesem Kapitel stelle ich die Grundzüge meines Endproduktes vor. Dies erleichtert später das Verständnis der Entwicklungsphase und des dahinterstehenden Konzepts.

Meine App ist für einen gemeinsamen Spaziergang durch den Sihlwald gedacht. Es können Personen jeder Altersstufe mitmachen, jedoch empfehle ich ein Alter ab 12 Jahren. Zudem ist eine Mindestanzahl von zwei Personen empfehlenswert. Der ganze Spaziergang dauert etwa zwei bis drei Stunden, je nach Schritttempo der Nutzer.

Ziel dieses Spaziergangs ist die Aufklärung eines Kriminalfalles. Die Teilnehmer sind erfolgreich, wenn sie den Mörder ausfindig gemacht haben. Das Rätsel kann gelöst werden, indem eine bestimmte Anzahl an Stationen nacheinander absolviert werden.

### 6.1. Konzept der App

Zu Beginn des Programmierens musste ich mir überlegen, wie ich meine App aufbauen möchte. Zur Entscheidungsfindung habe ich auf den Aufbau und die Struktur einiger Apps geachtet, die ich selber benutze. Schliesslich habe ich mich dazu entschieden, meine App in einzelne physikalische Themen zu unterteilen und für jedes Thema einen Posten zu kreieren.

#### 6.1.1. Übersichtsdisplay

Ich habe mir überlegt, dass es übersichtlich wäre, wenn es eine Art Auflistung aller Stationen auf dem Handydisplay gibt, über die man sich gut orientieren kann. Diese Auflistung sollte auch meine Startseite werden. So bekommt man gleich am Anfang einen kleinen Überblick. Es gibt die Möglichkeit Unterscreens aufzurufen und zum Überblick wieder zurückzugelangen. Zudem sollte es nicht nur möglich sein, über diese Liste die einzelnen Posten aufzurufen und sie nacheinander abzuarbeiten, sondern man sollte auch zu alten, schon fertig gelösten Posten zurückgehen können.

So habe ich einen ScrollView mit allen Posten als Startseite der App erstellt. Ein ScrollView erlaubt dem Nutzer den Bildschirminhalt zu vergrössern und hinauf- und hinunterzuwischen. Wenn man nun auf den ersten Posten klickt, kommt man in die einzelnen Unterscreens des Postens. Durch Pfeile am Bildschirmrand kann man zwischen den Seiten blättern und die Aufgabenstellung des Postens dann lösen. Wenn man am Ende der Station angekommen ist, gelangt man wieder automatisch auf den Übersichtsscreen und kann dort mit dem nächsten Posten darunter beginnen.

Ausserdem gibt es in meinem Übersichtsscreen auch noch einen Informationsbutton. Wenn man auf diesen drückt, gelangt man auf die Infoseite der App, welche die Inhalte und den Zweck beschreibt. Auch ist hier der Kontakt zu mir möglich, falls irgendetwas nicht funktionieren sollte oder Hilfe gebraucht wird. Dafür habe ich die Emailadresse [dasnotizbuch@gmx.ch](mailto:dasnotizbuch@gmx.ch) kreiert, über die die Nutzer mich kontaktieren können.

Damit man die Posten in der von mir vorgesehenen Reihenfolge gelöst werden, habe ich eine Art Sperre innerhalb dieser Übersichtsauflistung eingerichtet. So sind die

---

einzelnen Posten vor Sprüngen gesperrt und können nur nacheinander abgearbeitet werden. Das gleiche System habe ich auch bei den Rätseln gewählt. Hier kommt man nicht weiter, wenn man nicht die richtige Lösung eines vorherigen Rätsels gefunden hat. Damit man jedoch die Möglichkeit hat, das schon erfolgreich Gespielte nochmals anzusehen, wird der Fortschritt der Benutzung in der App gespeichert. Das heisst, wenn man eine Aufgabe einmal erfolgreich freigeschaltet hat, ist sie für immer freigeschaltet. Um diese Freischaltung zu löschen, müsste man die App vom Gerät löschen und nochmals neu installieren.

### 6.1.2. Navigation

Während der Nutzung meiner App bewegen sich die Spieler im Sihlwald, so sollte man Informationen erhalten, wohin man laufen soll und an welchen Orten gewisse Dinge auffindbar sind. Für diese Funktionalität habe ich eine Navigation in meine App integriert. Diese berechnet dem Nutzer, wie weit und in welche Richtung er bis zum nächsten, geplanten Standort gehen muss. Immer dort, wo ein Standortwechsel stattfindet, kann über einen Knopf auf diese Funktion zugegriffen werden. Diese Funktion hat gegenüber einer Landkarte, worin die Standorte einfach eingezeichnet sind, den Vorteil, dass man nicht von Anfang an genau weiss, wo genau man hingehen muss. Beim Testen der App, musste ich jedoch feststellen, dass diese Funktion leider eher ungenau war und habe deshalb doch noch eine grobe, zusätzliche Karte eingebaut. Diese wird am Anfang der App gezeigt, damit die Nutzer wissen, welcher Weg durch den Sihlwald im Verlauf der App geplant ist. Beim Spielen kann man diese Karte immer wieder aufrufen, wenn man ins Menü zurückgeht. Diese Karte ist als Notfall-Option vorgesehen, wenn man nicht mehr weiss, wo man langgehen soll. Sonst bleibt der Aspekt der Orientierung nicht so spannend. Hier ist anzufügen, dass man am Schluss wieder dort ankommt, wo man gestartet ist.

### 6.1.3. Notizbuchfunktion

Eine letzte und grosse Funktion in meiner App ist die Notizbuchfunktion. Diese Funktion hat auch wesentlich zum Namen meiner App beigetragen. Sie legt wie in einem herkömmlichen Notizbuch Informationen ab.

Diese Funktion wird zugänglich, sobald man das Notizbuch der Leiche, visualisiert durch ein Foto, gefunden hat. Dies erfolgt sogleich beim ersten Posten, dem Leichenfundort.

Es wird mittels dieser Funktion gewährleistet, dass alle Informationen, die man im Laufe der Zeit zum Kriminalfall sammelt, automatisch in diesem virtuellen Notizbuch abgelegt werden. Zudem können die Nutzer jederzeit (durch einen Button) auf das virtuelle Notizbuch und die darin enthaltenen Informationen zugreifen. Hierfür werden innerhalb der App alle Seiten regelmässig erneuert, wenn Informationen beim erfolgreichen Lösen der Rätsel hinzukommen. Mittels einer Speicherfunktion werden beim Hinzukommen von neuen Informationen die alten, bereits erhaltenen Informationen nicht überschrieben und gelöscht, sondern sind weiterhin verfügbar.

### 6.1.4. Die Stationen

Es gibt insgesamt neun Stationen bzw. Posten, die für den Kriminalfall gelöst werden müssen. Die einzelnen Postendisplays sind sehr unterschiedlich aufgebaut, denn sie sind abhängig von der Aufgabenstellung. Manchmal sind Bilder erforderlich, manchmal muss Text gelesen werden oder es gibt ein Multiple Choice-Quiz. Die Aufteilung habe ich möglichst abwechslungsreich gestaltet; sie hängt sehr von den einzelnen Posten, deren Inhalt und der konkreten Aufgabenstellung ab.

## 6.2. Kriminalfall

In meiner App wird ein Kriminalfall aufgeklärt. Im Verlaufe der Posten erhält man hierfür immer mehr Informationen und schlussendlich ist es das Ziel, den Mörder zu ermitteln und zu „verhaften“. Die gesamte Mordstory habe ich mir selbst ausgedacht. Darum möchte ich diese jetzt kurz zusammenfassen und auf die einzelnen Protagonisten genauer eingehen. Hierbei möchte ich anmerken, dass in den folgenden Abschnitten die ganze Geschichte erzählt wird, das heisst der Leser erfährt, wer der Mörder ist. **Deswegen empfehle ich Personen, die die App gerne selbst ausprobieren möchten, dieses Kapitel zu überspringen oder erst zu lesen, wenn sie die App bereits gespielt haben.** Andernfalls wäre es bedauerlich, wenn man das Ende schon kennt.

### 6.2.1. Geschichte

Meine App startet mit dem Fund einer Leiche, welche ans Ufer der Sihl angeschwemmt wurde. Diesen Todesfall gilt es aufzuklären. Die Nutzer schlüpfen in die Rolle der neu hinzugekommenen Mitarbeiter einer Polizeistation und sind noch in der Ausbildung. Sie dürfen zum ersten Mal mit zu einem Tatort gehen, doch keiner der anderen Mitarbeiter traut ihnen etwas zu.

Das Opfer ist ein Physiker namens Julian Gutenberg. Er wurde erschossen, ist daraufhin in die Sihl gefallen und wurde von der Strömung mitgerissen. In der unmittelbaren Umgebung des Fundorts der Leiche wurde ein Notizbuch gefunden. Dieses gehörte ihm, jedoch wurden gewisse Seiten herausgerissen, die man im Verlauf der Krimi-App finden wird.

Julian ist verheiratet mit Emma Gutenberg. Die beiden hatten vor zwei Jahren einen tragischen Autounfall, woraufhin Emma ins Koma fiel und erst vor einem halben Jahr aufgewacht ist. Mit Hilfe einer ersten gefundenen, herausgerissenen Seite aus dem Notizbuch findet man heraus, dass unser Opfer schon mehrmals verfolgt wurde. Ausserdem wurden mehrere Anschläge auf ihn verübt.

Er wusste etwas, was er nicht wissen durfte und zu seinem Unglück verfügte er sogar über Beweise. Er notierte in seinem Notizbuch, dass er diese Beweise versteckt hatte. Den Fundort verschlüsselte er, sodass man diesen nur durch einen aussergewöhnlichen Papierflieger finden könne. Auch erwähnte er in diesem Zusammenhang den Namen eines unbekanntes Unternehmens (Scheinfirma Techgeniosus einer Terroristenorganisation). Dadurch entsteht ein erster Verdacht auf ein mögliches

---

Mordmotiv. Der oder die Mörder haben Julian Gutenberg umgebracht, damit ein wichtiges Geheimnis nicht ans Licht kommt.

Um weitere Informationen zu bekommen und ständig auf dem Laufenden gehalten zu werden, müssen die Nutzer ihr Können zuerst den ermittelnden Polizisten zeigen. Wenn sie bewiesen haben, dass sie doch noch nützlich sein können, wird den Nutzern der Zugang zu allen bisher bekannten Informationen gestattet.

So werden auch die verdächtigen Personen aufgefächert. Der Verdacht richtet sich gegen Emma Gutenberg (Ehefrau), Oliver Gutenberg (jüngerer Bruder), Amelie Herzog (Mitarbeiterin bei Omnotitia), Hendrik Quedens (Chef von Omnotitia) und Techgeniosus (terroristische Scheinfirma). Ein Selbstmord kann ebenfalls nicht ausgeschlossen werden.

Daraufhin wird ein Beweisdokument (siehe Kapitel 13.3.2, S.209) gefunden, welches aufweist, dass ein Quantencomputer, der im Unternehmen, bei dem die Gutenbergs beschäftigt sind, entwickelt wird, illegal an ein Terrorunternehmen namens Techgeniosus verkauft wurde. Man weiss aber nicht von wem. Es besteht die Möglichkeit, dass der Quantencomputer von Julian Gutenberg stammt. Nun stellt sich die Frage, wer ihn verkaufen wollte oder schon verkauft hat.

Durch das Feststellen des Tatzeitpunkts können Alibis aufgenommen werden. In einem Verhör mit der Ehefrau Emma Gutenberg erfährt man weitere Informationen im Zusammenhang mit dem Kriminalfall. Schliesslich wird wieder ein Notizbuchzettel gefunden und entschlüsselt. So erfährt man, dass die Terrororganisation Techgeniosus vom kleinen Bruder Oliver Gutenberg geleitet wird. Zudem findet man auch heraus, dass der Chef des Unternehmens Omnotitia, bei dem die Gutenbergs beschäftigt sind, mitbekommen hat, dass der Quantencomputer von Julian und Amelie funktioniert. Hendrik Quedens möchte die Lorbeeren und das Geld dafür ernten, wenn diese neue Entwicklung veröffentlicht wird. Durch die verschlüsselten Seiten (siehe Abbildung 8 bis Abbildung 13) findet man zum Beispiel heraus, dass Julian und Amelie eine Affäre hatten, als Emma im Koma lag.

Somit werden im Verlaufe des Krimispiels die verschiedenen Motive der Verdächtigen bekannt.

Wenn schliesslich im Wald ein Fussabdruck gefunden wird, kann die Schuhgrösse und das Gewicht einem potentiellen Mörder von Julian Gutenberg zugeordnet werden. Da der Abdruck von einer leichteren Person mit kleinen Füessen stammt, reduziert sich der Verdacht eher auf die in Frage kommenden Frauen. Somit wird der Kreis der verdächtigen Personen auf die Arbeitskollegin und die Ehefrau eingegrenzt. Jedoch weiss man nicht hundertprozentig, ob der Fussabdruck wirklich vom Mörder stammt oder nicht.



---

In der kriminaltechnischen Untersuchung am Wohnort der Gutenbergs wird im Haus derselbe Fussabdruck wie im Wald gefunden. Als bekannt wird, dass das Opfer aus einiger Entfernung erschossen wurde, kann der Fall von Selbstmord (aufgrund von Schuldgefühlen gegenüber seiner Frau) ausgeschlossen werden. Auch findet man am Standpunkt des Täters Partikel von der gleichen Erde wie die vom Fussabdruck im Wald. Somit kann eine Verbindung zwischen dem Mörder und Emma hergestellt werden und damit ist es klar, dass der Mörder die Ehefrau war.

Am Ende der Krimi-App gibt es neben der Frage, wer verhaftet werden soll, auch noch die Frage, wo sich der Quantencomputer jetzt befindet. Diese Frage kann man nur aufklären, wenn man wirklich ganz genau aufgepasst hat. Am Anfang findet man beim Opfer ein Foto seiner Ehefrau (siehe Abbildung 14). Auf die Rückseite des Fotos hatte sie eine Widmung geschrieben (siehe Abbildung 15).



Abbildung 14: Foto von Emma Gutenberg

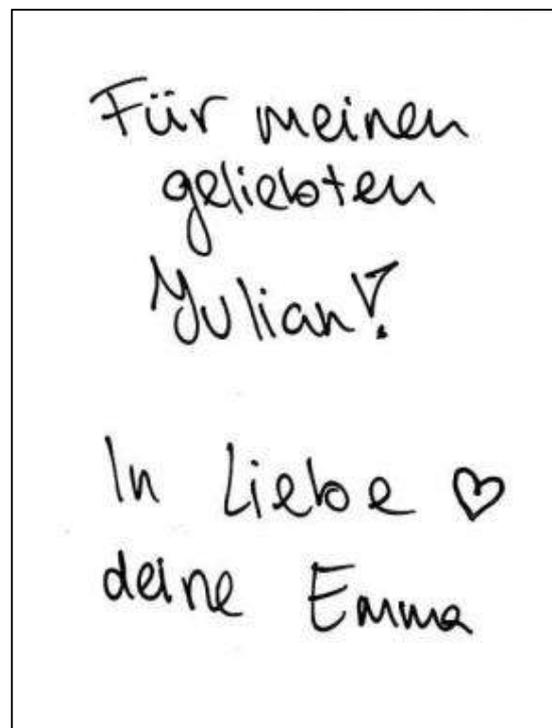


Abbildung 15: Widmung auf der Rückseite des Fotos

Schliesslich kann man feststellen, dass die Schrift der Einträge des Notizbuches, der Schriftprobe von Emma und die der Widmung genau dieselbe ist. Dies weist darauf hin, dass alle Informationen im gefundenen Notizbuch von der Ehefrau geschrieben worden sind und somit nicht der Wahrheit entsprechen. Die Ehefrau hatte alles ganz genau geplant. Auch das Verkaufsdokument ist gefälscht worden, was man auch durch einen Unterschriftenvergleich herausfinden kann. Sie wollte mit der Fälschung des Kaufverkaufes von dem Quantencomputer an die Terrororganisation den Verdacht auf die Arbeitskollegin lenken. Diese hatte eine Affäre mit ihrem Ehemann, während sie selbst hilflos im Koma lag. Mit der Tat wollte sie die beiden für diese Untreue bestrafen. Ausserdem war sie wütend darüber, dass die beiden den von ihr entwickelten

Quantencomputer übernommen und ohne ihr Einverständnis und Wissen weiterentwickelt hatten. Somit hat sie Rache ausgeübt, indem ihr Mann für seine Taten sterben musste und die Arbeitskollegin Amelie im Gefängnis landen sollte. Aufgrund dieser Informationen kann man darauf schliessen, dass der Quantencomputer immer noch sicher im Unternehmen steht. Man weiss jedoch nicht, ob er schon funktioniert oder nicht.

### 6.2.2. Charaktere

In meiner Mordstory habe ich neben der Person, die ermordet wird, mehrere Verdächtige erfunden. Über die Notizbuchfunktion erhält man zu jedem Verdächtigen eine Personalakte, in der alle wichtigen Informationen gespeichert werden.

Um die verschiedenen Verdächtigen nicht nur alleine über einen Namen im Spiel leben zu lassen, habe ich jedem von ihnen ein Gesicht gegeben. Deswegen habe ich zu jeder Person verschiedene Fotos und Schriftproben angefertigt, welche bei der Auflösung vom Krimi eine wichtige Rolle spielen. Die fünf Verdächtigen sind die Ehefrau Emma Gutenberg (siehe Abbildung 16), der jüngere Bruder Oliver Gutenberg (siehe Abbildung 17), die Arbeitskollegin Amelie Herzog (siehe Abbildung 18), Hendrik Quedens, der Chef der Firma Omnotitia, welche Arbeitgeber von Julian, Emma und Amelie ist (siehe Abbildung 19), und das Unternehmen Techgeniosus, welches eine Scheinfirma eines Terrorunternehmens ist. Die Firma Techgeniosus wird repräsentiert von dem von mir erstellten Logo mit dem Firmennamen (siehe Abbildung 20). Wenn man noch etwas mehr zu den einzelnen Charakteren erfahren möchte, befinden sich im Anhang die Personalakten zu den einzelnen Personen. Hierbei befindet sich nur die Endversion der Akten im Anhang. Diese werden im Verlaufe der Krimi-App aufgebaut und mit Hinweisen gefüllt (siehe Kapitel 13.3.1, S.203).



Abbildung 16: Emma Gutenberg (Ehefrau)



Abbildung 17: Oliver Gutenberg (Bruder)



Abbildung 18: Emma Herzog (Arbeitskollegin)



Abbildung 19: Hendrik Quedens (Chef)



Abbildung 20: Firmenlogo Techgeniosus

### 6.3. Beschreibung der einzelnen Posten

In den folgenden Unterkapiteln gehe ich auf die einzelnen Posten in meiner App ein und erläutere die physikalischen Aufgaben in meiner Arbeit.

#### 6.3.1. Posten 1: Einführung

Im ersten Posten meiner App erkläre ich den Nutzern den Aufbau meiner App und wie sie funktioniert, damit sie sich zurechtfinden. Hier erkläre ich die Funktion des Notizbuches, welche alles abspeichert, was man im Laufe meines Krimispiels gesammelt hat, und zeige auf, wie man mit Hilfe des Navigationspfeiles an den richtigen Standpunkt kommt. Die Story wird mit Hilfe des Polizisten Henry eingeführt. Dieser leitet die Untersuchung im Fall Gutenberg und ist auf die Hilfe der Nutzer angewiesen, da er kein grosses physikalisches Wissen hat und somit den Fall nicht ohne Hilfe lösen kann.

Schliesslich leitet die App die Nutzer an die Sihl an den Fundort der Leiche. Dort werden die Tatortbilder gezeigt (siehe Abbildung 21 bis Abbildung 28). Diese Tatortfotos sind

sehr wichtig, da sie die ganze Story visuell unterstützen, was das Einfühlungsvermögen der Nutzer steigert. Hierbei wird die Immersion erhöht.

Das Abtauchen von Menschen in eine virtuelle Realität wird als Immersion bezeichnet. Die Umstände in dieser künstlichen Umgebung berühren emotional ähnlich tief wie die Realität im wirklichen Leben. Eine neue Selbstwirksamkeit kann erlebt werden, man weiss zwar, dass das Geschehen eine Illusion darstellt, doch der Körper reagiert ähnlich wie im echten Leben.[11]

Sobald meine Spieler den Kriminalfall anfangen zu lesen, werden sie in diese Geschichte hineingezogen. Sie schlüpfen in die Rolle von Ermittlern und werden motiviert, den Kriminalfall aufzuklären. Anders wäre es zum Beispiel, wenn man lediglich mitteilt, dass die Spieler zur Station A laufen und Aufgabe X lösen sollen, dann zur Station B und Aufgabe Y etc. Es gäbe dann ein Oberthema „Physik“, jedoch wäre das Erlebnis beim Spielen der App nicht dasselbe. Ich wollte einen übergeordneten Anreiz erschaffen, damit meine Nutzer mittels der Krimi-App mitfiebern und eifrig agieren.

Man bekommt hier auch einige Informationen zum Opfer Julian Gutenberg und wird dann zum nächsten Posten geleitet. Bevor man anfangen kann, wird bei diesem Posten noch eine Kontrolle eingeführt, damit man die App nicht einfach auf dem Sofa von Zuhause lösen kann, sondern effektiv im Wald anwesend sein muss. Dazu habe ich die Zahl 28 in die Holzbrücke eingeritzt, die man finden und in die App eingeben muss. Über die Brücke müssen die Spieler zu Beginn gehen. Ausserdem wird die Zahl später noch einmal gebraucht.



Abbildung 21: Die Leiche



Abbildung 23: Der Kopf



Abbildung 22: Der Oberkörper



Abbildung 24: Die Schusswunde



Abbildung 25: Der linke Fuss mit Schuh



Abbildung 27: Das Notizbuch



Abbildung 26: Der rechte Fuss ohne Schuh



Abbildung 28: Der Leichenfundort

### 6.3.2. Posten 2: Optik

Im zweiten Posten meiner Arbeit geht es um das Thema Optik. Man findet an der Sihl eine Kiste mit drei verschiedenen Lupen. Die Polizei verfügt über einen kleinen Zettel mit winziger Schrift (siehe Abbildung 29), welche man mit blossem Auge nicht lesen kann, und ein Foto der Ehefrau mit einer Widmung auf der Rückseite (siehe Abbildung 14 und Abbildung 15, S.22), das in der Hosentasche des Opfers gefunden wurde.

Die Nutzer probieren die verschiedenen Lupen (Brennweiten 30mm, 100mm und 200mm) aus und finden heraus, mit welcher man am besten die winzige Schrift auf dem Papier lesen kann. Hierbei muss man die verschiedenen Eigenschaften wie Durchmesser, Dicke oder Brennweite der Lupe genauer untersuchen und schliesslich auch eine Begründung finden, wieso das Lesen mit der einen Lupe besser als mit den anderen funktioniert. Lupen gibt es in unterschiedlichen Brennweiten, welche den Abstand zum Papier bei einer perfekten Vergrößerung bestimmen. Ab einem bestimmten Punkt dreht sich das Bild, welches man sieht, um.[12]

Wenn man dies herausgefunden hat, kann man den Zettel entziffern und findet eine Anleitung für ein spezielles Schloss, welches ganz am Schluss geöffnet werden muss, um an die Auflösung des Krimirätsels heranzukommen.

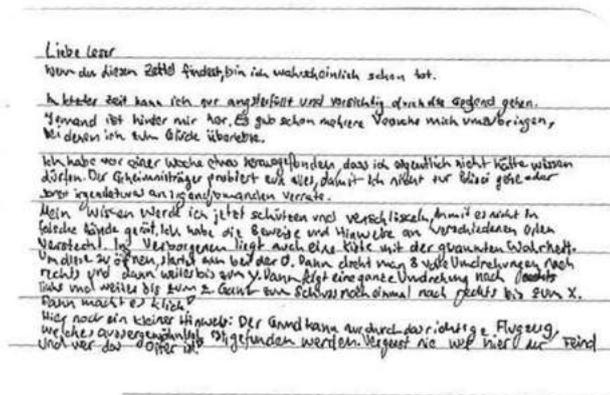


Abbildung 29: Zettel mit winziger Schrift

### 6.3.3. Posten 3: Akustik

Der dritte Posten dient dazu, den Nutzern die Akustik und ihre Eigenschaften etwas näher zu bringen. Da bei diesem Posten viel Teamarbeit benötigt wird, gibt es zuerst eine kleine, vorbereitende Aufgabe, welche diese stärkt. Man geht zu einem besonderen Walderlebnispfadposten. Dieser hat das Gleichgewicht als Thema.

Man nimmt von dort einen der Stöcke, welche auf dem Boden liegen. Nun probiert man in der Gruppe Folgendes aus: Der Stock wird von allen Personen so gehalten, dass pro Person zwei einzelne, vorgestreckte Finger ihn tragen. Jetzt soll der Stock von allen gemeinsam auf den Boden gelegt werden (siehe Abbildung 30). Da alle immer probieren, Kontakt mit dem Stock zu halten, ist dies eine relativ schwere Aufgabe und kann nur klappen, wenn wirklich alle miteinander zusammenarbeiten. Je grösser die Gruppe ist, desto schwerer ist die Aufgabe.

Die Gruppe kann es nur schaffen, wenn sie es gemeinsam bewerkstelligt und sich als etwas Neues begreift und nicht als Summe von Individuen.[13] Wenn man dies geschafft hat, kommt die eigentliche Aktion des Postens.



Abbildung 30: Teamworkaufgabe mit einem Stock

---

Hierbei entfernt sich eine Person weit von den anderen (ca. 80m) und klatscht stark in die Hände über den Kopf. Hier kann man dann feststellen, dass eine winzige Verzögerung zwischen dem Bild und dem Geräusch entsteht. Das heisst, dass man zuerst sieht, wie die entfernte Person in die Hände klatscht und dann hört man erst das Geräusch des Klatschens, da die Lichtgeschwindigkeit schneller als die Schallgeschwindigkeit ist.[14] Zirka 330 Meter pro Sekunde benötigt Schall in der Luft zur Ausbreitung. Demnach braucht man einigen Abstand, um diese Eigenschaft messbar werden zu lassen.[15] Dass das Wetter, insbesondere die Windrichtung Messungen der Zeit ungenau werden lässt, kann durchaus vernachlässigt werden. Auch die Trockenheit und Temperatur der Luft kann Einfluss auf die Schallwellen nehmen (effektiver Reflexionskoeffizient der Temperaturinversion).[16] Es geht bei diesem Posten jedoch nicht um die Genauigkeit der Messung, sondern um die Beobachtung und das Verständnis des Prinzips insgesamt. Man stellt fest, dass Licht sich schneller als Schall verbreitet, da Schall vom Luftwiderstand gebremst wird, sowie dass nur Schall hörbar ist und dass er materialabhängig ist.

Schliesslich geht man zum Waldxylophon, welches zum Erlebnispfad des Sihlwaldes gehört, und untersucht hier, inwiefern die Tonhöhe von der Länge und Grösse eines Gegenstandes, der in Schwingung versetzt wird, abhängt. Am Schluss bekommt man eine Auswahl von möglichen Eigenschaften des Schalls und man muss die Richtigen herausfiltern und anwählen. Wenn man alles absolviert hat, kann man durch diese Leistung in diesem Posten seine Fähigkeiten und Kompetenzen vor der Polizei beweisen. Diese traut nun den Nutzern und weicht diese auch in neue Informationen ein. Somit bekommt man Zugang zu allen Personalakten der Verdächtigen.

#### 6.3.4. Posten 4: Aerodynamik

Als vierten Posten habe ich mir etwas überlegt, bei dem man etwas Kreatives bastelt. Dieses Ziel habe ich mit dem Thema Aerodynamik verbunden, indem ich die Nutzer verschiedene Papierflugzeuge basteln lasse.[17] Jeder Papierflieger hat seine eigenen Besonderheiten. Der eine hat zum Beispiel eine flache Nase im Gegensatz zu einem anderen, der vorne spitz zuläuft. Wenn man die Papierflieger gefaltet hat, werden sie mehrmals hintereinander ausprobiert und fliegen gelassen. Es wird nach Besonderheiten gesucht. Es wird zum Beispiel die Frage gestellt, welcher der Flieger am weitesten fliegt oder ob einer im Vergleich zu den anderen besonders lange in der Luft bleibt. Am Schluss stellt man fest, dass es ein Papierflugzeug gibt, welches nicht sehr weit, jedoch immer in Kreisen fliegt und manchmal sogar einen Looping macht.

Julian Gutenberg hat an seinem Arbeitsplatz im Unternehmen selber sehr viele verschiedene Papierflieger gelagert, welche alle aus einem Dokument gefaltet wurden, welches von Papierflieger zu Papierflieger minimal anders ist. Da man nun den „speziellen“ Papierflieger gefunden hat, weiss man, welches das richtige und nicht gefälschte Dokument ist, welches den Verkauf eines Quantencomputers an eine Terroristenorganisation aufweist (siehe Kapitel 13.3.2, S.209). Zu diesem Posten findet man im Anhang die Falanleitung (siehe Kapitel 13.3.3, S.214).

### 6.3.5. Posten 5: Astronomie

An dem Posten Astronomie geht es darum, mit Hilfe des Sonnenstands zum Zeitpunkt des Schusses, mit dem das Opfer getötet wurde, den genauen Tatzeitpunkt herauszufinden. Dadurch bekommt man die abgebildete Grafik (siehe Abbildung 31) und die danebenstehende Tabelle (siehe Abbildung 32). Durch die Abschätzung des Einschusswinkels kann man in der angegebenen Tabelle die entsprechende Uhrzeit der Tat ablesen.

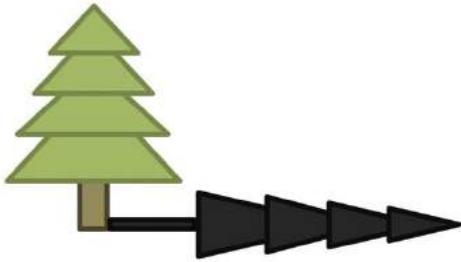


Abbildung 31: Grafik zur Tatzeitpunktbestimmung

**Tabelle zur Tatzeit**

Winkel	Uhrzeit
80°	14:00
60°	15:00
45°	16:00
30°	17:00
15°	18:00

Abbildung 32: Tabelle zur Tatzeitpunktbestimmung

Als Zusatzfrage ist hier zu beantworten, wo in der Grafik (siehe Abbildung 31) der Süden eingeordnet werden kann. Dies kann man durch reine Logik herausfinden. Es soll jedoch auch zeigen, dass nicht alle Seiten der Physik sehr kompliziert sind, sondern dass man durch bekannte physikalische Phänomene einfache Zusammenhänge herstellen kann.

### 6.3.6. Posten 6: Verschlüsselung

Der sechste Posten ist ein eher theoretischer Posten. Hier geht es darum, einen Teil der Hintergrundgeschichte besser zu verstehen. Denn hier behandelt man das Thema Quantencomputer. Dieses Thema kommt auch in der Mordstory vor und ich erkläre den Nutzern in diesem Posten die Grundidee von einem Quantencomputer und wie er in etwa funktioniert. Dies soll dazu dienen, dass man die Story besser verstehen und nachvollziehen kann und man auch eine Vorstellung bekommt, worum es bei dieser Erfindung des Quantencomputers geht. Hierbei beziehe ich mich vor allem auf das Wissen aus dem Interview mit Heike Riel. Um die Leistung eines Quantencomputers darzustellen, habe ich mir überlegt, dass man die Primfaktorzerlegung integrieren kann.[18] Ein Quantencomputer kann auch dazu eingesetzt werden, um Verschlüsselungen zu knacken, welche auch mit Primfaktorzerlegung funktionieren. Somit bekommt der Nutzer mehrere Zahlen, die er in Primzahlen zerlegen muss. Da die Zahlen stark ansteigend werden, wird es immer schwerer, die Primzahlen zu finden, und schliesslich erkennt man, dass man exponentiell länger braucht von Primzahl zu Primzahl. Das Gehirn eines Menschen soll hierbei einen normalen Computer symbolisieren und als Vergleich dafür stehen, dass ein Quantencomputer dies alles viel schneller lösen kann, da er eine viel grössere Leistungsfähigkeit hat. Schliesslich kann man durch die herausgefundenen Primzahlen Nachrichten vom Opfer entschlüsseln und bekommt hier wiederum neue Hinweise zu den Verdächtigen.

---

### 6.3.7. Posten 7: Druck

In dem Posten Nummer 7 geht es um das Thema Druck. Hier wird im Wald ein Fussabdruck (siehe Abbildung 33) in einer Betonmasse gefunden, welcher vom Täter stammen könnte. Damit man mehr über diesen herausfinden kann, beschäftigt man sich mit der Grösse und der Tiefe des Fussabdruckes. Erklärt werden die physikalischen Mechanismen der Gewichtskraft auf unterschiedlichen Materialien. Man wird aufgefordert, sich zu überlegen, durch welche Eigenschaften diese Faktoren verändert werden.

Schlussendlich erhält man eine Tabelle (siehe Abbildung 34), mit deren Hilfe man die geschätzte Abdrucktiefe einordnen kann und somit das Gewicht ablesen kann.



Abbildung 33: Schuhabdruck in Beton

**Tabelle zum Schuhabdruck**

Vertiefung	Gewicht
1 cm	50 - 60 kg
3 cm	60 - 70 kg
5 cm	70 - 80 kg
7 cm	80 - 90 kg

Abbildung 34: Tabelle zum Schuhabdruck

### 6.3.8. Posten 8: Mechanik

Im achten Posten wird das letzte physikalische Thema behandelt, welches sich mit der Mechanik befasst. Es wird von der Polizei festgestellt, dass das Opfer auf einer Brücke erschossen wurde, danach heruntergefallen ist und schliesslich am Sihlufer angeschwemmt wurde. Man ist also endlich am Tatort (siehe Abbildung 35 und Abbildung 36) angekommen. Die Ballistik hat schon herausgefunden, mit welchem Waffenmodell geschossen wurde. Mit Hilfe der Information der Durchschnittsgeschwindigkeit der Kugel und die Dauer des Schusses kann man die Entfernung vom Täter zum Opfer berechnen. Hierzu benötigt man folgende Formel:

$$s = v * t$$

Dazu bekommt man auch noch eine Grafik (siehe Abbildung 37), in der man bestimmen muss, wo der Täter gestanden haben muss. Ausserdem kann man aus dem Resultat ausschliessen, dass Julian Gutenberg Selbstmord begangen hat.



Abbildung 35: Verschmierter Blutabdruck einer linken Hand



Abbildung 36: Tatort

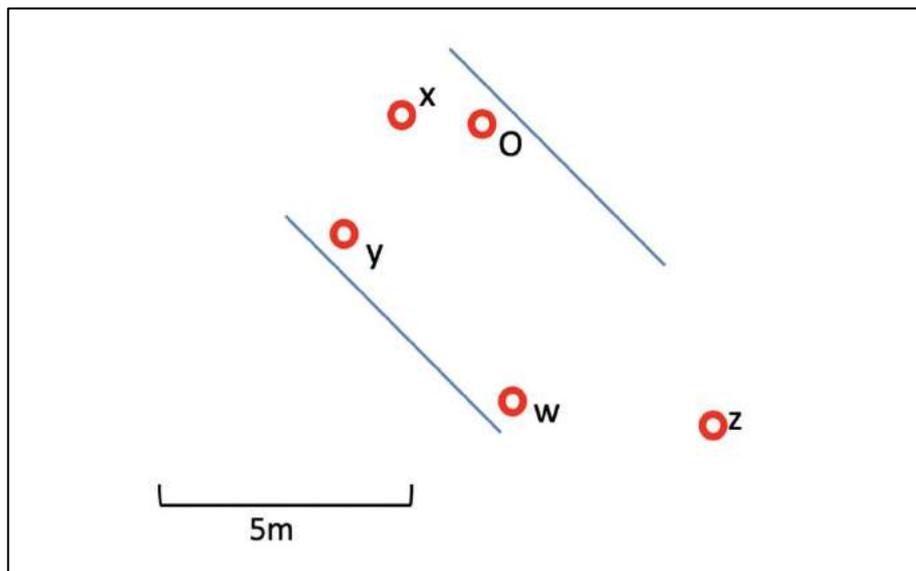


Abbildung 37: Grafik Tatort

### 6.3.9. Posten 9: Aufklärung

Dieser Posten ist der Letzte in meiner App und klärt den Fall schlussendlich auf. In den vorherigen Posten hat man ausreichend Informationen gesammelt, um den Mörder zu identifizieren und ihn zu verhaften.

In einer ersten Frage wählt man in der App den Mörder aus, der verhaftet werden soll.

Es gibt eine Zusatzfrage, die nach dem aktuellen Standort des Quantencomputers fragt. Beispielsweise ist hier eine Auswahlmöglichkeit, dass er an die Firma Techgeniosus verkauft wurde oder dass es ihn überhaupt nicht gibt.

Um an die Endauflösung zu gelangen, muss man eine Kiste mit einem speziellen Schloss öffnen. Die Anleitung hierfür hat man ganz am Anfang des Krimispiels erhalten. Den Code erfährt man im Verlaufe des Krimispiels. Wenn man dies geschafft hat, ist das Krimispiel zu Ende.



## 7. Vorgehensweise

In diesem Kapitel werde ich meine Vorgehensweise zur Entwicklung meiner App genauer erläutern. Diese Ausführungen habe ich in sechs Unterkapitel unterteilt. Zuerst beschreibe ich das gesamte Konzept, dann den organisatorischen Teil. Danach beziehe ich mich auf das App-Design und auf das Programmieren. Zum Schluss werte ich die Interviews und den Testdurchlauf aus, den ich mit meinen Freunden durchgeführt habe.

### 7.1. Konzept

#### 7.1.1. Route

Als ich mich am Anfang meiner Maturitätsarbeit dazu entschlossen habe, eine Art Rallye an der Sihl zu kreieren, musste ich mir auch die Frage stellen, wo genau an der Sihl alles stattfinden soll. So bin ich an mehreren Nachmittagen an der Sihl entlangspaziert und habe nach einer geeigneten Route für meine Arbeit gesucht. Von Anfang an war der Sihlwald eine mögliche Option für mich, aber auch in der Gegend in der Nähe der S-Bahn-Haltestelle Wildpark-Höfli fand ich passende Orte.

Einen Faktor, den ich bei der Wahl meiner Route berücksichtigen musste, war die Zeit. Wie lange sollte meine Rallye dauern? Dies ist schwer einzuschätzen und kommt auch auf die Nutzer an, wie schnell die Gruppe ist. Ich habe mir eine Zeitspanne von zwei bis drei Stunden Spielzeit vorgestellt. Denn in weniger Zeit kann man nur sehr wenige Aufgaben absolvieren, wenn man bedenkt, dass man sich von einem Posten zum anderen fortbewegen muss. Dann würde sich die Anreise eventuell nicht lohnen. Ich wollte meine Rallye aber auch nicht für einen Zeitraum von fünf Stunden oder länger planen, da es irgendwann langweilig wird und man dann nicht mehr mag.

Eine weitere Frage, die ich mir stellen musste, war, ob der Startpunkt derselbe wie der Endpunkt sein sollte oder nicht. Die Route soll gut zugänglich sein, vor allem mit dem öffentlichen Verkehr. Deswegen sollte ein Bahnhof in der Nähe stationiert sein. Die Strecke von einem Bahnhof zum nächsten ist zu weit zum Spaziergehen im Rahmen meiner Krimi-App. Zudem wäre es ungünstig, wenn man mit dem Auto unterwegs ist und mit dem Zug nachher eine Station zurückzufahren müsste, um wieder beim Standort des Autos zu sein. Somit muss die Route geschlossen sein.

Dieser Umstand „Startpunkt = Endpunkt“ hat mir das Problem mit den Brücken aufgebracht. Die „zum Spaziergehen schönere“ Seite der Sihl liegt nicht auf der Seite der Sihltalbahnstrecke. Um auf diese Seite der Sihl zu gelangen, muss man Brücken nutzen. Leider gibt es nur wenige Brücken über die Sihl, welche genug nah aneinander liegen, dass man einen Rundweg laufen könnte.

Als ich nun alle Faktoren betrachtet hatte, erschien mir der Sihlwald als die beste Lösung. Mein Fazit: Auch im Nachhinein finde ich, dass der Sihlwald perfekt für meine App geeignet ist. Es gibt eine Zugstation, positioniert zwischen zwei Brücken, welche genug nah aneinander gelegen sind. Zudem gibt es dort das Besucherzentrum im

---

Sihlwald, welches ein Museum, ein Restaurant, einen Spielplatz, ein Tiergehege und weitere Attraktivitäten anbietet. Es besteht somit die Möglichkeit, sich dort vor oder nach der Nutzung meiner App noch zu verpflegen, die Toilette aufzusuchen oder sich dort aufzuhalten. Weiterhin ist das Besucherzentrum im Sihlwald ein Ausgangspunkt für viele schöne Wanderwege an der Sihl und etwas weiter entfernt davon sind bereits Waldaktivitäten eingerichtet.

Es gibt zum Beispiel den Walderlebnispfad, der viele verschiedene Posten zum Erleben des Naturwaldes enthält und von dem ich einige Posten in meinen Parcours integrieren konnte. Schliesslich habe ich mich dann für eine abwechslungsreiche Route im Sihlwald entschieden (siehe Abbildung 38). Sie liegt teilweise an der Sihl, teilweise im Wald und teilweise auf dem Walderlebnispfad. Die Route ist sehr angenehm zum Laufen und hat eine sehr gut zu bewältigende Länge.

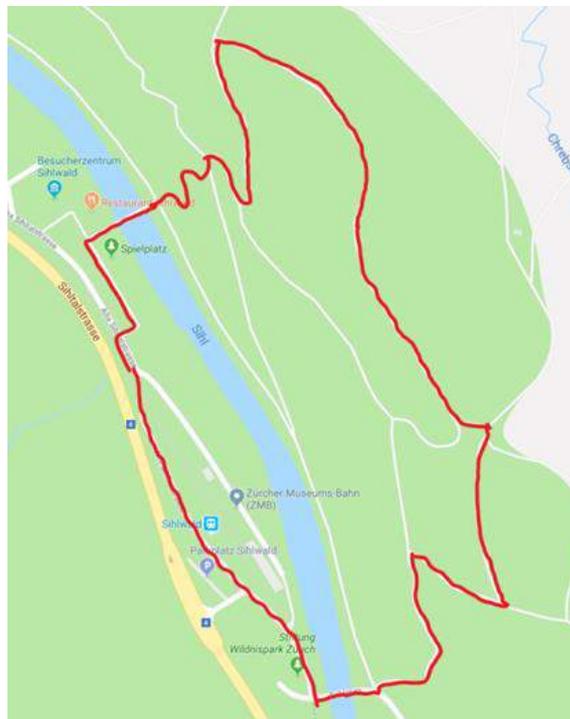


Abbildung 38: Route meiner App

Ausserdem möchte ich zur Bekanntheit des Besucherzentrums ein klein wenig beitragen, da ich dieses auch selbst (mit meiner Familie) gerne nutze.

Nachdem ich meine Location und Route festgelegt hatte, konnte ich mich um die Standorte der einzelnen Posten und deren Reihenfolge kümmern. Gewisse Posten müssen zwangsweise an der Sihl sein und so konnte ich diese zuerst festlegen. Dann habe ich die restlichen Posten so verteilt, dass die Reihenfolge abwechslungsreich wird und Sinn ergibt. Einige Posten benötigen bestimmte Materialien, für welche ich dann im Wald Verstecke gesucht habe, um diese dort in Dosen zu deponieren. Dazu werde ich im Kapitel 7.2 auf der Seite 37 noch genauer eingehen.

### 7.1.2. Physik-Elemente

Als ich das Thema meiner Maturitätsarbeit beschloss, wollte ich unbedingt die Physik integrieren. So musste ich mir auch überlegen, wie ich diese am besten einbinden könnte. Also habe ich mich durch viele verschiedene Bücher und drei Interviews inspirieren lassen und habe ein physikalisches Programm zusammengestellt. Hierbei habe ich darauf geachtet, dass die physikalischen Aspekte nicht zu schwer sind, sodass der „durchschnittlich gebildete Bürger“ diese auch verstehen kann. Zudem war es wichtig, dass man die Experimente im Wald durchführen kann und dafür möglichst wenig Material benötigt. Auch soll das Programm möglichst vielseitig sein und viele verschiedene Themen behandeln. Zum Schluss wäre es praktisch, wenn man die Physik irgendwie in Verbindung mit dem Kriminalfall bringen könnte. Mit diesen Anforderungen habe ich verschiedenste Phänomene und Experimente angeschaut und aussortiert, bis ich zu meiner jetzigen Auswahl kam. In meine Mordstory wollte ich ein aktuelles Forschungsthema einbauen.

Damit in meiner Mordstory der Bezug zum Thema Physik gut gelingt, wollte ich einen Physiker als Protagonisten mitwirken lassen, der in der Forschung tätig ist und über neueste Erkenntnisse verfügt. Diese sind bekanntlich nicht nur unter Fachleuten begehrt, sondern auch bei den Personen und Unternehmen, die die Informationen zu Geld machen wollen. Somit wird mein Thema bereits personell im Krimi implementiert. Weiterhin vermute ich, dass dann die Einbindung von physikalischen Experimenten und Erkenntnissen besser gelingen wird.

Zunächst habe ich an die Forschung am CERN gedacht. In Genf befindet sich das CERN (Conseil Européen pour la Recherche Nucléaire), der grösste Teilchenbeschleuniger der Welt.[19]

Dort wird in der theoretischen Physik bzw. Elementarteilchenphysik geforscht. In einem Experiment namens LHCb wurden zum Beispiel Abweichungen vom Standardmodell, welches den Bauplan unseres Universums beschreibt, gefunden. Es sind Abweichungen beim Zerfall des Bottom-Quarks in die Leptonen Myon und Tau aufgefallen, die auf eine neue Kraft im Standardmodell weisen könnten.[20] Vielleicht könnte mein Physiker die Lösung der Gravitation, eine Einsicht in die kosmisch relevante dunkle Materie oder die noch rätselhaftere dunkle Energie herausgefunden haben.

Ausserdem finde ich im Zeitalter der Digitalisierung den Hype um die Quantencomputer faszinierend, die anstatt mit Bits, die jeweils „an“ bzw. „1“ und „aus“ bzw. „0“ annehmen, mit mehr QBits arbeiten. In den Quantensystemen verhalten diese sich so, als könnten sie in allen möglichen Zuständen gleichzeitig existieren.[21]

Physiker im IBM-Forschungslabor in Rüslikon forschen an Quantencomputer und haben auch ein funktionsfähiges Modell entwickelt und gebaut. Dieser muss jedoch bis auf ein paar Kelvin vor dem absoluten Nullpunkt heruntergekühlt werden, um zu funktionieren.[22]

Da ich glücklicherweise die Gelegenheit hatte, mit einem IBM-Fellow direkt zu sprechen, fand ich das Thema Quantencomputertechnologie spannender und habe mich dafür entschieden.

---

### 7.1.3. Story

Die Konstruktion einer Mordstory ist keine Kompetenz, die wir im Schulunterricht oder im alltäglichen Leben erlernen. Ich habe mir viele Vorüberlegungen gemacht. Wichtig ist dabei, dass man sich zunächst eine so genannte Kernfrage überlegen muss, worum es in der Geschichte wirklich gehen sollte. Ausserdem müssen fünf Punkte bzw. Fragen festgelegt werden:

- Held in Person des Aufklärers, hier der Nutzer
- Person des Täters
- Ziele von Held und Täter
- Was steht auf dem Spiel?
- Wie endet der Konflikt?

Diese Vorüberlegungen dienen als Roter Faden, welcher zur Planung einer ausgewogenen Story hilfreich ist.[23]

Mir eine gute Mordstory zu überlegen, hat mir sehr viel Spass gemacht, obwohl es nicht immer ganz einfach war. Das Interview mit Heike Riel hat mir sehr geholfen, die Physik in die Mordstory einzubauen. So ist der Teil mit dem Quantencomputer in die Geschichte mit eingebunden worden. Danach habe ich einfach angefangen, mir weitere Verdächtige auszudenken und das Geschehen weiterzuentwickeln.

Wichtig war mir hierbei, dass der Ablauf der Geschichte nachvollziehbar ist. Hierbei ist es mir auch ein paar Mal passiert, dass die Story zu komplex geworden ist und ich selber den Überblick verloren hatte. Gegen den Schluss der Arbeit war es sehr schwer einzuschätzen, ob die Hinweise in meiner Krimi-App genug gewichtet auf den Mörder hinweisen. Die Nutzer sollen schliesslich herausfinden können, wer Julian umgebracht hat. Gleichzeitig durfte es auch nicht zu offensichtlich sein, wer der Mörder ist. So musste ich immer viel überlegen und zum Teil auch wieder gute Ideen verwerfen. Am Schluss habe ich mir noch eine Zusatzfrage für die ganz Aufmerksamen überlegt, welche nur gelöst werden kann, wenn man alle Hinweise beachtet und zusammensetzt.

Ein zusätzlicher Faktor, den ich berücksichtigen wollte, waren meine Physikposten. Diese wollte ich möglichst gut und sinnvoll in die Mordstory integrieren. Bei gewissen Posten hat dies besser funktioniert, bei anderen weniger gut. Schliesslich hoffe ich jedoch, dass die Geschichte spannend ist und der Schwierigkeitsgrad beim Lösen angemessen ist.

### 7.1.4. Namensgebung und App-Icon

Jede App braucht auch einen Namen. So habe ich lange nach einem passenden gesucht. Erste Ideen waren zum Beispiel die Akte mit dem Nachnamen des Opfers oder der Quantencomputerfall. Aber wieso so weit in die Ferne schweifen, wenn das Gute doch so nah liegt. Letztendlich habe ich meine App „Das Notizbuch“ genannt, da das Notizbuch von Julian Gutenberg, welches am Anfang bei der Leiche gefunden wurde, eine wichtige Rolle in meiner App spielt. Einerseits gibt es das Notizbuch in der App, in dem sofort alles Wichtige abgespeichert wird. Andererseits findet man im Verlaufe der Zeit

herausgerissene Seiten aus dem Notizbuch mit Informationen, welche beim Aufklären des Kriminalfalles wichtig sind.

Deshalb finde ich, dass das Notizbuch ein Schlüsselobjekt in meinem Krimispiel ist und einen passenden Namen und damit den Titel meiner Maturitätsarbeit abgibt.

Ich musste bei der Namengebung jedoch auch aufpassen, dass der Name nicht zu lang sein würde, da er sonst nicht mehr unter das App-Icon passt, wie zum Beispiel die Bezeichnung „Julians Notizbuch“.

Mein ausgewählter App-Name hat für mich einen einzigen Nachteil. Wenn man im App-Store nach „Das Notizbuch“ sucht, werden erst einmal viele andere Apps als Suchergebnis erscheinen, bis meine App in der Ergebnisliste auftaucht. Jedoch empfinde ich diesen Umstand als nicht sehr gravierend und habe mich am Schluss trotzdem für diesen Titel entschieden.

Auch auf meinem App-Icon befindet sich ein Notizbuch. Die Farbe des grünen Hintergrunds ist derselbe Grünton, welchen ich in meiner App verwende. Zudem symbolisiert die Farbe Grün die Natur, sie steht einerseits für den Wald, aber andererseits symbolisiert Grün die Hoffnung. Schliesslich kommt auf das Notizbuch noch ein Blutspritzer hinauf, der den Mord symbolisieren soll. So sind alle Aspekte meiner Arbeit in meinem Icon enthalten. (siehe Abbildung 39).



Abbildung 39: App-Icon

### 7.2. Organisation im Sihlwald

In diesem Abschnitt möchte ich genauer auf den administrativen und rechtlichen Teil meiner Arbeit eingehen.

Um meine Arbeit im Sihlwald durchführen zu können, habe ich mich zuerst erkundigt, ob ich dazu eine amtliche Bewilligung brauche. Ich habe einige Telefongespräche durchgeführt, bis ich überhaupt die richtige Ansprechperson für mein Anliegen gefunden hatte. Für mein Anliegen gibt es ein Bewilligungsverfahren, was ich durchlaufen habe, nachdem ich einige Ämter kontaktiert hatte.

---

Nach Art. 699 ZGB gibt es ein freies Zutrittsrecht zum Wald für alle Menschen. Jedoch gilt das nur innerhalb bestimmter Grenzen. Die Natur muss vor Beeinträchtigungen und unverhältnismässiger Nutzung geschützt werden. Bei besonderen Aktivitäten muss deswegen eine Bewilligung eingeholt werden. (§ 15 Bundesgesetz über den Wald in Verbindung mit dem Kantonalem Waldgesetz).[24]

Zwar sind zum Beispiel Exkursionen von Gruppen im ortsüblichen Umfang erlaubt, aber wenn die Infrastruktur verändert wird oder Terrainveränderungen auftreten, seien sie auch noch so klein, wird die so genannte Duldungspflicht im Rahmen eines allgemeinen Betretungsrechts überschritten.[25]

Ich musste ein Exposé (siehe im Anhang Kapitel 13.4.1, S.229) für den Wildnispark Zürich verfassen, indem ich kurz meine geplante Arbeit vorstellte. Daraufhin wurde mein Antrag bewilligt. Ich musste dann daraufhin noch ein offizielles Koordinationsblatt ausfüllen (siehe im Anhang Kapitel 13.4.2, S.233).

Dann habe ich mich mit meiner Ansprechperson Nicole Aebli, einer so genannten Rangerin des Wildnisparcs Zürich, getroffen, um genau abzuklären, was ich im Wald deponieren und montieren darf. Das Gespräch verlief sehr gut. Frau Aebli war auch sehr freundlich und mir wohlgesinnt. Wir beide hatten ähnliche Vorstellungen in Bezug auf die Infrastrukturveränderungen und haben uns darauf geeinigt, dass ich grundsätzlich alles im Wald deponieren darf, was nicht überdimensioniert ist und einfach wieder zu entfernen ist. So konnte ich zum Beispiel kleine Kisten mit Materialien für die Physikposten im Wald verstecken, ähnlich wie die Geocache-Verstecke.

Damit meine Verstecke jedoch nicht an ungeeigneten Stellen sind und keinen Tieren schaden könnten, hat Frau Aebli darum gebeten, diese vor dem Aufstellen überprüfen zu können. Nach den Sommerferien bin ich in den Sihlwald gegangen und habe verschiedene Verstecke für die Posten gesucht, bei denen bestimmte Materialien oder Dokumente benötigt werden. Damit ich diese auch wiederfinde und Frau Aebli sie kontrollieren kann, habe ich Fotos (siehe Abbildung 40 bis Abbildung 43) aufgenommen und mir die Koordinaten notiert, welche ich auch in meiner App einbinden wollte. Da die Koordinaten jedoch sehr ungenau waren, habe ich noch eine Karte (siehe Abbildung 44) angefertigt, mit der Frau Aebli schliesslich die Verstecke finden und mir ein Feedback geben konnte. Einige Verstecke waren ungeeignet und ich musste noch einmal Neue suchen. Am Schluss konnte ich alle verbindlich festlegen.



Abbildung 40: Versteck 1 bei Posten 2



Abbildung 42: Versteck 3 bei Posten 4



Abbildung 41: Versteck 2 bei Posten 4



Abbildung 43: Versteck 4 bei Posten 6



Abbildung 44: Karte im Sihlwald mit eingezeichneten Verstecken

## 7.3. Design der App

Damit meine App nicht nur funktionell ist, sondern auch optisch schön anzuschauen ist, habe ich mir Vorüberlegungen in Bezug auf das App-Design überlegt. Das App-Design ist für den Nutzer sehr wichtig, da ein gutes Design die Nutzerfreundlichkeit erhöht. Zudem macht es einfach viel mehr Spass mit einer optisch ansprechenden App zu arbeiten als mit einer App, die nur aus Funktionen besteht und gestalterisch nicht viel hermacht.

### 7.3.1. Platzierung der Elemente

Damit man sich gut in der App orientieren kann, habe ich Toolbars auf jedem Screen eingefügt. Diese Toolbars sind je ein Balken am oberen Bildschirmrand und einer am unteren. Diese sind konstant vorhanden und beinhalten die Buttons zu den einzelnen Funktionen.

In der oberen Toolbar steht der jeweilige Titel des aktuellen Postens. Zudem kann man dort über einen Pfeil zurück auf die letzte Seite gelangen. Die untere Toolbar hat drei Buttons.

Der eine hat die Funktion, auf die nächste Seite zu gehen. In manchen Fällen muss dieser Button zuerst freigeschaltet werden, bevor man ihn benutzen kann. Dann gibt es noch den Notizbuchbutton in dieser Toolbar. Über diesen Button gelangt man in das Notizbuch, in dem alles Wichtige abgespeichert wird. Der letzte Button führt zu dem Navigationspfeil, der dem Nutzer anzeigt, wohin er gehen muss.

### 7.3.2. Farben

Zu dem Thema Design gehört auch die Farbgestaltung. Weil die Nutzung meiner Krimi-App mit einem Waldspaziergang verbunden ist und viel mit der Natur zu tun hat, habe ich mir die Farbe Grün als Hauptfarbe ausgesucht. So sind zum Beispiel die beiden Toolbars in einem Grünton dargestellt. Wenn man einen Button auswählt, färbt dieser sich dunkelgrün. Dadurch bemerkt der Nutzer das Anwählen des Buttons. So habe ich verschieden starke Grüntöne ausgesucht, welche ich innerhalb der App immer für die gleiche Funktionalität verwende.

### 7.3.3. Fotos

Fotos stehen im Zeitalter der Digitalisierung bei vielen Menschen tagtäglich im Vordergrund. Sie werden in sozialen Medien hochgeladen und ständig aktualisiert. Deshalb möchte ich ebenfalls Fotos einbinden und damit die Visualisierung der App stärken. Auch lockern tolle Bilder meine App auf, da sie sonst sehr textlastig wäre.

Das Leichenfundortfoto war für mich sehr wichtig, da es den Nutzern einen ersten Eindruck vermittelt. Es sollte möglichst authentisch sein und die Nutzer in den Kriminalfall hineinziehen. Ich musste lange nach einer geeigneten Person suchen. Sie sollte für meine Mordstory das richtige Alter haben und auch gewissermassen vom Aussehen her ins Schema passen. Zudem musste sich die Person dazu bereiterklären, sich in das kalte Wasser der Sihl für die Fotos zu legen. Mein Englischlehrer Marc Emery hat sich glücklicherweise als Darsteller der Leiche zur Verfügung gestellt. Wir haben uns an einem Samstagnachmittag getroffen, um die Fotos (siehe Abbildung 21 bis Abbildung 28, S.25 und S.26) aufzunehmen.

Bei der Tatortfotografie geht es insbesondere darum, möglichst genau den Zustand und die Verhältnisse um und am Ort eines Tatorts fotografisch zu dokumentieren. Man sollte dabei mindestens eine Übersichtsaufnahme der Umgebung und eine Detailaufnahme der Beweisstücke fertigen.[26] Beim Fototermin habe ich mir angemessen Zeit genommen und darauf geachtet, diese Vorgaben umzusetzen. Damit die Fotos möglichst echt aussahen, habe ich zuvor bei der Spurensicherung der Zürcher Kantonspolizei angefragt, ob ich von ihnen originale Nummerntafeln und Absperrband ausleihen dürfte. Dies wurde mir gestattet und ich konnte diese bei der Kriminalpolizeistation abholen.

Für die Leiche habe ich auch noch Kunstblut gekauft, um den Einschussbereich auf dem Brustkorb rot zu färben. Zudem habe ich auch noch weitere Fotos vom Fundort des zweiten Schuhs für den Auftriebsposten (siehe Abbildung 45 und Abbildung 46) angefertigt. Diesen Posten musste ich am Schluss aus Zeitgründen leider hinausnehmen, doch ich fand die Fotos trotzdem gelungen. Auch an dem Tatort selber,

---

den man ganz am Schluss findet, werden Blutspuren gefunden, von denen ich auch Fotos geschossen habe (siehe Abbildung 35 und Abbildung 36, S.31).



Abbildung 45: 7. rechter Schuh



Abbildung 46: rechter Schuh mit Umgebung

Für mich waren jedoch nicht nur die Leichen- und Tatortfotos wichtig, sondern auch die Visualisierung der einzelnen Personen. Dafür habe ich Fotos von den einzelnen Charakteren gefertigt und diese dann in die Personalakten integriert (siehe Kapitel 6.2.2, S.23).

Damit der Schluss, an dem man den Täter verhaftet, möglichst echt wirkt, habe ich auch hierfür spezielle Verhaftungsfotos geschossen (siehe Abbildung 47 bis Abbildung 50). Diese habe ich mit einer Vignette (Ecken des Fotos werden verdunkelt) bearbeitet und Schriftzeichen eingefügt.



Abbildung 47: Verhaftungsfoto Amelie Herzog



Abbildung 48: Verhaftungsfoto Emma Gutenberg



Abbildung 49: Verhaftungsfoto Oliver Gutenberg

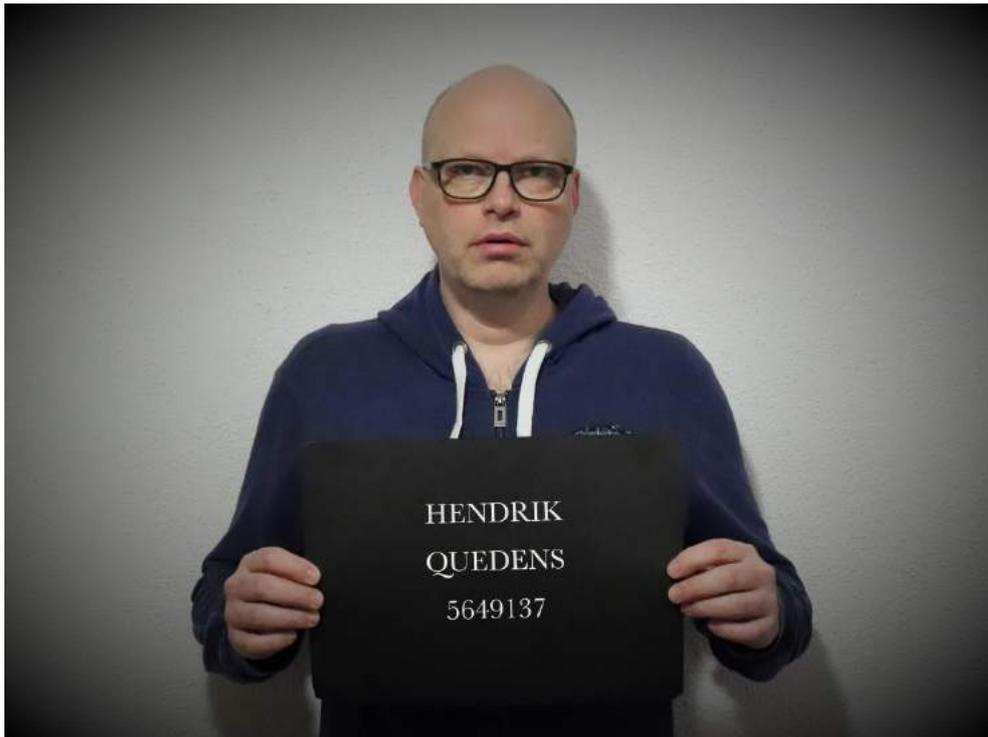


Abbildung 50: Verhaftungsfoto Hendrik Quedens

## 7.4. Programmierung

### 7.4.1. iOS-App

Wenn man sich dazu entschliesst, eine App zu programmieren, dann kommt an erster Stelle die Frage auf, ob es eine Android-App oder eine iOS-App sein soll. Heutzutage gibt es die meisten Apps für beide Betriebssysteme, jedoch ist dafür der Aufwand sehr gross, da man alles zweimal in verschiedenen Programmiersprachen programmieren muss. Da dies im Rahmen einer Maturitätsarbeit nicht möglich ist, vor allem für mich, da ich noch nie zuvor programmiert habe, musste ich mich für ein Betriebssystem entscheiden. In meiner Familie besitzen alle iPhones, welche auf dem iOS-Betriebssystem basieren und deswegen habe ich mich dazu entschieden, ein iOS-App zu programmieren. Zudem wird eine iOS-App mit der Programmiersprache Swift programmiert, welche laut Gian Ginetta (langjähriger Freund mit eigener Swift-Programmiererfahrung) einen guten Einstieg in die Programmierwelt bietet.

Diese Entscheidung, eine iOS-App zu programmieren, hat jedoch ein Problem mit sich gebracht. Denn wenn man in der Programmiersprache Swift programmiert, liegt dem das Programm Xcode zugrunde, welches nur auf einem macOS-Betriebssystem läuft. Ein solches besitzen wir Zuhause nicht. Xcode ist eine Programmierumgebung von Apple, mit der man Applications entwickeln kann. Man kann mit ihr Apps für iOS, watchOS, macOS oder tvOS in den Programmiersprachen Objective-C und Swift programmieren.

Obwohl wir alle Apple Handys haben, laufen all unsere Computer mit dem Betriebssystem von Windows. Damit ich mir kein MacBook organisieren muss, habe ich nach einer Alternative gesucht und bin schliesslich auf die Programme VMWare und VirtualBox gestossen. Diese beiden Programme erlauben es, ein Betriebssystem auf einem Rechner zu simulieren. So habe ich es durch eine lange Installation geschafft, auf meinem Windows-Computer einen simulierten Mac laufen zu lassen und darauf mit Xcode zu programmieren. Dies hat auch sehr lange gut funktioniert, bis es ein Programmupdate gegeben hat, welches mir nicht mehr ermöglicht hat, ein macOS Betriebssystem zu simulieren. Schlussendlich musste ich mir nach vielen Stunden vergeblichen Suchens nach einer Lösung doch noch ein MacBook besorgen.



Abbildung 51: Icon von Swift

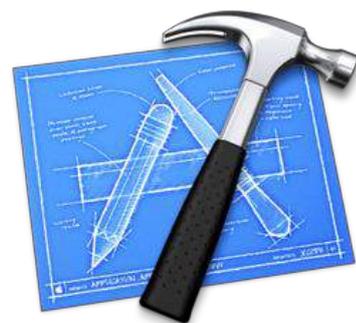


Abbildung 52: Icon von Xcode

---

## 7.4.2. Kleine Einführung

Damit man in diesem Kapitel nachvollziehen kann, wie ich agiert habe, möchte ich zuerst kurz die Struktur der Programmierung einer App aufzeigen und ein paar wichtige Begriffe erklären, welche ich im weitergehenden Teil gebrauchen werde.

Eine App ist immer auf zwei Teilen aufgebaut. Der erste Teil ist die Nutzeroberfläche oder auch Graphical User Interface (GUI) genannt, welches schlussendlich das ist, was in der App sichtbar ist. Die GUI bestimmt, was auf den verschiedenen Screens abgebildet wird oder wo sich bestimmte Buttons (Knöpfe) oder andere Objekte befinden. Diesen Teil konstruiert man in dem `main.storyboard`-Dokument. In diesem Dokument kann man `UIViewController` einfügen. Dies sind Objekte, welche einen Screen darstellen. Xcode definiert: „A controller that manages a view“. In diesem `UIViewController` kann man dann beliebige Buttons, Labels oder Textfelder einfügen. Alle Objekte in einem `UIViewController` können sehr individuell designt werden. Man kann Farbe, Grösse und vieles mehr selber bestimmen. Auch muss man Objekte mit sogenannten Constraints festsetzen. Das heisst, man muss die genaue Position eines Items innerhalb des Screens bestimmen und durch Orientierungskonstanten festlegen. Hierbei ist es schwer, die Constraints so zu setzen, dass die App möglichst gut auf Geräten aussieht, die andere Bildschirmgrößen besitzen. Wenn man nur ein Constraint falsch setzt oder vergisst, gibt es leider sofort eine Fehlermeldung. Deswegen ist der Zeitaufwand für das Designen eines Screens sehr gross.

Auf diese Art und Weise kann man viele verschiedene `UIViewController` mit ihrem individuellen Aussehen erstellen und das Interface der App bilden. Am Schluss ist es wichtig, dass man als Verbindung zwischen den einzelnen `UITableViewController` sogenannte Segues einrichtet. Segues bestimmen, welcher Screen nach dem aktuellen kommt. (siehe Abbildung 53)

Der zweite Teil einer App besteht aus dem Programmcode. Hierzu gibt es für jeden `UIViewController` ein `ViewController.swift`-File. In diesem `ViewController` kann man jeden View individuell programmieren und ihm spezielle Funktionen zuschreiben. Wichtig hierbei ist es, dass man den `UIViewController` mit dem entsprechenden `ViewController` verbindet, da sonst eine Fehlermeldung entsteht. Wenn man dies vergisst, kann das dazu führen, dass man sehr lange nach einem Fehler im Programmcode sucht und dann schliesslich feststellt, dass man bloss vergessen hat, den `UIViewController` mit dem entsprechenden `ViewController` zu verlinken.

Ein `ViewController` ist wiederum in zwei Teile aufgeteilt. Der erste Teil beinhaltet Informationen über den Controller. Hier kann man Texte verändern, Farben oder gewisse Eigenschaften des Controllers festlegen. Der zweite Teil ist der Aktionsteil. In diesem können bestimmte Funktionen definiert werden, welche ausgeführt werden, wenn man in der App auf den entsprechenden `UIViewController` gelangt. Auch werden hier zum Beispiel die Aktionen definiert, die ausgeführt werden, wenn man auf einen Button drückt.

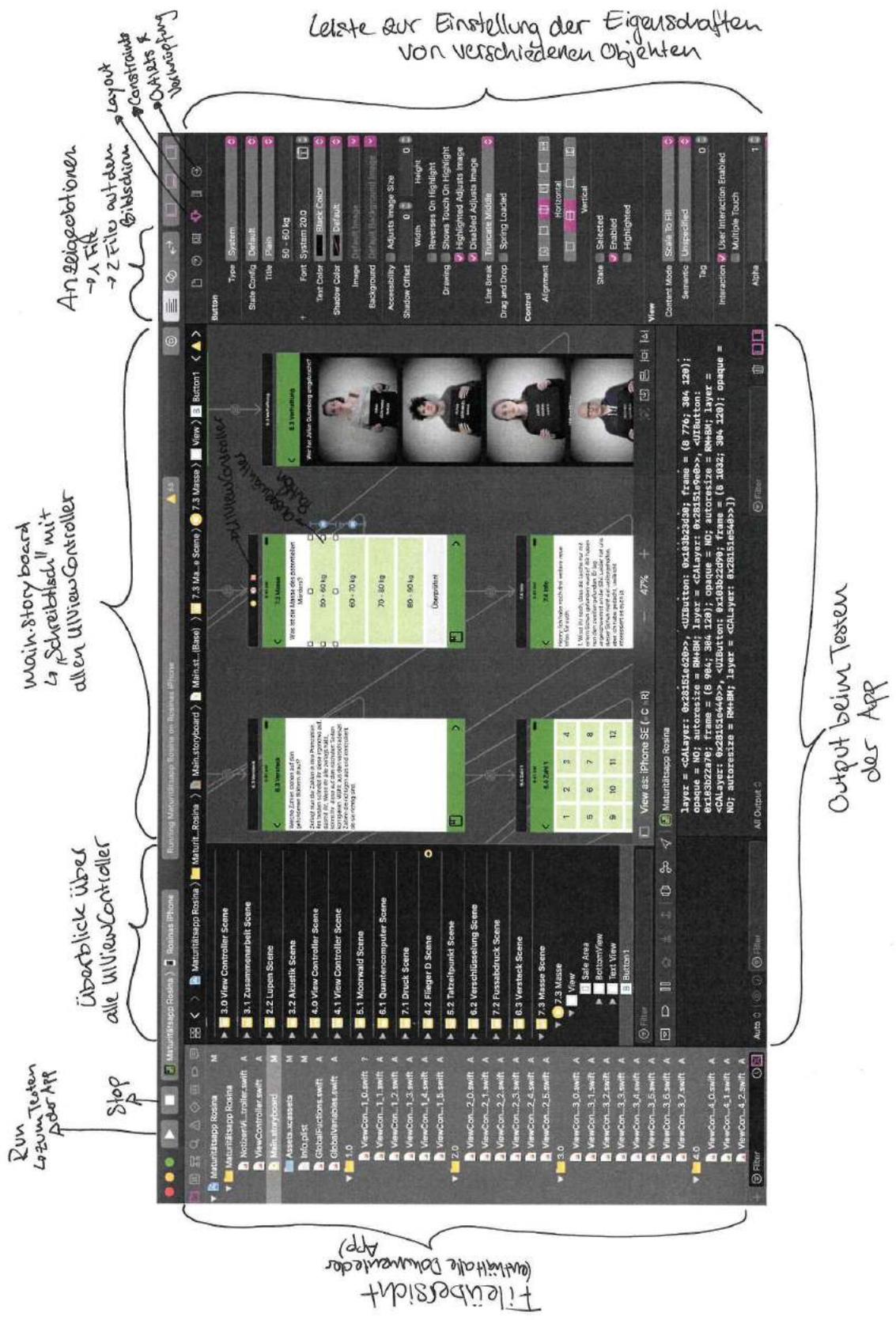


Abbildung 53: Screenshot Xcode Programmoberfläche im main.storyboard

---

### 7.4.3. Erlernen von Swift

Als ich das erste Mal Xcode gestartet hatte, hatte ich keine Ahnung, wie die Programmierumgebung, geschweige denn die Programmiersprache funktioniert. Ich hatte noch nie in meinem Leben zuvor etwas programmiert und habe mich einfach ins kalte Wasser geworfen, um dies zu ändern.

So habe ich mich mit der Hilfe von Gian Gentinetta, vielen Videos[27] und einem dicken Swift 4-Handbuch [28] ins Programmieren hineingearbeitet. Da man schliesslich am besten durch direktes Anwenden lernt, sind einige Probeapps entstanden, welche ein paar Funktionen meiner endgültigen App ausprobieren und in denen ich viel Neues entdecken konnte. So ist zum Beispiel die App „Rosinas Rosen App“ entstanden. Hier habe ich ausprobiert, wie man ein Quiz aufbauen kann.

Wie man in der Abbildung 54 sieht, befindet sich auf dem Screen der App in der Mitte ein Bild und man muss auswählen, welche Farbe die Blume auf dem Bild hat. Schliesslich drückt man auf einen der vier Buttons. Wenn man richtig geraten hat, kommt ein neues Bild mit einer neuen Blume darauf.

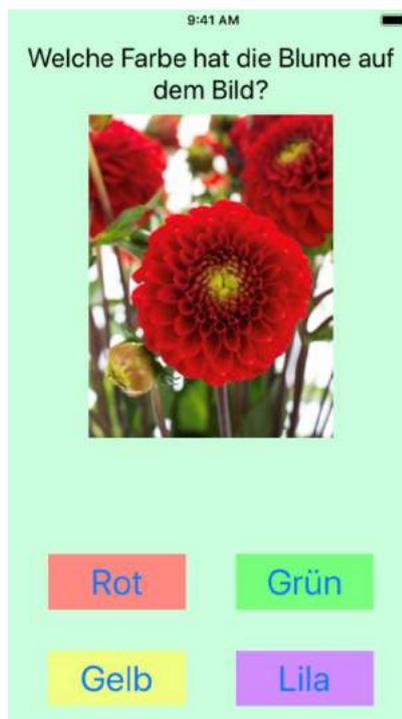


Abbildung 54: Rosinas Rosen-App

Zusätzlich habe ich noch eine Audiodatei abspielen lassen, wenn ein falscher Knopf gedrückt wurde. Hier muss ich jedoch anfügen, dass der allgemeine Inhalt irrelevant war und die Lösung nicht mit der wirklichen Farbe der Blume übereingestimmt hat. Es wurde durch Zufall bestimmt, ob die Antwort richtig oder falsch war. Es ging einfach darum auszuprobieren, wie man ein Bild neu lädt, wenn man auf den richtigen Knopf gedrückt hat. Ein ähnliches Vorgehen benutze ich auch in meiner endgültigen App innerhalb der Notizbuchfunktion.

Mit solchen kleinen Mini-Apps habe ich unterschiedliche Funktionen ausgetestet und kennen gelernt, wie zum Beispiel einen ScrollView programmiert wird. Ein ScrollView erlaubt, in einem View nach oben und unten zu scrollen.

Schliesslich habe ich mit meiner eigenen App angefangen. Auch in meinem Programmierprozess habe ich immer noch viel im Internet oder in meinem Handbuch nachgeschaut, Videoanleitungen angesehen oder konnte Gian um Rat fragen. Bei Fehlermeldungen habe ich oft im Internet nach möglichen Lösungsvorschlägen gesucht, vor allem das Forum Stackoverflow hat mir sehr geholfen.[29]

### 7.4.4. Technischer Aufbau der App

Für jeden Screen habe ich einen eigenen UIViewController angefertigt: Diese können im Anhang eingesehen werden. Jedoch verwende ich, wie im Kapitel 6.1 auf der Seite 17 beschrieben, verschiedene Funktionen. Jetzt werde ich die einzelnen Grundfunktionen erklären.

#### Navigation und Orientierung innerhalb der App

Damit man sich innerhalb der App gut zurechtfindet, benutze ich einen sogenannten UINavigationController. Dieser hat die Eigenschaft, dass er auf jedem Screen eine obere Tooleiste generiert, welche einen Pfeil aufweist. Dieser führt direkt zu der zuletzt besuchten Seite, ohne dass man irgendetwas navigieren muss. Diese Funktion ist sehr praktisch. Wenn man sich zum Beispiel auf dem Notizbuch befindet und zurück zu den Posten möchte, dann kommt man durch diesen Controller direkt auf den letzten Posten zurück, da er automatisch die zuletzt besuchte Seite aufruft. Wenn ich das nicht so programmiere, würde der Nutzer immer wieder von vorne starten.

#### Notizbuchfunktion

Mit der Notizbuchfunktion beschäftigt sich das File NotizenViewController.swift. Der Controller kümmert sich um das Notizbuch innerhalb der App, indem die vielen gesammelten Informationen abgespeichert werden. Hierbei ist speziell, dass ich nicht für jede Seite im Notizbuch einen einzelnen UIViewController programmiert habe, sondern nur einen einzigen. Dieser hat in der Mitte ein UIImageView (Objekt, in das man ein Bild hereinladen kann), Vorwärts- und Rückwärtsknöpfe und ein Label, welches die Seitenzahl angibt. In das UIImageView lädt man dann durch den Programmcode verschiedene Bilder hinein. Dies funktioniert dadurch, dass man von Anfang an angibt, wie viele Seiten das Buch hat und dann mit den Pfeilen das folgende oder letzte Bild aufruft, indem man den Screen aktualisiert.

Das Schwierige bei dieser Funktion ist, dass man gewisse Seiten hat, die überschrieben werden sollen. Ein Beispiel hierfür sind die Personalakten, bei denen ständig neue Informationen hinzukommen. Diese müssen deswegen immer aktualisiert werden. Dazu habe ich einen Array (Ansammlung von Daten) eingeführt, in welchem man für jede Seitenzahl die geplanten Bilder angibt und dazuschreibt, bei welchem Posten diese freigeschaltet werden. So muss man in den einzelnen ViewController der Posten jedoch am richtigen Ort die Bilder freischalten, damit die Aktualisierung stattfinden kann.

---

Schliesslich habe ich für die Bilder eine Zoommöglichkeit eingefügt. Dazu habe ich einen ScrollView, der eine Scroll- und Zoomfunktion hat, eingefügt und diesen dann entsprechend aktiviert.[30]

Zusätzlich habe ich mit einer if-Funktion erreicht, dass der back-Button versteckt ist, wenn man auf der ersten Seite ist und dass der forward-Button versteckt ist, wenn man auf der letzten Seite angekommen ist. Sobald man das Notizbuch verlässt, wird die zuletzt besuchte Seite gespeichert und man kommt auf diese, wenn man das Notizbuch wieder besucht. Dies habe ich durch die UserDefaults bewerkstelligt, welche für das Speichern zuständig sind.

### **Navigationsfunktion in der Umgebung**

Die Navigationsfunktion ist eine der aufwändigsten Funktionen meiner App. Der Sinn dahinter ist, dass in einem normalen View ein Pfeil gezeichnet wird und dieser in die Richtung zu der nächsten Position, zu der man gehen soll, zeigen soll. Zudem gibt es verschiedene Labels, welche die gesuchte Position, die aktuelle Position, den Titel und die aktuelle Entfernung zur nächsten Position anzeigen. Die verschiedenen Positionen werden durch Koordinaten angegeben. Auch hier hat man nur einen UIViewController, der sich je nach Fortschritt in der App aktualisiert und die entsprechenden Informationen liefert. Hier werden die Koordinaten von einem Posten in den jeweiligen ViewController eingespeichert und durch ein Informationssegue (transportiert gewisse Informationen von einem Controller zum nächsten) an den NavigationViewController weitergegeben, welcher sich um das Navigationsgerät kümmert. Da man hier den Standort der Nutzer benötigt, muss der Nutzer am Anfang der App erlauben, dass der Standort benutzt werden darf. Dies passiert in meinem allerersten ViewController, welcher mit dem Überblickscreen verknüpft ist. Wenn man die App startet, erscheint dort eine Meldung, bei der man die Standortfreigabe bewilligt. Ohne diese Erlaubnis des Nutzers funktioniert es nicht.

Um die Position gut abzuspeichern, habe ich zuerst das File Position.swift erstellt, indem ich den Struct oder auch die Klasse „Position“ definiere und deren Eigenschaften darlege. Ich habe die Möglichkeit genutzt, bereits fertig programmierten Code für diese Funktion aus meinem Swift 4-Handbuch herauszukopieren.[31] Damit die Koordinaten der Position umgerechnet werden können, habe ich in einem File für globale Funktionen (GlobalFuctions.swift) zwei Umrechnungsfunktionen erstellt, die aus einer Dezimalzahl eine Zahl mit Bogenminuten und –sekunden entwickelt.

Als weiteres habe ich ein File namens ArrowView.swift angelegt, welches sich um die graphische Konturierung des Richtungspfeils kümmert. Auch hierbei konnte ich von vorgefertigtem Programmcode profitieren.[32] Dies wird durch eine Draw-Methode bewerkstelligt.

Schliesslich habe ich meinem NavigationViewController programmiert. Dieser beschäftigt sich erstens damit, die Entfernung zwischen dem Ziel und der aktuellen Position zu berechnen und zweitens die Richtung vom Standort zum Ziel zu berechnen, damit der Richtungspfeil weiss, in welche Richtung er zeigen muss. Ursprünglich habe ich den Code für beide Funktionen aus meinem Swift 4-Handbuch[33] übernommen, doch es hat sich herausgestellt, dass die Berechnung der Richtung nicht funktioniert, so

wie es im Buch steht. Deswegen habe ich mir dort eine eigene Lösung überlegt. Dies habe ich bewerkstelligt, in dem ich die Approximation gemacht habe, dass die Erde flach ist. Schliesslich kann man durch den Tangens von den Zielkoordinaten minus die aktuellen Koordinaten den Richtungswinkel berechnen (siehe Abbildung 55).

```
//Richtung vom Standort zum Ziel berechnen
let rad = atan2((pos.long -
  manager.location!.coordinate.longitude)*3.75, (pos.lat -
  manager.location!.coordinate.latitude)*5.5)
print(heading)
heading = rad * 180 / .pi
```

Abbildung 55: Ausschnitt aus UINavigationController zur Richtungsberechnung

Man kann bei geringen Entfernungen die Erdkrümmung vernachlässigen, weil sie so klein ist, dass sie keine Rolle bei der Navigation spielt. So kann ich auf einfache Art und Weise mit dem Tangens von den Breiten- und Längengraden einer Position arbeiten.

### Quizfunktion

Für die Quizfunktion habe ich einen Grundbaustein erstellt, den ich dann bei allen anderen Quizaufgaben in leicht abgeänderter Form verwenden konnte. Ein Quiz ist immer so aufgebaut, dass es eine Frage und dann mehrere Auswahlmöglichkeiten als Lösung gibt, also wie bei einem Multiples Choice Quiz.

Am Anfang jedes Quiz gebe ich zwei Variablen an. Eine heisst pressedButtons und diese speichert, welche Buttons angewählt sind oder nicht. Sie bewerkstelligt auch, dass alle Buttons am Anfang deaktiviert sind. In der zweiten Variablen ist die richtige Lösung eingespeichert. Sie dient zur Überprüfung, welcher Button ausgewählt werden muss, damit das Quiz korrekt gelöst wird. Wenn man eine mögliche Antwort auswählt, wechselt der Button die Farbe und symbolisiert somit, dass er angewählt wurde. Schliesslich wird bei jedem Button am Schluss überprüft, welche Farbe er hat. Dadurch weiss man, welche Buttons aktiviert wurden und kann dieses Ergebnis mit der richtigen Lösung vergleichen. Wenn man das Quiz richtig gelöst hat, bekommt man einen Alert (Push-up-Nachricht) mit der Information, dass man das Quiz gelöst hat und wieso. Hierfür habe ich in meinem GlobalFuctions.swift-File einen zufälligen Komplimentgenerator entwickelt, welcher zufällig ein Kompliment als Titel der Nachricht auswählt. Dies hat den Sinn, dass die Nutzer gelobt werden, wenn sie etwas richtig gelöst haben. Für dieses Lob habe ich mehrere unterschiedliche Texte geschrieben, damit es für die Nutzer abwechslungsreich ist. Etwas Ähnliches passiert auch bei einem falschen Ergebnis. Hier bekommt man eine Nachricht mit einem zufälligen Titel, dass man etwas falsch gemacht hat und es doch noch einmal probieren soll.

---

## Freischaltungs- und Speicherfunktion

Die letzte grosse Funktion in meiner App ist die Speicherfunktion, die fast überall vorkommt. Diese ist sehr einfach durch die UserDefaults umsetzbar. Wenn man die Defaults an bestimmten Stellen freischaltet, ermöglichen diese, dass der Fortschritt der Nutzer abgespeichert wird.

Einmal kommt diese Funktion bei der Freischaltung der Posten vor. Somit sind die späteren Posten deaktiviert und könnten nur durch das Lösen der vorherigen Posten freigeschaltet werden. Dies bezweckt, dass man nicht einfach so Stationen überspringen kann.

Ein zweites Mal kommt diese Funktion, wie schon erwähnt, beim Notizbuch vor. Weiterhin befindet sie sich innerhalb der Posten bei dem Quiz. Hier ist der Button zum nächsten Screen zuerst auch deaktiviert und wird durch das erfolgreiche Lösen der Aufgaben aktiviert. Dadurch kann man die App fortsetzen. Dabei ist wichtig, dass der Fortschritt gespeichert wird. Das heisst, wenn man zurückgeht, muss man das Quiz nicht noch ein zweites Mal lösen, sondern kann einfach fortfahren.

## 7.5. Interviews

Im Juni als ich am Anfang meiner Arbeit stand, habe ich drei Interviews mit Physikern geführt. Von den Interviews habe ich mir versprochen, einen Einblick in die Arbeit eines Physikers zu erhalten. Mein physikalisches Verständnis erwarb ich im Schulunterricht. Innerhalb meiner App wollte ich jedoch realitätsnahe Physik präsentieren, praktisch „Physik zum Anfassen“. Im Rahmen des nationalen Zukunftstages, der jährlich stattfindet, habe ich meinen Vater, auch Physiker, besucht und dort einen Einblick auf das Teilgebiet der Laserphysik erhalten. Er konnte mir den Kontakt zu einem IBM-Fellow und einem Professor für Physikdidaktik herstellen. Ausserdem habe ich den Vorsitzenden der Naturforschenden Gesellschaft Zürich kontaktiert, der an einer Veranstaltung publikumsbeeindruckende Experimente vorgeführt hat. Diese Interviews sind in schriftlicher Form im Anhang angefügt (siehe Kapitel 13.5, S.240).

### 7.5.1. Interview mit Heike Riel

Das erste Interview habe ich mit Dr. Heike E. Riel durchgeführt. Sie ist seit mehreren Jahren ein IBM-Fellow und forscht bei den neusten Projekten mit. Durch sie habe ich ein physikalisches Element in meine Mordstory integriert, den Quantencomputer. Sie hat mir genau erklärt, wie ein solcher funktioniert und wie ich diesen bestmöglich in meinen Kriminalfall integrieren kann. Mich hat dieses allerneuste Forschungsprojekt sehr beeindruckt.

### 7.5.2. Interview mit Jan-Peter Meyn

Prof. Dr. Jan-Peter Meyn, Professor für Physikdidaktik an der Universität Erlangen-Nürnberg, hat ein Buch über Physikdidaktik geschrieben. Mit ihm habe ich die didaktische Seite meiner Arbeit besprochen und wie ich diese verbessern kann. Er hat mir erklärt, wieso ich als Zielperson einen „Otto-Normal-Bürger“ als Vorbild nehmen sollte und kein Wissen voraussetzen sollte.

### 7.5.3. Interview mit Fritz Gassmann

Schliesslich habe ich mit Dr. Fritz Gassmann gesprochen. Mit ihm habe ich verschiedene Ansätze für physikalische Experimente für meine Posten gesucht. Dieses Interview hat mir viele Anregungen geboten und mir geholfen, die physikalischen Aspekte meiner Arbeit festzulegen.

## 7.6. Testen der App

Als eine erste vollständige Version meiner App fertig war, habe ich mit meinen Klassenkollegen Noémie Veuve, Robin Feuz und Jonas Maurer an einem Dienstagnachmittag bei Nieselregen meine App ausprobiert.

Insgesamt haben wir zweieinhalb Stunden benötigt, um die ganze App durchzuspielen. Durch diesen Test konnte ich herausfinden, wo die App noch Fehler oder Defizite aufwies, wo gewisse Inhalte keinen Sinn ergaben und ob die Mordstory überhaupt lösbar oder viel zu einfach war. Am Schluss hatten sie den richtigen Mörder nicht bestimmt, doch sie waren sehr nah am Ergebnis dran. Alle drei hatten angegeben, dass ihnen die App und deren Inhalte sehr gut gefallen hat und sie die Mordstory komplex und spannend fanden. Der Aerodynamikposten mit dem Basteln der Papierflieger hatte ihnen am besten gefallen, so wie mir persönlich auch. Noémie hatte dem hinzugefügt, dass ihr einfach auch das Laufen gefallen hatte, während man von einem Posten zum nächsten gelangte. Sie empfand den Weg, den ich ausgesucht hatte, als sehr schön und friedlich und genoss es, an der frischen Luft über mögliche Motive des Täters nachzudenken.

Der Probedurchlauf hat mir bestätigt, dass meine App durchführbar ist. Ich habe wahrgenommen, dass die drei Testpersonen als Gruppe zusammengearbeitet hatten und bei einem netten Spaziergang viel Spass hatten. Sie hatten engagiert an meinem Kriminalfall gekniffelt und dabei noch einen kleinen Einblick in die physikalische Welt erhalten. Meine Ziele der App sehe ich als erfüllt an.



## 8. Schlusswort

### 8.1. Reflexion

Am Ende meiner Arbeit angekommen, kann ich feststellen, dass ich unglaublich viel im Laufe dieser Arbeit gelernt habe. Das Programmieren habe ich von Null auf gelernt und ich finde, dass ich mein persönliches Ziel der Entwicklung einer Krimi-App erreichen konnte. Zudem habe ich aber auch noch Schwierigkeiten bewältigt, wie zum Beispiel wichtige Entscheidungen zu treffen. Ich habe festgestellt, dass ich am Anfang eher die kleinen Sachen erledigt und für die Appentwicklung relevante Entscheidungen vor mir hergeschoben habe.

Ich kam an gewisse Punkte in meiner Arbeit, bei denen ich einfach wichtige Entscheidungen treffen musste. Dies war immer eine grosse Überwindung für mich, da ich dann wusste, dass ich nicht mehr frei ändern kann, was ich bereits festgesetzt hatte. Somit musste ich mich mit meiner Entscheidung abfinden. Damit fiel eine Belastung weg und ich konnte feststellen, dass ich nach dem Entschluss sehr erleichtert war. Erstaunlicherweise erfuhr ich einen Motivationsschub und durch das Treffen wichtiger Entscheidungen konnte vieles gleich miterledigt werden. Dieser Punkt hat zusammen mit der gesamten Organisation meiner Arbeit mein Selbstbewusstsein und viele andere Kompetenzen gestärkt.

Auch habe ich gelernt, Dinge zu akzeptieren, die nicht so funktionieren, wie sie sollten. Ich habe probiert dafür eine möglichst gute Alternative zu finden. Dies ist mir oft beim Programmieren passiert, aber auch manchmal in anderen Situationen. Zum Beispiel musste ich gegen Ende meiner Arbeit feststellen, dass ich einfach viel zu wenig Zeit hatte und so musste ich mich schweren Herzens dazu entscheiden, den Auftriebsposten zu streichen. Ich bedaure diesen Entschluss nicht, obwohl die viele Arbeit für den Posten als verloren bezeichnet werden kann.

Die Zeit ist etwas, was ich bei meiner Arbeit stark unterschätzt habe. Ich habe zwar sehr früh angefangen, doch war ich erst gegen Ende richtig effektiv in meinem Tun. Ich habe den Zeitaufwand des Programmierens stark unterschätzt. Auch über die Mühseligkeit von sehr einfachen Tätigkeiten, wie zum Beispiel dem Einfügen von Fotos, habe ich zeitlich nicht gut kalkuliert. Dies hat auch dazu geführt, dass ich mein Projekt erst im neuen Jahr mit einer für den App-Store hundertprozentig fertigen Version abschliessen kann. Es fehlen noch Tipps zu den Stationen innerhalb der App, die ich mir ursprünglich vorgenommen hatte einzufügen. Diese Einbindung von Tipps werde ich noch vor der geplanten Veröffentlichung tätigen.

Alles im Ganzen betrachtet, bin ich zufrieden mit mir und auch ein bisschen stolz, denn ich habe eine umfassende und abwechslungsreiche Arbeit angefertigt, dabei sehr viel gelernt und Spass gehabt.

---

## 8.2. Ausblick

Ich denke, dass man an meiner Arbeit noch ewig weiterarbeiten kann. Jedes Mal, wenn ich die App durchschaue, fällt mir wieder Neues auf, was man ändern oder verbessern könnte.

Einige Verbesserungen und Erweiterungen:

- Man könnte zum Beispiel die physikalischen Details besser herausheben, indem man einen Theorieteil integriert.
- Man könnte eine Erweiterung konstruieren, dass man zum Beispiel nach der Aufklärung den fliehenden Täter jagt und nach dessen Spuren sucht.
- Ebenso könnte man sich überlegen, verschiedene Schwierigkeitsstufen einzuführen. Zum Beispiel könnte es eine einfache Version für Kinder und eine anspruchsvollere Version für Jugendliche und Erwachsene geben.
- Auch könnte man eine Version für Schulklassen und grössere Gruppen konstruieren. Hierbei wäre daran zu denken, dass grössere Gruppen sich mit unterschiedlichen Aufgaben aufteilen könnten.
- Eine weitere Möglichkeit wäre, die ganze App ins Englische zu übersetzen, damit ausländische Touristen diese auch benutzen können.

Ich werde mich jedoch zukünftig zuerst darauf konzentrieren, dass ich Hilfestellungen in die App einbaue, damit man in der App weiterfahren kann, auch wenn etwas fehlt oder man eine Aufgabe überhaupt nicht schafft. Wenn diese Hilfestellungen in der App implementiert sind, würde ich sie gerne, wie von Anfang an geplant, in den App-Store hochladen, damit sie dann für alle zugänglich ist. In meiner jetzigen Version könnte ich mir dies zwar auch schon vorstellen, doch ich empfinde das Einbauen von Tipps als sehr wichtig. Zudem müssen meine Screen-Texte noch Korrektur gelesen werden. Sie enthalten noch ein paar Flüchtigkeitsfehler, welche ich diese noch nicht korrigiert hatte, da ich ursprünglich geplant hatte, diese vor dem Upload zu bearbeiten.

Wenn ich die App öffentlich zugänglich gemacht habe, bin ich sehr gespannt, wie sie genutzt wird, und hoffe auf gutes Feedback.

Ein weiterer Schritt nach der Veröffentlichung, wäre die Vermarktung der App. Dazu könnte man Werbung durch Plakate und Flyer aufgleisen oder eine Informationswebseite erstellen. Auf jeden Fall werde ich persönlich in meinem Wohnort Langnau am Albis in den drei Schulen und bei Vereinen meine App bekannt machen. Ich möchte, dass die lokale Bevölkerung, insbesondere die Jugendlichen, auf die App aufmerksam gemacht werden. Im Besucherzentrum Sihlwald können Flyer ausgelegt werden. Soziale Medien werden ebenfalls miteinbezogen.

Schliesslich werde ich mir auch einen Plan erstellen und umsetzen, in welchen Abständen ich die verschiedenen Verstecke kontrollieren werde, damit die Nutzer alles so vorfinden, wie ursprünglich geplant und nichts fehlt.

### 9. Danksagung

Als Allererstes möchte ich mich ganz herzlich bei meinem Betreuer und Physiklehrer Fabio Torriani für die grossartige Unterstützung und Hilfe, wann immer ich sie brauchte, bedanken.

An Gian Gentinetta geht auch ein grosses Dankeschön. Ohne ihn wäre meine Arbeit nicht das, was sie heute ist. Er hat mir geholfen einen Einstieg ins Programmieren zu finden und war jederzeit bereit, mir meine Fragen bezüglich Swift und Xcode zu beantworten.

Meiner Ansprechperson im Sihlwald, Nicole Aebli, möchte ich für all ihr Wohlwollen danken. Sie gab mir die Möglichkeit, meine Idee praktisch umzusetzen und im Sihlwald zu stationieren.

Auch bei meiner Familie möchte ich mich ganz herzlich für ihre Unterstützung und die ideenreichen Diskussionen bedanken.

Bei meiner Mutter, meinem Vater, meiner Schwester Marike und Robin Feuz bedanke ich mich für das Korrekturlesen meiner Arbeit.

Meiner Schwester Marike möchte dafür danken, dass ich mit ihr meine Mordstory besprechen konnte, obwohl ich sie dadurch gespoilert habe.

Meinem Englischlehrer Marc Emery möchte ich dafür danken, dass er sich dazu bereit erklärt hat, die Leiche auf meinen Leichenfundortfotos in der kalten Sihl liegend zu spielen. Sowie danke ich Cinja Müller, Gian Gentinetta, meiner Schwester Kerrin und meinem Vater dafür, dass ich von ihnen verschiedene Fotos für meine App aufnehmen konnte.

Noémie Veuve, Robin Feuz und Jonas Maurer danke ich für ihre Bereitschaft, die App mit mir zusammen an einem kalten Nachmittag bei Nieselregen auszuprobieren, und ihr facettenreiches Feedback.

Last but not least danke ich meinen drei Interviewpartnern Heike Riel, Jan-Peter Meyn und Fritz Gassmann für die interessanten und anregungsvollen Unterhaltungen in freundlicher Gesprächsatmosphäre.



## 10. Quellen- und Literaturverzeichnis

- [1] M. Kramer, "Physik als Abenteuer. Bd. 1. Didaktik, Akustik, Optik, E-Lehre und Kernphysik," Aulis-Verlag, 2011, p. 22.
- [2] J. Wittmann, "Physik in Wald und Flur: Beobachtungen und Gedanken eines Physikers in der freien Natur," Aulis-Verlag Deubner, 2000, p. 8.
- [3] M. Kommert, "55 Stundeneinstiege Physik einfach, kreativ, motivierend," Auer, 2016, p. 5.
- [4] H. D. Meyer Lothar, Schmidt Gerd-Dietrich, "Duden, Basiswissen Schule: Physik - Abitur," 3rd ed., Duden-Schulbuchverlag, 2011, p. 15.
- [5] G. Knecht, "Spielen, Leben, Lernen: Bildungschancen durch Spielmobile," Kopaed, 2011, p. 190.
- [6] C. Ucke and H.-J. Schlichting, "Spiel, Physik und Spaß Physik zum Mitdenken und Nachmachen," Wiley-VCH-Verlag, 2011, p. 1 und 2.
- [7] "Rangliste von Escape Rooms in der Schweiz - Lockme." [Online]. Available: <https://lockme.ch/escape-room-rangliste/>. [Accessed: 16-Dec-2018].
- [8] T. Graf, "Per Smartphone dem Mörder auf der Spur," Berner Zeitung, p. 2, 28-Aug-2018.
- [9] G. Knecht, "Spielen, Leben, Lernen: Bildungschancen durch Spielmobile," Kopaed, 2011, p. 75.
- [10] M. Birnthaler, "Praxisbuch Erlebnispädagogik," Verlag Freies Geistesleben, 2010, p. 16.
- [11] W. Stangl, "Immersion." [Online]. Available: <http://lexikon.stangl.eu/1939/immersion/>. [Accessed: 16-Dec-2018].
- [12] M. Kramer, "Naturphänomene: im Spiel die Physik erforschen," Aulis-Verlag Deubner, 2007, p. 96.
- [13] M. Kramer, "Naturphänomene: im Spiel die Physik erforschen," Aulis-Verlag Deubner, 2007, p. 76.
- [14] M. Kramer, "Naturphänomene: im Spiel die Physik erforschen," Aulis-Verlag Deubner, 2007, p. 122.
- [15] M. Kramer, "Physik als Abenteuer. Bd. 1. Didaktik, Akustik, Optik, E-Lehre und Kernphysik," Aulis-Verlag, 2011, p. 116.
- [16] C. Jargodzki and F. Potter, "Singender Schnee und verschwindende Elefanten : physikalische Rätsel und Paradoxien," Reclam, 2005, p. 152 und 153.
- [17] M. Kramer, "Naturphänomene: im Spiel die Physik erforschen," Aulis-Verlag Deubner, 2007, p. 107.
- [18] M. Homeister, "Quantum Computing verstehen: Grundlagen – Anwendungen – Perspektiven," Springer Fachmedien Wiesbaden, 2013, p. 194.
- [19] CERN, "Our History | CERN." [Online]. Available: <https://home.cern/about/who-we-are/our-history>. [Accessed: 16-Dec-2018].

- 
- [20] R. Breuer, "Spuren einer neuen Kraft - News Wissen: Technik - tagesanzeiger.ch," Tages-Anzeiger, p. 40, 25-Jul-2018.
- [21] U. von Rauchhaupt, "Zeichen der Zukunft," Frankfurter Allgemeine Sonntagszeitung, p. 59, 29-Apr-2018.
- [22] IBM, "Quantum computing at IBM - IBM Q - US," 2018. [Online]. Available: <https://www.research.ibm.com/ibm-q/learn/what-is-ibm-q/>. [Accessed: 16-Dec-2018].
- [23] M. Mansour, "Businessplan: Mord: erfolgreich einen Krimi schreiben.," Arisverlag, 2017, p. 111.
- [24] "SR 921.0 Bundesgesetz vom 4. Oktober 1991 über den Wald (Waldgesetz, WaG)." [Online]. Available: <https://www.admin.ch/opc/de/classified-compilation/19910255/index.html>. [Accessed: 16-Dec-2018].
- [25] H.-P. Stutz, H. Reifler, and S. Wegmann, "Veranstaltungen im Wald: Was müssen die Gemeinden tun?," Umweltpraxis Nr. 69, p. 13 - 16, 2012.
- [26] W. Zirk and G. Vordermaier, "Kriminaltechnik und Spurenkunde: Lehrbuch für Ausbildung und Praxis," Boorberg, 1998, p. 67.
- [27] Im Verlaufe meiner Arbeit habe ich so viele verschiedene Videos zum Programmieren angeschaut, dass es den Rahmen sprengen würde, sie hier alle aufzulisten. Manche Videos habe ich arch nur anteilmässig geschaut.
- [28] M. Kofler, "Swift 4: das umfassende Handbuch," Rheinwerk Verlag, 2018.
- [29] "Stack Overflow - Where Developers Learn, Share, & Build Careers." [Online]. Available: <https://stackoverflow.com/>. [Accessed: 27-Oct-2018].
- [30] "UIImageView pinch zoom swift - Stack Overflow." [Online]. Available: <https://stackoverflow.com/questions/30014241/uiimageView-pinch-zoom-swift>. [Accessed: 08-Dec-2018].
- [31] M. Kofler, "Swift 4: das umfassende Handbuch," Rheinwerk Verlag, 2018, p. 1080.
- [32] M. Kofler, "Swift 4: das umfassende Handbuch," Rheinwerk Verlag, 2018, p. 1086 und 1087.
- [33] M. Kofler, "Swift 4: das umfassende Handbuch," Rheinwerk Verlag, 2018, p. 1097.
- [34] "Schnell wie ein Pfeil - Papierflieger basteln - In nur 6 Schritten falten!" [Online]. Available: <https://einfach-basteln.com/papierflieger-basteln-der-pfeil/>. [Accessed: 15-Nov-2018].
- [35] "Gleiter Papierflieger basteln - In nur 5 Schritten zum perfekten Flieger." [Online]. Available: <https://einfach-basteln.com/papierflieger-basteln-gleiter/>. [Accessed: 15-Nov-2018].
- [36] "Weltrekord Papierflieger basteln 6 Schritte zum besten Papierflugzeug!" [Online]. Available: <https://einfach-basteln.com/weltrekord/>. [Accessed: 15-Nov-2018].
- [37] "Ausgereifter Kunstflieger Papierflieger basteln mit Anleitung." [Online]. Available: <https://einfach-basteln.com/kunstflieger/>. [Accessed: 15-Nov-2018].

## 11. Abbildungsverzeichnis

Abbildung 1: Obergeschoss des Polizeimuseums Hamburg mit Kriminaltechnik .....	11
Abbildung 2: Rosina Weiß vor dem Polizeimuseum Hamburg .....	11
Abbildung 3: KRIMI total - Zum Geier mit dem Mord.....	12
Abbildung 4: Exitgame - The Secret Lab von Kosmos .....	13
Abbildung 6: Screenshot der Website krimisspass.ch.....	13
Abbildung 7: Screenshot von der Website geocaching.com .....	15
Abbildung 5: Logo Foxtrail.....	15
Abbildung 8: Verschlüsselte Seite 1.....	21
Abbildung 9: Verschlüsselte Seite 2.....	21
Abbildung 10: Verschlüsselte Seite 3.....	21
Abbildung 11: Entschlüsselte Seite 1 .....	21
Abbildung 12: Entschlüsselte Seite 2 .....	21
Abbildung 13: Entschlüsselte Seite 3 .....	21
Abbildung 14: Foto von Emma Gutenberg .....	22
Abbildung 15: Widmung auf der Rückseite des Fotos.....	22
Abbildung 16: Emma Gutenberg (Ehefrau).....	23
Abbildung 17: Oliver Gutenberg (Bruder) .....	23
Abbildung 18: Emma Herzog (Arbeitskollegin).....	24
Abbildung 19: Hendrik Quedens (Chef).....	24
Abbildung 20: Firmenlogo Techgeniosus .....	24
Abbildung 21: Die Leiche .....	25
Abbildung 22: Der Oberkörper.....	25
Abbildung 23: Der Kopf.....	25
Abbildung 24: Die Schusswunde.....	25
Abbildung 25: Der linke Fuss mit Schuh.....	26
Abbildung 26: Der rechte Fuss ohne Schuh .....	26
Abbildung 27: Das Notizbuch.....	26
Abbildung 28: Der Leichenfundort .....	26
Abbildung 29: Zettel mit winziger Schrift .....	27
Abbildung 30: Teamworkaufgabe mit einem Stock.....	27
Abbildung 31: Grafik zur Tatzeitpunktbestimmung.....	29
Abbildung 32: Tabelle zur Tatzeitpunktbestimmung .....	29
Abbildung 33: Schuhabdruck in Beton .....	30
Abbildung 34: Tabelle zum Schuhabdruck.....	30
Abbildung 35: Verschmierter Blutabdruck einer linken Hand.....	31
Abbildung 36: Tatort .....	31
Abbildung 37: Grafik Tatort .....	31
Abbildung 38: Route meiner App.....	34
Abbildung 39: App-Icon.....	37
Abbildung 40: Versteck 1 bei Posten 2 .....	39
Abbildung 41: Versteck 2 bei Posten 4 .....	39
Abbildung 42: Versteck 3 bei Posten 4 .....	39
Abbildung 43: Versteck 4 bei Posten 6 .....	39
Abbildung 44: Karte im Sihlwald mit eingezeichneten Verstecken .....	40

---

Abbildung 45: 7. rechter Schuh .....	42
Abbildung 46: rechter Schuh mit Umgebung .....	42
Abbildung 47: Verhaftungsfoto Amelie Herzog .....	42
Abbildung 48: Verhaftungsfoto Emma Gutenberg .....	43
Abbildung 49: Verhaftungsfoto Oliver Gutenberg .....	43
Abbildung 50: Verhaftungsfoto Hendrik Quedens .....	44
Abbildung 51: Icon von Swift.....	45
Abbildung 52: Icon von Xcode.....	45
Abbildung 53: Screenshot Xcode Programmoberfläche im main.storyboard .....	47
Abbildung 54: Rosinas Rosen-App .....	48
Abbildung 55: Ausschnitt aus UINavigationController zur Richtungsberechnung ..	51

### Quellen der Abbildungen

Abbildung 1	<a href="https://www.polizeimuseum.hamburg/neuer-inhalt/">https://www.polizeimuseum.hamburg/neuer-inhalt/</a> (16.12.18)
Abbildung 2	Eigenes Foto
Abbildung 3	<a href="https://www.milan-spiele.de/images/kto012.jpg">https://www.milan-spiele.de/images/kto012.jpg</a> (14.12.18)
Abbildung 4	<a href="https://stoysnetcdn.com/tgtg/tgtg_ho17_692742/tgtg_ho17_692742.jpg">https://stoysnetcdn.com/tgtg/tgtg_ho17_692742/tgtg_ho17_692742.jpg</a> (12.12.18)
Abbildung 7	<a href="https://duckduckgo.com/?q=foxtral+&amp;t=ffsb&amp;iax=images&amp;ia=images&amp;iai=https%3A%2F%2Fwww.migros.ch%2Fdam%2Fjcr%3Aa3141f59-5c0f-47b6-93c7660ab431ae2%2F4_3_gross_FFFFFFFF_Foxtrail.jpg">https://duckduckgo.com/?q=foxtral+&amp;t=ffsb&amp;iax=images&amp;ia=images&amp;iai=https%3A%2F%2Fwww.migros.ch%2Fdam%2Fjcr%3Aa3141f59-5c0f-47b6-93c7660ab431ae2%2F4_3_gross_FFFFFFFF_Foxtrail.jpg</a> (12.12.18)
Abbildung 5	<a href="https://krimispass.ch/swiss/">https://krimispass.ch/swiss/</a> (25.11.18)
Abbildung 6	<a href="https://www.geocaching.com/play">https://www.geocaching.com/play</a> (15.12.18)
Abbildung 8 - 19	Eigene Fotos
Abbildung 20	Logo selbst erstellt
Abbildung 21 - 29	Eigene Fotos
Abbildung 30	M. Kramer, "Naturphänomene : im Spiel die Physik erforschen," Aulis-Verl. Deubner, 2007, p. 76.
Abbildung 31 - 50	Eigene Fotos
Abbildung 51	<a href="http://www.tosketch.com/wp-content/uploads/tosketch-86-Swift-Icon.png">http://www.tosketch.com/wp-content/uploads/tosketch-86-Swift-Icon.png</a> (25.11.18)
Abbildung 52	<a href="https://shazronatadobe.files.wordpress.com/2011/03/apple-xcode-icon.png">https://shazronatadobe.files.wordpress.com/2011/03/apple-xcode-icon.png</a> (25.11.18)
Abbildung 53/54	Eigene Fotos

## 12. Authentizitätserklärung

Hiermit bestätige ich, dass meine Maturitätsarbeit „Das Notizbuch – Eine physikalische Krimi-App im Sihlwald“ von mir persönlich und ohne fremde Hilfe verfasst wurde. Die verwendeten Quellen und Hilfsmittel wurden wahrheitsgetreu verwendet und sind im Quellenverzeichnis angegeben. Zudem entspricht meine Arbeit den gültigen Weisungen für die Maturitätsarbeit der Kantonsschule Wiedikon und wurde in selber Form noch nie zuvor veröffentlicht.

Langnau am Albis, Dezember 2018

A handwritten signature in blue ink that reads "Rosina Weiß". The signature is written in a cursive style with a large initial 'R'.



## 13. Anhang

### 13.1. Programmcode

Auf den folgenden Seiten befindet sich der gesamte Programmcode meiner Krimi-App. Zuerst wird der ViewController des Übersichtsscreens aufgeführt, darauf folgt der NotizbuchViewController. Nach den GlobalFuctions und den GlobalVariables folgt das AppDelegate. Dann kommen die verschiedenen Files zur Navigation: KartenViewController, Position, ArrowView und NavigationViewController. Schliesslich werden alle ViewController der Posten von 1 bis 10 sortiert gelistet. Da ich den Posten 8 streichen musste, gibt es eine Lücke zwischen 7 und 9, welche ignoriert werden kann.

---

## ViewController

```
//
// ViewController.swift
// Maturitätsapp Rosina
//
// Created by Rosina Weiß on 08.10.18.
// Copyright © 2018 Rosina Weiß. All rights reserved.
//

import UIKit
import CoreLocation

class ViewController: UIViewController {
    let locmgr = CLLocationManager()

    @IBOutlet var buttonss: [UIButton]!
    @IBOutlet weak var button1: UIButton!

    override func viewDidLoad() {
        super.viewDidLoad()

        button1.backgroundColor = rosGreen2
        // Do any additional setup after loading the view,
        typically from a nib.

        //um Erlaubnis fragen, ob die Ortungsdienste verwendet
        werden dürfen, kommt nur beim 1. Mal
        locmgr.requestWhenInUseAuthorization()
        let defaults = UserDefaults.standard
        print(buttonss)

        for i in 1...buttonss.count - 1 {
            let key = "Posten\(i+1)"
            let unlocked = defaults.bool(forKey: key)
            buttonss[i].isEnabled = unlocked
            if unlocked {
                buttonss[i].backgroundColor = rosGreen2
            } else {
                buttonss[i].backgroundColor = rosGrey1
            }
        }
    }
}
```

## NotizenViewController

```
//
// NotizenViewController.swift
// Maturitätsapp Rosina
//
// Created by Rosina Weiß on 18.10.18.
// Copyright © 2018 Rosina Weiß. All rights reserved.
//

import UIKit

struct ImageObject {
    let page: Int
    let images: [UIImage]
    let unlockedAt: [Float]
}

class NotizenViewController: UIViewController,
    UIScrollViewDelegate {

    @IBOutlet weak var imageView: UIImageView!
    @IBOutlet weak var nextButton: UIButton!
    @IBOutlet weak var backButton: UIButton!
    @IBOutlet weak var scrollView: UIScrollView!
    @IBOutlet weak var pagelabel: UILabel!

    var page = 1
    let pageCount = 30
    var currentState: Float = 0

    var images: [ImageObject] = []

    override func viewDidLoad() {
        super.viewDidLoad()

        pagelabel.text = String(page)

        //Zoom fürs Bild
        scrollView.minimumZoomScale = 1.0
        scrollView.maximumZoomScale = 6.0
        scrollView.delegate = self

        images = [
            ImageObject(page: 1, images: [UIImage(named:
                "y_28")!], unlockedAt: [1.2]),
            ImageObject(page: 2, images: [UIImage(named: "1.
                Leiche")!], unlockedAt: [1.41]),
            ImageObject(page: 3, images: [UIImage(named: "2, 3 &
                6 Überblick")!], unlockedAt: [1.42]),
```

---

```
ImageObject(page: 4, images: [UIImage(named: "2.
Kopf")!], unlockedAt: [1.43]),
ImageObject(page: 5, images: [UIImage(named: "3.
Schusswunde")!], unlockedAt: [1.44]),
ImageObject(page: 6, images: [UIImage(named: "4. Fuss
mit Schuh")!], unlockedAt: [1.45]),
ImageObject(page: 7, images: [UIImage(named: "5. Fuss
ohne Schuh")!], unlockedAt: [1.46]),
ImageObject(page: 8, images: [UIImage(named: "6.
Notizbuch")!], unlockedAt: [1.47]),
ImageObject(page: 9, images: [UIImage(named: "Tatort
Umgebung")!], unlockedAt: [1.48]),
ImageObject(page: 10, images: [UIImage(named:
"Personalakte Julian Gutenberg")!,UIImage(named:
"Winzige Schrift")!,UIImage(named: "Winzige Schrift
gross")!], unlockedAt: [1.49,2.02,2.5]),
ImageObject(page: 11, images: [UIImage(named:
"Ehefrau schön")!], unlockedAt: [2.03]),
ImageObject(page: 12, images: [UIImage(named:
"Widmung Version 1")!], unlockedAt: [2.04]),
ImageObject(page: 13, images: [UIImage(named:
"Personalakte Julian Gutenberg")!,UIImage(named:
"Personalakte Julian Gutenberg 2")!], unlockedAt:
[2.01,6.81]),
ImageObject(page: 14, images: [UIImage(named:
"Personalakte Amelie Herzog")!,UIImage(named:
"Personalakte Amelie Herzog 2")!,UIImage(named:
"Personalakte Amelie Herzog 3")!,UIImage(named:
"Personalakte Amelie Herzog 4")!,UIImage(named:
"Personalakte Amelie Herzog 5")!], unlockedAt:
[3.71,4.42,5.51,6.82,7.31]),
ImageObject(page: 15, images: [UIImage(named:
"Personalakte Emma Gutenberg")!,UIImage(named:
"Personalakte Emma Gutenberg 2")!,UIImage(named:
"Personalakte Emma Gutenberg 3")!,UIImage(named:
"Personalakte Emma Gutenberg 4")!], unlockedAt:
[3.72,5.52,7.32,7.51]),
ImageObject(page: 16, images: [UIImage(named:
"Personalakte Oliver Gutenberg")!,UIImage(named:
"Personalakte Oliver Gutenberg 2")!,UIImage(named:
"Personalakte Oliver Gutenberg 3")!,UIImage(named:
"Personalakte Oliver Gutenberg 4")!], unlockedAt:
[3.73,5.53,6.83,7.33]),
```

```

ImageObject(page: 17, images: [UIImage(named:
  "Personalakte Hendrik Quedens")!,UIImage(named:
  "Personalakte Hendrik Quedens 2")!,UIImage(named:
  "Personalakte Hendrik Quedens 3")!,UIImage(named:
  "Personalakte Hendrik Quedens 4")!], unlockedAt:
  [3.74,5.54,6.84,7.34]),
ImageObject(page: 18, images: [UIImage(named:
  "Kaufvertrag QC")!], unlockedAt: [4.41]),
ImageObject(page: 19, images: [UIImage(named:
  "Verschlüsselung Zahlen")!], unlockedAt: [6.31]),
ImageObject(page: 20, images: [UIImage(named:
  "Verschlüsselung Seite 1")!], unlockedAt: [6.32]),
ImageObject(page: 21, images: [UIImage(named:
  "Verschlüsselung Seite 2")!,UIImage(named:
  "Entschlüsselung 1")!], unlockedAt: [6.33,6.85]),
ImageObject(page: 22, images: [UIImage(named:
  "Verschlüsselung Seite 3")!,UIImage(named:
  "Verschlüsselung Seite 2")!], unlockedAt:
  [6.34,6.86]),
ImageObject(page: 23, images: [UIImage(named:
  "Entschlüsselung 2")!], unlockedAt: [6.87]),
ImageObject(page: 24, images: [UIImage(named:
  "Verschlüsselung Seite 3")!], unlockedAt: [6.88]),
ImageObject(page: 25, images: [UIImage(named:
  "Entschlüsselung 3")!,], unlockedAt: [6.89]),
ImageObject(page: 26, images: [UIImage(named:
  "Fussabdruck")!], unlockedAt: [7.1]),
ImageObject(page: 27, images: [UIImage(named: "7.
  Schuh Umgebung")!], unlockedAt: [7.41]),
ImageObject(page: 28, images: [UIImage(named: "7.
  Schuh")!], unlockedAt: [7.42]),
ImageObject(page: 29, images: [UIImage(named: "8.
  verschmierte Hand nah")!], unlockedAt: [9.01]),
ImageObject(page: 30, images: [UIImage(named: "8.
  verschmierte Hand Umgebung")!], unlockedAt: [9.02])
]

page = UserDefaults.standard.integer(forKey: "notesPage")
if page == 0 {page = 1} // Zeigt die zuletzt besuchte
  Seite an
currentState = UserDefaults.standard.float(forKey:
  "notes")
// Do any additional setup after loading the view.
showImage(for: page)
}

func showImage(for page: Int) {
  pagelabel.text = String(page)
}

```

---

```

var found = false
//Sucht nach dem Bild mit Seitennummer "page"
for img in images {
    if img.page == page {
        //Zählt anzahl Bildversionen für diese Seite
        let count = img.images.count
        for i in 0...count - 1 {
            if img.unlockedAt[i] <= currentState {
                found = true
                imageView.image = img.images[i]
            }
        }
    }
    if !found {
        imageView.image = UIImage(named: "Nicht
            verfügbar")
    }
}

//Verstecke Randknöpfe
if page > 1 {
    backButton.isHidden = false
} else {
    backButton.isHidden = true
}
if page < pageCount {
    nextButton.isHidden = false
} else {
    nextButton.isHidden = true
}
}

//Für den Zoom
func viewForZooming(in scrollView: UIScrollView) -> UIView? {

    return imageView
}

@IBAction func back(_ sender: Any) {
    if page > 1 {
        page -= 1
        showImage(for: page)
    }
}

@IBAction func next(_ sender: Any) {
    if page < pageCount {

```

```
        page += 1
        showImage(for: page)
    }
}

override func viewWillAppear(_ animated: Bool) {
    super.viewWillAppear(animated)
    UserDefaults.standard.set(page, forKey: "notesPage")
}
}
```

---

## GlobalFuctions

```
//
// GlobalFuctions.swift
// Maturitätsapp Rosina
//
// Created by Rosina Weiß on 11.10.18.
// Copyright © 2018 Rosina Weiß. All rights reserved.
//

import Foundation
import UIKit
func degreesMinutes(_ x:Double) -> String {
    let remainder = x.truncatingRemainder(dividingBy: 1)
    let minutes = abs(remainder * 60)
    let remainder2 = minutes.truncatingRemainder(dividingBy: 1)
    let seconds = abs(remainder2 * 60)
    return "\(Int(x))° \((Int(minutes))' \((Int(seconds)))'"
}

func degreesDecimal(deg: Double, min: Double, sec: Double) ->
Double {
    var result: Double = deg
    result += min/60
    result += sec/3600
    return result
}
// diese Funktion wandelt die Koordinaten schön um und zeigt
// diese richtig an. Auch so, dass es keine zwei Minus entstehen
// oder ähnliches.

func randomCompliment() -> String {
    let rand = Int.random(in: 0...7)
    var compliment = ""
    switch rand {
    case 0: compliment = "Tatsächlich!"
    case 1: compliment = "Wunderbar!"
    case 2: compliment = "Das ist korrekt!"
    case 3: compliment = "Richtig!"
    case 4: compliment = "Weiter so!"
    case 5: compliment = "Du Schlaumeier!"
    case 6: compliment = "Perfekt!"
    case 7: compliment = "Super!"
    case 8: compliment = "Ausgezeichnet!"
    default: compliment = "Korrekt!"
    }
    return compliment
}
```

```
// diese Funktion generiert eine zufälligen
Complimentemacher. Wenn man ein Quiz richtig gelöst hat,
bekommt man eine Nachricht, dass man die Aufgabe geschafft
hat. Damit dort nicht immer der gleiche text im Titel steht,
wird per Zufall ein Kompliment Ausgewählt
}

func randomWrong() -> String {
    let rand = Int.random(in: 0...4)
    var compliment = ""
    switch rand {
    case 0: compliment = "Leider Falsch"
    case 1: compliment = "Nicht ganz"
    case 2: compliment = "Knapp daneben"
    case 3: compliment = "Nicht korrekt"
    case 4: compliment = "Fast richtig"
    default: compliment = "Falsch!"
    }
    return compliment
    //Gegenteil von der ComplimentFunktion, diese sagt auf
    verschiedene Weisen, dass man etwas falsch gemacht hat.
}

func correctAlert(vc: UIViewController, why: String) {
    let alert = UIAlertController(title: randomCompliment(),
    message: why, preferredStyle: .alert)
    alert.addAction(UIAlertAction(title: "Weiter",
    style: .default, handler: nil))
    vc.present(alert,animated: true)
}

func wrongAlert(vc: UIViewController) {
    let alert = UIAlertController(title: randomWrong(), message:
    "Versuch es doch nochmals :)", preferredStyle: .alert)
    alert.addAction(UIAlertAction(title: "Ok", style: .default,
    handler: nil))
    vc.present(alert,animated: true)
}

func setNotes(notes: Float) {
    let defaults = UserDefaults.standard
    if defaults.float(forKey: "notes") < notes {
        defaults.set(notes, forKey: "notes")
    }
}
```

---

## GlobalVariables

```
//  
// GlobalVariables.swift  
// Maturitätsapp Rosina  
//  
// Created by Rosina Weiß on 18.10.18.  
// Copyright © 2018 Rosina Weiß. All rights reserved.  
//  
  
import Foundation  
import UIKit  
  
var murderer = false  
let murdererCorrect = true  
  
var qcplace = false  
let qcCorrect = true  
  
// Verschiedene Farben für meine App  
let rosGreen1 = UIColor(displayP3Red: 82/255, green: 142/255,  
    blue: 51/255, alpha: 1)  
let rosGreen2 = UIColor(displayP3Red: 103/255, green: 164/255,  
    blue: 23/255, alpha: 0.51)  
let rosGreen3 = UIColor(displayP3Red: 82/255, green: 142/255,  
    blue: 51/255, alpha: 0.78)  
  
let rosGrey1 = UIColor(displayP3Red: 146/255, green: 146/255,  
    blue: 146/255, alpha: 0.78)  
let rosGrey2 = UIColor(displayP3Red: 146/255, green: 146/255,  
    blue: 146/255, alpha: 0.5)  
  
let rosDefault = UIColor(displayP3Red: 0, green: 0, blue: 0,  
    alpha: 0)
```

## AppDelegate

```
//
// AppDelegate.swift
// Maturitätsapp Rosina
//
// Created by Rosina Weiß on 08.10.18.
// Copyright © 2018 Rosina Weiß. All rights reserved.
//

import UIKit

@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {

    var window: UIWindow?

    func application(_ application: UIApplication,
        didFinishLaunchingWithOptions launchOptions:
        [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
        // Override point for customization after application
        launch.
        return true
    }

    func applicationWillResignActive(_ application:
        UIApplication) {
        // Sent when the application is about to move from active
        to inactive state. This can occur for certain types of
        temporary interruptions (such as an incoming phone call
        or SMS message) or when the user quits the application
        and it begins the transition to the background state.
        // Use this method to pause ongoing tasks, disable
        timers, and invalidate graphics rendering callbacks.
        Games should use this method to pause the game.
    }

    func applicationDidEnterBackground(_ application:
        UIApplication) {
        // Use this method to release shared resources, save user
        data, invalidate timers, and store enough application
        state information to restore your application to its
        current state in case it is terminated later.
        // If your application supports background execution,
        this method is called instead of
        applicationWillTerminate: when the user quits.
    }
}
```

---

```
func applicationWillEnterForeground(_ application:
UIApplication) {
    // Called as part of the transition from the background
    to the active state; here you can undo many of the
    changes made on entering the background.
}

func applicationDidBecomeActive(_ application: UIApplication)
{
    // Restart any tasks that were paused (or not yet
    started) while the application was inactive. If the
    application was previously in the background, optionally
    refresh the user interface.
}

func applicationWillTerminate(_ application: UIApplication) {
    // Called when the application is about to terminate.
    Save data if appropriate. See also
    applicationDidEnterBackground:.
}
}
```

## KartenViewController

```
//
// KartenViewController.swift
// Maturitätsapp Rosina
//
// Created by Rosina Weiß on 14.12.18.
// Copyright © 2018 Rosina Weiß. All rights reserved.
//

import UIKit

class KartenViewController: UIViewController,
    UIScrollViewDelegate {

    @IBOutlet weak var scrollView: UIScrollView!
    @IBOutlet weak var imageView: UIImageView!

    override func viewDidLoad() {
        super.viewDidLoad()
        //Zoom fürs Bild
        scrollView.minimumZoomScale = 1.0
        scrollView.maximumZoomScale = 6.0
        scrollView.delegate = self
    }

    //Für den Zoom
    func viewForZooming(in scrollView: UIScrollView) -> UIView? {

        return imageView
    }
}
```

---

## Position

```
//  
// Position.swift  
// Maturitätsapp Rosina  
//  
// Created by Rosina Weiß on 11.10.18.  
// Copyright © 2018 Rosina Weiß. All rights reserved.  
//  
  
import Foundation  
  
struct Position: Codable, CustomStringConvertible {  
    var name: String // Name des Ortes  
    var lat: Double // geogr. Breite  
    var long: Double // geogr. Länge  
  
    var description: String {  
        return name + "\(lat) \(long)"  
        // unkomplizierte Ausgabe von Elementen durch  
        print(posvar)  
    }  
  
    init(name: String, lat: Double, long: Double)  
    {  
        self.name = name  
        self.lat = lat  
        self.long = long  
    }  
}
```

## ArrowView

```

//
// ArrowView.swift
// Maturitätsapp Rosina
//
// Created by Rosina Weiß on 11.10.18.
// Copyright © 2018 Rosina Weiß. All rights reserved.
//

import Foundation
import UIKit
@IBDesignable class ArrowView: UIView {
    @IBInspectable var color: UIColor = .black
    //Farbe des Pfeils
    @IBInspectable var heading: Double = 0.0 { //Richtung des
        Pfeils
            didSet { //bei Änderungen neu zeichnen
                setNeedsDisplay()
            }
        }
    //Methode, um den Inhalt der View zu zeichnen
    override func draw(_ rect: CGRect) {
        let rad = heading / 180.0 * .pi;
        let side = min(frame.size.width, frame.size.height)
        let side2 = side/2 //halbe Seitenlänge
        let radius1 = side2 * 0.95 //Radius aussen
        let radius2 = side2 * 0.25 //Radius innen
        let x0 = frame.size.width / 2 //Mittelpunkt
        let y0 = frame.size.height / 2
        let x1 = x0 + radius1 * CGFloat(cos(rad)) //Spitze
        let y1 = y0 - radius1 * CGFloat(sin(rad))
        let x2 = x0 + radius1 * CGFloat(cos(rad + .pi * 0.8))
        let y2 = y0 - radius1 * CGFloat(sin(rad + .pi * 0.8))
        let x3 = x0 + radius2 * CGFloat(cos(rad + .pi)) //unten
        let y3 = y0 - radius2 * CGFloat(sin(rad + .pi))
        let x4 = x0 + radius1 * CGFloat(cos(rad + .pi * 1.2))
        let y4 = y0 - radius1 * CGFloat(sin(rad + .pi * 1.2))
        let myBezier = UIBezierPath()
        myBezier.move(to: CGPoint(x: x1, y: y1))
        myBezier.addLine(to: CGPoint(x: x2, y: y2))
        myBezier.addLine(to: CGPoint(x: x3, y: y3))
        myBezier.addLine(to: CGPoint(x: x4, y:y4))
        myBezier.close()
        color.setFill()
        myBezier.fill()
        color.setStroke()
        myBezier.stroke()
    }
}

```

---

## NavigationViewController

```
//
// NavigationViewController.swift
// Maturitätsapp Rosina
//
// Created by Rosina Weiß on 13.10.18.
// Copyright © 2018 Rosina Weiß. All rights reserved.
//

import UIKit
import CoreLocation

class NavigationViewController:
    UIViewController, CLLocationManagerDelegate {

    var pos: Position! //Positionsinstanz

    var heading = 0.0 //Richtung zum Ziel 0 - 360 Grad

    @IBOutlet weak var postenLabel: UILabel!
    @IBOutlet weak var entfernungsLabel: UILabel!
    @IBOutlet weak var koordinatenLabel: UILabel!
    @IBOutlet weak var arrowView: ArrowView!
    @IBOutlet weak var actualPosition: UILabel!
    @IBOutlet var UIView: UIView!

    var mylocmgr = CLLocationManager()

    override func viewDidLoad() {

        super.viewDidLoad()

        mylocmgr.requestWhenInUseAuthorization()

        if CLLocationManager.locationServicesEnabled() {

            mylocmgr.delegate = self
            mylocmgr.desiredAccuracy = kCLLocationAccuracyBest
            mylocmgr.startUpdatingLocation()
            mylocmgr.startUpdatingHeading()
        }

        if pos == nil { return }

        //Steuerelement mit Datrn aus "pos" initialisieren
        let dfmt = DateFormatter ()
        dfmt.dateStyle = .medium
    }
}
```

```

        dfmt.timeStyle = .short
        postenLabel.text = pos.name
    }

    func locationManager(_ manager: CLLocationManager,
        didUpdateLocations locations: [CLLocation]) {

        let long = pos.long
        let lat = pos.lat

        koordinatenLabel.text = " \((lat) N / \((long) O"
        //Entfernung zwischen "pos" und aktuellem Standort
        //errechnen
        let loc = CLLocation(latitude: pos.lat, longitude:
            pos.long)
        print(loc)
        let dist = manager.location!.distance(from: loc)
        entfernungsLabel.text = "→ " + String(format: "%.0f m",
            dist)
        actualPosition.text = "\((manager.location!)" + " ←"

        if dist <= 15 {
            UIView.backgroundColor = rosGreen3
            arrowView.backgroundColor = rosGreen3
        }

        //Richtung vom Standort zum Ziel berechnen
        let rad = atan2((pos.long -
            manager.location!.coordinate.longitude)*3.75, (pos.lat -
            manager.location!.coordinate.latitude)*5.5)
        print(heading)
        heading = rad * 180 / .pi
    }

    func locationManager(_ manager: CLLocationManager,
        didUpdateHeading newHeading: CLHeading) {
        //richtung von der aktuellemn Richtung zum Ziel
        //ausrechnen
        arrowView.heading = mylocmgr.heading!.trueHeading -
            heading + 90
    }
}

```

---

## ViewController1\_0

```
//  
// ViewController1_0.swift  
// Maturitätsapp Rosina  
//  
// Created by Rosina Weiß on 14.10.18.  
// Copyright © 2018 Rosina Weiß. All rights reserved.  
//  
  
import UIKit  
  
class ViewController1_0: UIViewController {  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
    }  
  
}
```

## ViewController1\_1

```
//  
// ViewController1_1.swift  
// Maturitätsapp Rosina  
//  
// Created by Rosina Weiß on 16.10.18.  
// Copyright © 2018 Rosina Weiß. All rights reserved.  
//  
  
import UIKit  
  
class ViewController1_1: UIViewController {  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
  
    }  
  
}
```

---

## ViewController1\_2

```
//
// ViewController1_2.swift
// Maturitätsapp Rosina
//
// Created by Rosina Weiß on 28.10.18.
// Copyright © 2018 Rosina Weiß. All rights reserved.
//

import UIKit

class ViewController1_2: UIViewController, UITextFieldDelegate {

    @IBOutlet weak var textField: UITextField!
    @IBOutlet weak var nextButton: UIButton!
    @IBOutlet weak var überprüfenButton: UIButton!

    var pos: Position!

    override func viewDidLoad() {
        super.viewDidLoad()

        textField.delegate = self

        //Standortpositionen
        let pos1 = Position(name: "Brückenziffer", lat:
            47.27127972, long: 8.55652470)
        pos = pos1

        let quizSolved = UserDefaults.standard.bool(forKey:
            "Quiz1")

        nextButton.isEnabled = quizSolved

        textField.backgroundColor = rosGreen2
        überprüfenButton.backgroundColor = rosGrey2

        //Schalte Notitz Frei
        setNotes(notes: 1.2)
    }

    func textFieldShouldReturn(_ textField: UITextField) -> Bool
    {
        textField.resignFirstResponder()
        return true
        //macht, dass das Keyboard verschwindet, wenn man auf
        //return drückt.
    }
}
```

```
@IBAction func ueberpruefenButton(_ sender: UITextField) {
    if textField.text == "28" {
        correctAlert(vc: self, why: "Die Zahl 28 stimmt
            absolut! Sie wird nun in deinem Notizbuch
            abgespeichert, da sie später vielleicht noch eine
            wichtige Rolle spielen könnte.")
        nextButton.isEnabled = true
        UserDefaults.standard.set(true, forKey: "Quiz1")
        sender.backgroundColor = rosGrey1
    } else {
        wrongAlert(vc: self)
    }
}
override func prepare(for segue: UIStoryboardSegue, sender:
Any?) {
    if let dest = segue.destination as?
        UINavigationController {
        dest.pos = pos
    }
}
}
```

---

## ViewController1\_3

```
//  
// ViewController1_3.swift  
// Maturitätsapp Rosina  
//  
// Created by Rosina Weiß on 04.12.18.  
// Copyright © 2018 Rosina Weiß. All rights reserved.  
//  
  
import UIKit  
  
class ViewController1_3: UIViewController {  
  
    var pos: Position!  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
  
        //Standortpositionen  
        let pos2 = Position(name: "Leichenfundort", lat:  
            47.27039417 , long: 8.55732123)  
        pos = pos2  
  
    }  
  
    override func prepare(for segue: UIStoryboardSegue, sender:  
        Any?) {  
  
        if let dest = segue.destination as?  
            UINavigationController {  
            dest.pos = pos  
        }  
    }  
  
}
```

## ViewController1\_4

```
//  
// ViewController1_4.swift  
// Maturitätsapp Rosina  
//  
// Created by Rosina Weiß on 28.10.18.  
// Copyright © 2018 Rosina Weiß. All rights reserved.  
//  
  
import UIKit  
  
class ViewController1_4: UIViewController {  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
  
        //Schalte Notitz Frei  
        setNotes(notes: 1.41)  
        setNotes(notes: 1.42)  
        setNotes(notes: 1.43)  
        setNotes(notes: 1.44)  
        setNotes(notes: 1.45)  
        setNotes(notes: 1.46)  
        setNotes(notes: 1.47)  
        setNotes(notes: 1.48)  
        setNotes(notes: 1.49)  
    }  
}
```

---

## ViewController1\_5

```
//  
// ViewController1_5.swift  
// Maturitätsapp Rosina  
//  
// Created by Rosina Weiß on 28.10.18.  
// Copyright © 2018 Rosina Weiß. All rights reserved.  
//  
  
import UIKit  
  
class ViewController1_5: UIViewController {  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
  
        // Schaltet den nächsten Posten im Info View Controller  
        frei  
        UserDefaults.standard.set(true, forKey: "Posten2")  
    }  
  
}
```

## ViewController2\_0

```
//  
// ViewController2_0.swift  
// Maturitätsapp Rosina  
//  
// Created by Rosina Weiß on 15.10.18.  
// Copyright © 2018 Rosina Weiß. All rights reserved.  
//  
  
import UIKit  
  
class ViewController2_0: UIViewController {  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
  
        setNotes(notes: 2.01)  
        setNotes(notes: 2.02)  
        setNotes(notes: 2.03)  
        setNotes(notes: 2.04)  
  
    }  
  
}
```

---

## ViewController2\_1

```
//  
// ViewController2_1.swift  
// Maturitätsapp Rosina  
//  
// Created by Rosina Weiß on 29.11.18.  
// Copyright © 2018 Rosina Weiß. All rights reserved.  
//  
  
import UIKit  
  
class ViewController2_1: UIViewController {  
  
    var pos: Position!  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
  
        //Standortpositionen  
        let pos3 = Position(name: "Versteck Lupen", lat:  
            47.27077919, long: 8.55734462)  
        pos = pos3  
  
        //Schalte Notitz 1.0 Frei  
        setNotes(notes: 2.0)  
  
    }  
  
    override func prepare(for segue: UIStoryboardSegue, sender:  
        Any?) {  
  
        if let dest = segue.destination as?  
            UINavigationController {  
            dest.pos = pos  
        }  
    }  
}
```

## ViewController2\_2

```
//
// ViewController2_2.swift
// Maturitätsapp Rosina
//
// Created by Rosina Weiß on 29.11.18.
// Copyright © 2018 Rosina Weiß. All rights reserved.
//

import UIKit

class ViewController2_2: UIViewController {

    @IBOutlet weak var button1: UIButton!
    @IBOutlet weak var button2: UIButton!
    @IBOutlet weak var button3: UIButton!
    @IBOutlet weak var überprüfenButton: UIButton!
    @IBOutlet weak var nextButton: UIButton!

    var pressedButtons = [false, false, false]
    let correct = [true, false, false]

    override func viewDidLoad() {
        super.viewDidLoad()

        let quizSolved = UserDefaults.standard.bool(forKey:
            "Quiz2.2")
        nextButton.isEnabled = quizSolved

        button1.backgroundColor = rosGreen2
        button2.backgroundColor = rosGreen2
        button3.backgroundColor = rosGreen2
        überprüfenButton.backgroundColor = rosGrey2
    }
    @IBAction func action1(_ sender: UIButton) {
        if sender.backgroundColor != rosGreen3 {
            sender.backgroundColor = rosGreen3
            pressedButtons[0] = true
        } else {
            sender.backgroundColor = rosGreen2
            pressedButtons[0] = false
        }
    }
    @IBAction func aciton2(_ sender: UIButton) {
        if sender.backgroundColor != rosGreen3 {
            sender.backgroundColor = rosGreen3
            pressedButtons[1] = true
        } else {
```



## ViewController2\_3

```
//
// ViewController2_3.swift
// Maturitätsapp Rosina
//
// Created by Rosina Weiß on 29.11.18.
// Copyright © 2018 Rosina Weiß. All rights reserved.
//

import UIKit

class ViewController2_3: UIViewController {

    @IBOutlet weak var button1: UIButton!
    @IBOutlet weak var button2: UIButton!
    @IBOutlet weak var button3: UIButton!
    @IBOutlet weak var überprüfenButton: UIButton!
    @IBOutlet weak var nextButton: UIButton!

    var pressedButtons = [false, false, false]
    let correct = [false, true, false]

    override func viewDidLoad() {
        super.viewDidLoad()

        let quizSolved = UserDefaults.standard.bool(forKey:
            "Quiz2.3")
        nextButton.isEnabled = quizSolved

        button1.backgroundColor = rosGreen2
        button2.backgroundColor = rosGreen2
        button3.backgroundColor = rosGreen2
        überprüfenButton.backgroundColor = rosGrey2
    }
    @IBAction func action1(_ sender: UIButton) {
        if sender.backgroundColor != rosGreen3 {
            sender.backgroundColor = rosGreen3
            pressedButtons[0] = true
        } else {
            sender.backgroundColor = rosGreen2
            pressedButtons[0] = false
        }
    }
    @IBAction func aciton2(_ sender: UIButton) {
        if sender.backgroundColor != rosGreen3 {
            sender.backgroundColor = rosGreen3
            pressedButtons[1] = true
        } else {
```



## ViewController2\_4

```
//  
// ViewController2_4.swift  
// Maturitätsapp Rosina  
//  
// Created by Rosina Weiß on 29.11.18.  
// Copyright © 2018 Rosina Weiß. All rights reserved.  
//  
  
import UIKit  
  
class ViewController2_4: UIViewController {  
  
    @IBOutlet weak var button1: UIButton!  
    @IBOutlet weak var button2: UIButton!  
    @IBOutlet weak var button3: UIButton!  
    @IBOutlet weak var überprüfenButton: UIButton!  
    @IBOutlet weak var nextButton: UIButton!  
  
    var pressedButtons = [false, false, false]  
    let correct = [false, false, true]  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
  
        let quizSolved = UserDefaults.standard.bool(forKey:  
            "Quiz2.4")  
        nextButton.isEnabled = quizSolved  
  
        button1.backgroundColor = rosGreen2  
        button2.backgroundColor = rosGreen2  
        button3.backgroundColor = rosGreen2  
        überprüfenButton.backgroundColor = rosGrey2  
  
    }  
    @IBAction func action1(_ sender: UIButton) {  
        if sender.backgroundColor != rosGreen3 {  
            sender.backgroundColor = rosGreen3  
            pressedButtons[0] = true  
        } else {  
            sender.backgroundColor = rosGreen2  
            pressedButtons[0] = false  
        }  
    }  
    @IBAction func aciton2(_ sender: UIButton) {  
        if sender.backgroundColor != rosGreen3 {  
            sender.backgroundColor = rosGreen3  
            pressedButtons[1] = true  
        } else {
```



## ViewController2\_5

```
//  
// ViewController2_5.swift  
// Maturitätsapp Rosina  
//  
// Created by Rosina Weiß on 29.11.18.  
// Copyright © 2018 Rosina Weiß. All rights reserved.  
//  
  
import UIKit  
  
class ViewController2_5: UIViewController {  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
  
        // Schaltet den nächsten Posten im Info View Controller  
        frei  
        UserDefaults.standard.set(true, forKey: "Posten3")  
  
        //Schalte Notitz Frei  
        setNotes(notes: 2.5)  
    }  
}
```

---

## ViewController3\_0

```
//  
// ViewController3_0.swift  
// Maturitätsapp Rosina  
//  
// Created by Rosina Weiß on 15.10.18.  
// Copyright © 2018 Rosina Weiß. All rights reserved.  
//  
  
import UIKit  
  
class ViewController3_0: UIViewController {  
  
    var pos: Position!  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
  
        //Standortpositionen  
        let pos4 = Position(name: "Gleichgewichtsstation", lat:  
            47.27168108, long: 8.55729642)  
        pos = pos4  
    }  
  
    override func prepare(for segue: UIStoryboardSegue, sender:  
        Any?) {  
  
        if let dest = segue.destination as?  
            UINavigationController {  
            dest.pos = pos  
        }  
    }  
}
```

## ViewController3\_1

```
//  
// ViewController3_1.swift  
// Maturitätsapp Rosina  
//  
// Created by Rosina Weiß on 29.11.18.  
// Copyright © 2018 Rosina Weiß. All rights reserved.  
//  
  
import UIKit  
  
class ViewController3_1: UIViewController {  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
  
    }  
  
    @IBAction func nextAction(_ sender: Any) {  
        showAlert(vc: self, why: "Diese Aufgabe ist nicht so  
        einfach, wie man denkt, da jeder Kontakt zum Stab  
        probiert zu halten. Deswegen hat der Stab eher eine  
        Tendenz nach oben zu wandern. Wenn man jedoch genau die  
        Gegenbewegung bewirken möchte muss man sich sehr gut  
        absprechen und gut zusammenarbeiten.")  
    }  
}
```

---

## ViewController3\_2

```
//  
// ViewController3_2.swift  
// Maturitätsapp Rosina  
//  
// Created by Rosina Weiß on 29.11.18.  
// Copyright © 2018 Rosina Weiß. All rights reserved.  
//  
  
import UIKit  
  
class ViewController3_2: UIViewController {  
  
    var pos: Position!  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
  
        //Standortpositionen  
        let pos5 = Position(name: "Schallexperiment", lat:  
            47.27128701, long: 8.55809086)  
        pos = pos5  
  
    }  
    override func prepare(for segue: UIStoryboardSegue, sender:  
        Any?) {  
  
        if let dest = segue.destination as?  
            UINavigationController {  
            dest.pos = pos  
        }  
    }  
}
```

## ViewController3\_3

```
//  
// ViewController3_3.swift  
// Maturitätsapp Rosina  
//  
// Created by Rosina Weiß on 29.11.18.  
// Copyright © 2018 Rosina Weiß. All rights reserved.  
//  
  
import UIKit  
  
class ViewController3_3: UIViewController {  
    override func viewDidLoad() {  
        super.viewDidLoad()  
    }  
}
```

---

## ViewController3\_4

```
//
// ViewController3_4.swift
// Maturitätsapp Rosina
//
// Created by Rosina Weiß on 29.11.18.
// Copyright © 2018 Rosina Weiß. All rights reserved.
//

import UIKit

class ViewController3_4: UIViewController {

    @IBOutlet weak var button1: UIButton!
    @IBOutlet weak var button2: UIButton!
    @IBOutlet weak var button3: UIButton!
    @IBOutlet weak var button4: UIButton!
    @IBOutlet weak var button5: UIButton!
    @IBOutlet weak var button6: UIButton!
    @IBOutlet weak var überprüfenButton: UIButton!
    @IBOutlet weak var nextButton: UIButton!

    var pressedButtons = [false, false, false, false, false, false]
    let correct = [false, false, true, false, false, true]

    override func viewDidLoad() {
        super.viewDidLoad()

        let quizSolved = UserDefaults.standard.bool(forKey:
            "Quiz3.4")
        nextButton.isEnabled = quizSolved

        button1.backgroundColor = rosGreen2
        button2.backgroundColor = rosGreen2
        button3.backgroundColor = rosGreen2
        button4.backgroundColor = rosGreen2
        button5.backgroundColor = rosGreen2
        button6.backgroundColor = rosGreen2
        überprüfenButton.backgroundColor = rosGrey2
    }

    @IBAction func action1(_ sender: UIButton) {
        if sender.backgroundColor != rosGreen3 {
            sender.backgroundColor = rosGreen3
            pressedButtons[0] = true
        } else {
            sender.backgroundColor = rosGreen2
            pressedButtons[0] = false
        }
    }
}
```

```
}
@IBAction func aciton2(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[1] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[1] = false
    }
}
@IBAction func action3(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[2] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[2] = false
    }
}
@IBAction func action4(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[3] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[3] = false
    }
}
@IBAction func action5(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[4] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[4] = false
    }
}
@IBAction func action6(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[5] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[5] = false
    }
}
@IBAction func überprüfenAciton(_ sender: Any) {
    if pressedButtons == correct{
```

---

```
correctAlert(vc: self, why: "Licht ist schneller als  
Schall, da Schall vom Luftwiderstand gebremst wird.  
Auch können unsere Ohren Schall aufnehmen und dann  
können wir etwas hören.")  
nextButton.isEnabled = true  
UserDefaults.standard.set(true, forKey: "Quiz3.4")  
überprüfenButton.backgroundColor = rosGrey1  
  
} else {  
    wrongAlert(vc: self)  
    button1.backgroundColor = rosGreen2  
    button2.backgroundColor = rosGreen2  
    button3.backgroundColor = rosGreen2  
    button4.backgroundColor = rosGreen2  
    button5.backgroundColor = rosGreen2  
    button6.backgroundColor = rosGreen2  
    pressedButtons[0] = false  
    pressedButtons[1] = false  
    pressedButtons[2] = false  
    pressedButtons[3] = false  
    pressedButtons[4] = false  
    pressedButtons[5] = false  
}  
}  
}
```

## ViewController3\_5

```
//  
// ViewController3_5.swift  
// Maturitätsapp Rosina  
//  
// Created by Rosina Weiß on 29.11.18.  
// Copyright © 2018 Rosina Weiß. All rights reserved.  
//  
  
import UIKit  
  
class ViewController3_5: UIViewController {  
  
    var pos: Position!  
    override func viewDidLoad() {  
        super.viewDidLoad()  
  
        //Standortpositionen  
        let pos6 = Position(name: "Waldxylophon", lat:  
            47.27306716, long: 8.55769482)  
        pos = pos6  
  
    }  
  
    override func prepare(for segue: UIStoryboardSegue, sender:  
        Any?) {  
  
        if let dest = segue.destination as?  
            UINavigationController {  
            dest.pos = pos  
        }  
    }  
}
```

---

## ViewController3\_6

```
//
// ViewController3_6.swift
// Maturitätsapp Rosina
//
// Created by Rosina Weiß on 03.12.18.
// Copyright © 2018 Rosina Weiß. All rights reserved.
//

import UIKit

class ViewController3_6: UIViewController {

    @IBOutlet weak var button1: UIButton!
    @IBOutlet weak var button2: UIButton!
    @IBOutlet weak var button3: UIButton!
    @IBOutlet weak var button4: UIButton!
    @IBOutlet weak var button5: UIButton!
    @IBOutlet weak var button6: UIButton!
    @IBOutlet weak var überprüfenButton: UIButton!
    @IBOutlet weak var nextButton: UIButton!

    var pressedButtons = [false, false, false, false, false, false]
    let correct = [false, true, true, false, true, false]

    override func viewDidLoad() {
        super.viewDidLoad()

        let quizSolved = UserDefaults.standard.bool(forKey:
            "Quiz3.6")
        nextButton.isEnabled = quizSolved

        button1.backgroundColor = rosGreen2
        button2.backgroundColor = rosGreen2
        button3.backgroundColor = rosGreen2
        button4.backgroundColor = rosGreen2
        button5.backgroundColor = rosGreen2
        button6.backgroundColor = rosGreen2
        überprüfenButton.backgroundColor = rosGrey2
    }

    @IBAction func action1(_ sender: UIButton) {
        if sender.backgroundColor != rosGreen3 {
            sender.backgroundColor = rosGreen3
            pressedButtons[0] = true
        } else {
            sender.backgroundColor = rosGreen2
            pressedButtons[0] = false
        }
    }
}
```

```
}
@IBAction func aciton2(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[1] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[1] = false
    }
}
}
@IBAction func action3(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[2] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[2] = false
    }
}
}
@IBAction func action4(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[3] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[3] = false
    }
}
}
@IBAction func action5(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[4] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[4] = false
    }
}
}
@IBAction func action6(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[5] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[5] = false
    }
}
}
@IBAction func überprüfenAciton(_ sender: Any) {
    if pressedButtons == correct{
```

---

```
correctAlert(vc: self, why: "Dies stimmt alles. :)")
nextButton.isEnabled = true
UserDefaults.standard.set(true, forKey: "Quiz3.6")
überprüfenButton.backgroundColor = rosGrey1
} else {
    wrongAlert(vc: self)
    button1.backgroundColor = rosGreen2
    button2.backgroundColor = rosGreen2
    button3.backgroundColor = rosGreen2
    button4.backgroundColor = rosGreen2
    button5.backgroundColor = rosGreen2
    button6.backgroundColor = rosGreen2
    pressedButtons[0] = false
    pressedButtons[1] = false
    pressedButtons[2] = false
    pressedButtons[3] = false
    pressedButtons[4] = false
    pressedButtons[5] = false
}
}
}
```

## ViewController3\_7

```
//  
// ViewController3_7.swift  
// Maturitätsapp Rosina  
//  
// Created by Rosina Weiß on 03.12.18.  
// Copyright © 2018 Rosina Weiß. All rights reserved.  
//  
  
import UIKit  
  
class ViewController3_7: UIViewController {  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
  
        // Schaltet den nächsten Posten im Info View Controller  
        frei  
        UserDefaults.standard.set(true, forKey: "Posten4")  
  
        setNotes(notes: 3.71)  
        setNotes(notes: 3.72)  
        setNotes(notes: 3.73)  
        setNotes(notes: 3.74)  
    }  
}
```

---

## ViewController4\_0

```
//  
// ViewController4_0.swift  
// Maturitätsapp Rosina  
//  
// Created by Rosina Weiß on 15.10.18.  
// Copyright © 2018 Rosina Weiß. All rights reserved.  
//  
  
import UIKit  
  
class ViewController4_0: UIViewController {  
  
    var pos: Position!  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
  
        //Standortpositionen  
        let pos7 = Position(name: "Material Papierflieger", lat:  
            47.27227762, long: 8.55902494)  
        pos = pos7  
    }  
  
    override func prepare(for segue: UIStoryboardSegue, sender:  
        Any?) {  
  
        if let dest = segue.destination as?  
            UINavigationController {  
            dest.pos = pos  
        }  
    }  
}
```

## ViewController4\_1

```
//
// ViewController4_1.swift
// Maturitätsapp Rosina
//
// Created by Rosina Weiß on 29.11.18.
// Copyright © 2018 Rosina Weiß. All rights reserved.
//

import UIKit

class ViewController4_1: UIViewController {

    @IBOutlet weak var buttonA: UIButton!
    @IBOutlet weak var buttonB: UIButton!
    @IBOutlet weak var buttonC: UIButton!
    @IBOutlet weak var buttonD: UIButton!
    @IBOutlet weak var überprüfenButton: UIButton!
    @IBOutlet weak var nextButton: UIButton!

    var pressedButtons = [false, false, false, false]
    let correct = [false, false, false, true]

    override func viewDidLoad() {
        super.viewDidLoad()

        let quizSolved = UserDefaults.standard.bool(forKey:
            "Quiz4.1")
        nextButton.isEnabled = quizSolved

        buttonA.backgroundColor = rosGreen2
        buttonB.backgroundColor = rosGreen2
        buttonC.backgroundColor = rosGreen2
        buttonD.backgroundColor = rosGreen2
        überprüfenButton.backgroundColor = rosGrey2
    }

    @IBAction func actionA(_ sender: UIButton) {
        if sender.backgroundColor != rosGreen3 {
            sender.backgroundColor = rosGreen3
            pressedButtons[0] = true
        } else {
            sender.backgroundColor = rosGreen2
            pressedButtons[0] = false
        }
    }

    @IBAction func actionB(_ sender: UIButton) {
        if sender.backgroundColor != rosGreen3 {
            sender.backgroundColor = rosGreen3
        }
    }
}
```



## ViewController4\_2

```
//
// ViewController4_2.swift
// Maturitätsapp Rosina
//
// Created by Rosina Weiß on 29.11.18.
// Copyright © 2018 Rosina Weiß. All rights reserved.
//

import UIKit

class ViewController4_2: UIViewController {

    @IBOutlet weak var button1: UIButton!
    @IBOutlet weak var button2: UIButton!
    @IBOutlet weak var button3: UIButton!
    @IBOutlet weak var button4: UIButton!
    @IBOutlet weak var button5: UIButton!
    @IBOutlet weak var button6: UIButton!
    @IBOutlet weak var button7: UIButton!
    @IBOutlet weak var button8: UIButton!
    @IBOutlet weak var bestätigenButton: UIButton!
    @IBOutlet weak var nextButton: UIButton!

    var pressedButtons =
        [false, false, false, false, false, false, false, false]
    let correct =
        [false, true, false, false, false, false, false, false]

    override func viewDidLoad() {
        super.viewDidLoad()

        let quizSolved = UserDefaults.standard.bool(forKey:
            "Quiz4.2")
        nextButton.isEnabled = quizSolved

        button1.backgroundColor = rosGreen2
        button2.backgroundColor = rosGreen2
        button3.backgroundColor = rosGreen2
        button4.backgroundColor = rosGreen2
        button5.backgroundColor = rosGreen2
        button6.backgroundColor = rosGreen2
        button7.backgroundColor = rosGreen2
        button8.backgroundColor = rosGreen2
        bestätigenButton.backgroundColor = rosGrey2
    }

    @IBAction func action1(_ sender: UIButton) {
        if sender.backgroundColor != rosGreen3 {
```

---

```
        sender.backgroundColor = rosGreen3
        pressedButtons[0] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[0] = false
    }
}

@IBAction func action2(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[1] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[1] = false
    }
}

@IBAction func action3(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[2] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[2] = false
    }
}

@IBAction func action4(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[3] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[3] = false
    }
}

@IBAction func action5(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[4] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[4] = false
    }
}

@IBAction func action6(_ sender: UIButton) {
```

```
        if sender.backgroundColor != rosGreen3 {
            sender.backgroundColor = rosGreen3
            pressedButtons[5] = true
        } else {
            sender.backgroundColor = rosGreen2
            pressedButtons[5] = false
        }
    }
}
@IBAction func action7(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[6] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[6] = false
    }
}
}
@IBAction func action8(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[7] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[7] = false
    }
}
}
@IBAction func bestätigenAction(_ sender: Any) {
    if pressedButtons == correct{
        correctAlert(vc: self, why: "Er fliegt im Kreis und nicht stur gerade aus. Auch kann er einen Looping drehen, wenn man ihn richtig wirft. Deswegen wird dieser Papierflieger auch Kunstflieger genannt. Dies hat damit zu tun, dass er eine sehr breite Nase hat und somit eine grössere Fläche als Widerstand darstellt. Dadurch wird er sehr stark gebremst und nur durch minimale Abweichungen vom Gleichgewicht abgelenkt.")
        nextButton.isEnabled = true
        UserDefaults.standard.set(true, forKey: "Quiz4.2")
        bestätigenButton.backgroundColor = rosGrey1
    } else {
        wrongAlert(vc: self)
        button1.backgroundColor = rosGreen2
        button2.backgroundColor = rosGreen2
        button3.backgroundColor = rosGreen2
        button4.backgroundColor = rosGreen2
        button5.backgroundColor = rosGreen2
        button6.backgroundColor = rosGreen2
    }
}
```

---

```
        button7.backgroundColor = rosGreen2
        button8.backgroundColor = rosGreen2
        pressedButtons[0] = false
        pressedButtons[1] = false
        pressedButtons[2] = false
        pressedButtons[3] = false
        pressedButtons[4] = false
        pressedButtons[5] = false
        pressedButtons[6] = false
        pressedButtons[7] = false
    }
}
}
```

## ViewController4\_3

```
//  
// ViewController4_3.swift  
// Maturitätsapp Rosina  
//  
// Created by Rosina Weiß on 29.11.18.  
// Copyright © 2018 Rosina Weiß. All rights reserved.  
//  
  
import UIKit  
  
class ViewController4_3: UIViewController {  
  
    var pos: Position!  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
  
        //Standortpositionen  
        let pos8 = Position(name: "Beweisstück Papierflieger",  
                            lat: 47.27115474, long: 8.55916542)  
        pos = pos8  
    }  
    override func prepare(for segue: UIStoryboardSegue, sender:  
        Any?) {  
  
        if let dest = segue.destination as?  
            UINavigationController {  
            dest.pos = pos  
        }  
    }  
}
```

---

## ViewController4\_4

```
//
// ViewController4_4.swift
// Maturitätsapp Rosina
//
// Created by Rosina Weiß on 29.11.18.
// Copyright © 2018 Rosina Weiß. All rights reserved.
//

import UIKit

class ViewController4_4: UIViewController {

    @IBOutlet weak var button1: UIButton!
    @IBOutlet weak var button2: UIButton!
    @IBOutlet weak var button3: UIButton!
    @IBOutlet weak var button4: UIButton!
    @IBOutlet weak var überprüfenButton: UIButton!
    @IBOutlet weak var nextButton: UIButton!

    var pressedButtons = [false, false, false, false]
    let correct = [false, true, false, false]

    override func viewDidLoad() {
        super.viewDidLoad()

        let quizSolved = UserDefaults.standard.bool(forKey:
            "Quiz4.4")
        nextButton.isEnabled = quizSolved

        button1.backgroundColor = rosGreen2
        button2.backgroundColor = rosGreen2
        button3.backgroundColor = rosGreen2
        button4.backgroundColor = rosGreen2
        überprüfenButton.backgroundColor = rosGrey2

        // Do any additional setup after loading the view.
    }

    @IBAction func action1(_ sender: UIButton) {
        if sender.backgroundColor != rosGreen3 {
            sender.backgroundColor = rosGreen3
            pressedButtons[0] = true
        } else {
            sender.backgroundColor = rosGreen2
            pressedButtons[0] = false
        }
    }

    @IBAction func action2(_ sender: UIButton) {
```

```

        if sender.backgroundColor != rosGreen3 {
            sender.backgroundColor = rosGreen3
            pressedButtons[1] = true
        } else {
            sender.backgroundColor = rosGreen2
            pressedButtons[1] = false
        }
    }
}
@IBAction func action3(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[2] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[2] = false
    }
}
@IBAction func action4(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[3] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[3] = false
    }
}
@IBAction func überprüfenAction(_ sender: Any) {
    if pressedButtons == correct{
        correctAlert(vc: self, why: "Ihr habt durch den
richtigen Papierflieger das richtige Dokument
gefunden. Dies wird sofort in eurem Notizbuch
gespeichert. Da diese Station zum Thema Aerodynamik
zu Ende ist, wurde der nächste Posten
freigeschaltet. Klickt auf das Übersichtssymbol und
startet den nächsten Posten.")
        nextButton.isEnabled = true
        UserDefaults.standard.set(true, forKey: "Quiz4.4")
        überprüfenButton.backgroundColor = rosGrey1

        // Schaltet den nächsten Posten im Info View
        Controller frei
        UserDefaults.standard.set(true, forKey: "Posten5")
    } else {
        wrongAlert(vc: self)
        button1.backgroundColor = rosGreen2
        button2.backgroundColor = rosGreen2
        button3.backgroundColor = rosGreen2
        button4.backgroundColor = rosGreen2
    }
}

```

---

```
        pressedButtons[0] = false
        pressedButtons[1] = false
        pressedButtons[2] = false
        pressedButtons[3] = false

        setNotes(notes: 4.41)
        setNotes(notes: 4.42)
    }
}
```

## ViewController5\_0

```
//  
// ViewController5_0.swift  
// Maturitätsapp Rosina  
//  
// Created by Rosina Weiß on 16.10.18.  
// Copyright © 2018 Rosina Weiß. All rights reserved.  
//  
  
import UIKit  
  
class ViewController5_0: UIViewController {  
  
    var pos: Position!  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
  
        //Standortpositionen  
        let pos9 = Position(name: "Moorwald Eingang", lat:  
            47.26911857, long: 8.56116140)  
        pos = pos9  
    }  
    override func prepare(for segue: UIStoryboardSegue, sender:  
        Any?) {  
        if let dest = segue.destination as?  
            UINavigationController {  
            dest.pos = pos  
        }  
    }  
}
```

---

## ViewController5\_1

```
//  
// ViewController5_1.swift  
// Maturitätsapp Rosina  
//  
// Created by Rosina Weiß on 29.11.18.  
// Copyright © 2018 Rosina Weiß. All rights reserved.  
//  
  
import UIKit  
  
class ViewController5_1: UIViewController {  
  
    var pos: Position!  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
  
        //Standortpositionen  
        let pos10 = Position(name: "Moorwald Aussichtsplattform",  
                             lat: 47.26926680, long: 8.56285505)  
        pos = pos10  
    }  
  
    override func prepare(for segue: UIStoryboardSegue, sender:  
        Any?) {  
  
        if let dest = segue.destination as?  
            UINavigationController {  
            dest.pos = pos  
        }  
    }  
}
```

## ViewController5\_2

```
//  
// ViewController5_2.swift  
// Maturitätsapp Rosina  
//  
// Created by Rosina Weiß on 29.11.18.  
// Copyright © 2018 Rosina Weiß. All rights reserved.  
//  
  
import UIKit  
  
class ViewController5_2: UIViewController {  
    override func viewDidLoad() {  
        super.viewDidLoad()  
    }  
}
```

---

## ViewController5\_3

```
//
// ViewController5_3.swift
// Maturitätsapp Rosina
//
// Created by Rosina Weiß on 29.11.18.
// Copyright © 2018 Rosina Weiß. All rights reserved.
//

import UIKit

class ViewController5_3: UIViewController, UITextFieldDelegate {

    @IBOutlet weak var textField: UITextField!
    @IBOutlet weak var überprüfenButton: UIButton!
    @IBOutlet weak var nextButton: UIButton!

    override func viewDidLoad() {
        super.viewDidLoad()
        textField.delegate = self

        let quizSolved = UserDefaults.standard.bool(forKey:
            "Quiz5.3")
        nextButton.isEnabled = quizSolved
        textField.backgroundColor = UIColor.green
        überprüfenButton.backgroundColor = UIColor.gray

    }
    func textFieldShouldReturn(_ textField: UITextField) -> Bool
    {
        textField.resignFirstResponder()
        return true
        //macht, dass das Keyboard verschwindet, wenn man auf
        //return drückt.
    }
    @IBAction func überprüfenAction(_ sender: UIButton) {
        if textField.text == "17:00" {
            correctAlert(vc: self, why: "Ihr habt nun
                herausgefunden, wann das Opfer erschossen wurde. Nun
                findet auf der nächsten Seite als Zusatzaufgabe
                heraus, wo in der Graphik mit dem Schatten des
                Baumes Süden liegt.")
            nextButton.isEnabled = true
            UserDefaults.standard.set(true, forKey: "Quiz5.3")
            sender.backgroundColor = UIColor.gray
        } else {
            wrongAlert(vc: self)
        }
    }
}
}
```

## ViewController5\_4

```
//
// ViewController5_4.swift
// Maturitätsapp Rosina
//
// Created by Rosina Weiß on 29.11.18.
// Copyright © 2018 Rosina Weiß. All rights reserved.
//

import UIKit

class ViewController5_4: UIViewController {

    @IBOutlet weak var button1: UIButton!
    @IBOutlet weak var button2: UIButton!
    @IBOutlet weak var button3: UIButton!
    @IBOutlet weak var button4: UIButton!
    @IBOutlet weak var button5: UIButton!
    @IBOutlet weak var button6: UIButton!
    @IBOutlet weak var überprüfenButton: UIButton!
    @IBOutlet weak var nextButton: UIButton!

    var pressedButtons = [false, false, false, false, false, false]
    let correct = [false, false, false, false, true, false]

    override func viewDidLoad() {
        super.viewDidLoad()

        let quizSolved = UserDefaults.standard.bool(forKey:
            "Quiz5.4")
        nextButton.isEnabled = quizSolved

        button1.backgroundColor = rosGreen2
        button2.backgroundColor = rosGreen2
        button3.backgroundColor = rosGreen2
        button4.backgroundColor = rosGreen2
        button5.backgroundColor = rosGreen2
        button6.backgroundColor = rosGreen2
        überprüfenButton.backgroundColor = rosGrey2

    }

    @IBAction func action1(_ sender: UIButton) {
        if sender.backgroundColor != rosGreen3 {
            sender.backgroundColor = rosGreen3
            pressedButtons[0] = true
        } else {
            sender.backgroundColor = rosGreen2
        }
    }
}
```

```

        pressedButtons[0] = false
    }
}
@IBAction func action2(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[1] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[1] = false
    }
}
@IBAction func action3(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[2] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[2] = false
    }
}
@IBAction func action4(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[3] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[3] = false
    }
}
@IBAction func action5(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[4] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[4] = false
    }
}
@IBAction func action6(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[5] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[5] = false
    }
}
}

```

```
@IBAction func überprüfenAciton(_ sender: Any) {
    if pressedButtons == correct{
        correctAlert(vc: self, why: "Wenn das Foto am Ende
            des Tages aufgenommen wurde, steht die Sonne im
            Westen. Darum ist die Sonne in der Abbildung links.
            Daraus kann man schliessen, dass der Süden vorne
            ist.")
        nextButton.isEnabled = true
        UserDefaults.standard.set(true, forKey: "Quiz5.4")
        überprüfenButton.backgroundColor = rosGrey1
    } else {
        wrongAlert(vc: self)
        button1.backgroundColor = rosGreen2
        button2.backgroundColor = rosGreen2
        button3.backgroundColor = rosGreen2
        button4.backgroundColor = rosGreen2
        button5.backgroundColor = rosGreen2
        button6.backgroundColor = rosGreen2
        pressedButtons[0] = false
        pressedButtons[1] = false
        pressedButtons[2] = false
        pressedButtons[3] = false
        pressedButtons[4] = false
        pressedButtons[5] = false
    }
}
}
```

---

## ViewController5\_5

```
//  
// ViewController5_5.swift  
// Maturitätsapp Rosina  
//  
// Created by Rosina Weiß on 29.11.18.  
// Copyright © 2018 Rosina Weiß. All rights reserved.  
//  
  
import UIKit  
  
class ViewController5_5: UIViewController {  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
  
        // Schaltet den nächsten Posten im Info View Controller  
        frei  
        UserDefaults.standard.set(true, forKey: "Posten6")  
  
        setNotes(notes: 5.51)  
        setNotes(notes: 5.52)  
        setNotes(notes: 5.53)  
        setNotes(notes: 5.54)  
    }  
}
```

## ViewController6\_0

```
//  
// ViewController6_0.swift  
// Maturitätsapp Rosina  
//  
// Created by Rosina Weiß on 16.10.18.  
// Copyright © 2018 Rosina Weiß. All rights reserved.  
//  
  
import UIKit  
  
class ViewController6_0: UIViewController {  
    var pos: Position!  
    override func viewDidLoad() {  
        super.viewDidLoad()  
  
        //Standortpositionen  
        let pos11 = Position(name: "QC bei Baumringen", lat:  
            47.26681111, long: 8.56049856)  
        pos = pos11  
    }  
  
    override func prepare(for segue: UIStoryboardSegue, sender:  
        Any?) {  
        if let dest = segue.destination as?  
            UINavigationController {  
            dest.pos = pos  
        }  
    }  
}
```

---

## ViewController6\_1

```
//  
// ViewController6_1.swift  
// Maturitätsapp Rosina  
//  
// Created by Rosina Weiß on 29.11.18.  
// Copyright © 2018 Rosina Weiß. All rights reserved.  
//  
  
import UIKit  
  
class ViewController6_1: UIViewController {  
    override func viewDidLoad() {  
        super.viewDidLoad()  
    }  
}
```

## ViewController6\_2

```
//  
// ViewController6_2.swift  
// Maturitätsapp Rosina  
//  
// Created by Rosina Weiß on 29.11.18.  
// Copyright © 2018 Rosina Weiß. All rights reserved.  
//  
  
import UIKit  
  
class ViewController6_2: UIViewController {  
    override func viewDidLoad() {  
        super.viewDidLoad()  
    }  
}
```

---

## ViewController6\_3

```
//  
// ViewController6_3.swift  
// Maturitätsapp Rosina  
//  
// Created by Rosina Weiß on 29.11.18.  
// Copyright © 2018 Rosina Weiß. All rights reserved.  
//  
  
import UIKit  
  
class ViewController6_3: UIViewController{  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
  
        setNotes(notes: 6.31)  
        setNotes(notes: 6.32)  
        setNotes(notes: 6.33)  
        setNotes(notes: 6.34)  
    }  
}
```

## ViewController6\_4

```
//  
// ViewController6_4.swift  
// Maturitätsapp Rosina  
//  
// Created by Rosina Weiß on 29.11.18.  
// Copyright © 2018 Rosina Weiß. All rights reserved.  
//  
  
import UIKit  
  
class ViewController6_4: UIViewController {  
  
    @IBOutlet weak var button1: UIButton!  
    @IBOutlet weak var button2: UIButton!  
    @IBOutlet weak var button3: UIButton!  
    @IBOutlet weak var button4: UIButton!  
    @IBOutlet weak var button5: UIButton!  
    @IBOutlet weak var button6: UIButton!  
    @IBOutlet weak var button7: UIButton!  
    @IBOutlet weak var button8: UIButton!  
    @IBOutlet weak var button9: UIButton!  
    @IBOutlet weak var button10: UIButton!  
    @IBOutlet weak var button11: UIButton!  
    @IBOutlet weak var button12: UIButton!  
    @IBOutlet weak var button13: UIButton!  
    @IBOutlet weak var button14: UIButton!  
    @IBOutlet weak var button15: UIButton!  
    @IBOutlet weak var button16: UIButton!  
    @IBOutlet weak var button17: UIButton!  
    @IBOutlet weak var button18: UIButton!  
    @IBOutlet weak var button19: UIButton!  
    @IBOutlet weak var button20: UIButton!  
    @IBOutlet weak var button21: UIButton!  
    @IBOutlet weak var button22: UIButton!  
    @IBOutlet weak var button23: UIButton!  
    @IBOutlet weak var button24: UIButton!  
    @IBOutlet weak var überprüfenbutton: UIButton!  
    @IBOutlet weak var nextButton: UIButton!  
  
    var pressedButtons =  
        [false, false, false, false, false, false, false, false, false, false,  
          false, false, false, false, false, false, false, false, false, false,  
          false, false, false, false]  
    let correct =  
        [false, false, true, false, false, false, true, false, false, false, f  
          alse, false, false, false, false, false, false, false, false, false, f  
          alse, false, false, false]  
}
```

---

```
override func viewDidLoad() {
    super.viewDidLoad()

    let quizSolved = UserDefaults.standard.bool(forKey:
        "Quiz6.4")
    nextButton.isEnabled = quizSolved

    button1.backgroundColor = rosGreen2
    button2.backgroundColor = rosGreen2
    button3.backgroundColor = rosGreen2
    button4.backgroundColor = rosGreen2
    button5.backgroundColor = rosGreen2
    button6.backgroundColor = rosGreen2
    button7.backgroundColor = rosGreen2
    button8.backgroundColor = rosGreen2
    button9.backgroundColor = rosGreen2
    button10.backgroundColor = rosGreen2
    button11.backgroundColor = rosGreen2
    button12.backgroundColor = rosGreen2
    button13.backgroundColor = rosGreen2
    button14.backgroundColor = rosGreen2
    button15.backgroundColor = rosGreen2
    button16.backgroundColor = rosGreen2
    button17.backgroundColor = rosGreen2
    button18.backgroundColor = rosGreen2
    button19.backgroundColor = rosGreen2
    button20.backgroundColor = rosGreen2
    button21.backgroundColor = rosGreen2
    button22.backgroundColor = rosGreen2
    button23.backgroundColor = rosGreen2
    button24.backgroundColor = rosGreen2
}

@IBAction func action1(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[0] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[0] = false
    }
}

@IBAction func action2(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[1] = true
    } else {
```

```
        sender.backgroundColor = rosGreen2
        pressedButtons[1] = false
    }
}
@IBAction func action3(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[2] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[2] = false
    }
}
@IBAction func action4(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[3] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[3] = false
    }
}
@IBAction func action5(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[4] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[4] = false
    }
}
@IBAction func action6(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[5] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[5] = false
    }
}
@IBAction func action7(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[6] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[6] = false
    }
}
```

```

}
@IBAction func action8(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[7] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[7] = false
    }
}
@IBAction func action9(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[8] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[8] = false
    }
}
@IBAction func action10(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[9] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[9] = false
    }
}
@IBAction func action11(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[10] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[10] = false
    }
}
@IBAction func action12(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[11] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[11] = false
    }
}
@IBAction func action13(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {

```

```
        sender.backgroundColor = rosGreen3
        pressedButtons[12] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[12] = false
    }
}
@IBAction func action14(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[13] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[13] = false
    }
}
@IBAction func action15(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[14] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[14] = false
    }
}
@IBAction func action16(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[15] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[15] = false
    }
}
@IBAction func action17(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[16] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[16] = false
    }
}
@IBAction func action18(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[17] = true
    } else {
```

---

```
        sender.backgroundColor = rosGreen2
        pressedButtons[17] = false
    }
}
@IBAction func action19(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[18] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[18] = false
    }
}
@IBAction func action20(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[19] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[19] = false
    }
}
@IBAction func action21(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[20] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[20] = false
    }
}
@IBAction func action22(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[21] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[21] = false
    }
}
@IBAction func action23(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[22] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[22] = false
    }
}
```

```
}
@IBAction func action24(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[23] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[23] = false
    }
}
@IBAction func überprüfenaction(_ sender: Any) {
    if pressedButtons == correct{
        correctAlert(vc: self, why: "Ihr habt absolut richtig
        zerlegt. Geht zur nächsten Zahl.")
        nextButton.isEnabled = true
        UserDefaults.standard.set(true, forKey: "Quiz6.4")
        überprüfenbutton.backgroundColor = rosGrey1
    } else {
        wrongAlert(vc: self)
    }
}
}
```

---

## ViewController6\_5

```
//
// ViewController6_5.swift
// Maturitätsapp Rosina
//
// Created by Rosina Weiß on 29.11.18.
// Copyright © 2018 Rosina Weiß. All rights reserved.
//

import UIKit

class ViewController6_5: UIViewController {

    @IBOutlet weak var button1: UIButton!
    @IBOutlet weak var button2: UIButton!
    @IBOutlet weak var button3: UIButton!
    @IBOutlet weak var button4: UIButton!
    @IBOutlet weak var button5: UIButton!
    @IBOutlet weak var button6: UIButton!
    @IBOutlet weak var button7: UIButton!
    @IBOutlet weak var button8: UIButton!
    @IBOutlet weak var button9: UIButton!
    @IBOutlet weak var button10: UIButton!
    @IBOutlet weak var button11: UIButton!
    @IBOutlet weak var button12: UIButton!
    @IBOutlet weak var button13: UIButton!
    @IBOutlet weak var button14: UIButton!
    @IBOutlet weak var button15: UIButton!
    @IBOutlet weak var button16: UIButton!
    @IBOutlet weak var button17: UIButton!
    @IBOutlet weak var button18: UIButton!
    @IBOutlet weak var button19: UIButton!
    @IBOutlet weak var button20: UIButton!
    @IBOutlet weak var button21: UIButton!
    @IBOutlet weak var button22: UIButton!
    @IBOutlet weak var button23: UIButton!
    @IBOutlet weak var button24: UIButton!
    @IBOutlet weak var überprüfenbutton: UIButton!
    @IBOutlet weak var nextButton: UIButton!

    var pressedButtons =
        [false, false, false, false, false, false, false, false, false, false,
        , false, false, false, false, false, false, false, false, false, false,
        , false, false, false, false]
    let correct =
        [false, true, true, false, false, false, false, false, false, false, f
        alse, false, true, false, false, false, false, false, false, false, fa
        lse, false, false, false]
```

```
override func viewDidLoad() {
    super.viewDidLoad()

    let quizSolved = UserDefaults.standard.bool(forKey:
        "Quiz6.5")
    nextButton.isEnabled = quizSolved

    button1.backgroundColor = rosGreen2
    button2.backgroundColor = rosGreen2
    button3.backgroundColor = rosGreen2
    button4.backgroundColor = rosGreen2
    button5.backgroundColor = rosGreen2
    button6.backgroundColor = rosGreen2
    button7.backgroundColor = rosGreen2
    button8.backgroundColor = rosGreen2
    button9.backgroundColor = rosGreen2
    button10.backgroundColor = rosGreen2
    button11.backgroundColor = rosGreen2
    button12.backgroundColor = rosGreen2
    button13.backgroundColor = rosGreen2
    button14.backgroundColor = rosGreen2
    button15.backgroundColor = rosGreen2
    button16.backgroundColor = rosGreen2
    button17.backgroundColor = rosGreen2
    button18.backgroundColor = rosGreen2
    button19.backgroundColor = rosGreen2
    button20.backgroundColor = rosGreen2
    button21.backgroundColor = rosGreen2
    button22.backgroundColor = rosGreen2
    button23.backgroundColor = rosGreen2
    button24.backgroundColor = rosGreen2

}

@IBAction func action1(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[0] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[0] = false
    }
}

@IBAction func action2(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[1] = true
    } else {
```

---

```
        sender.backgroundColor = rosGreen2
        pressedButtons[1] = false
    }
}
@IBAction func action3(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[2] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[2] = false
    }
}
@IBAction func action4(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[3] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[3] = false
    }
}
@IBAction func action5(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[4] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[4] = false
    }
}
@IBAction func action6(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[5] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[5] = false
    }
}
@IBAction func action7(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[6] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[6] = false
    }
}
```

```
}
@IBAction func action8(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[7] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[7] = false
    }
}
}
@IBAction func action9(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[8] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[8] = false
    }
}
}
@IBAction func action10(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[9] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[9] = false
    }
}
}
@IBAction func action11(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[10] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[10] = false
    }
}
}
@IBAction func action12(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[11] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[11] = false
    }
}
}
@IBAction func action13(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
```

---

```
        sender.backgroundColor = rosGreen3
        pressedButtons[12] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[12] = false
    }
}
}
@IBAction func action14(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[13] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[13] = false
    }
}
}
@IBAction func action15(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[14] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[14] = false
    }
}
}
@IBAction func action16(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[15] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[15] = false
    }
}
}
@IBAction func action17(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[16] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[16] = false
    }
}
}
@IBAction func action18(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[17] = true
    } else {
```

```
        sender.backgroundColor = rosGreen2
        pressedButtons[17] = false
    }
}
@IBAction func action19(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[18] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[18] = false
    }
}
@IBAction func action20(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[19] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[19] = false
    }
}
@IBAction func action21(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[20] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[20] = false
    }
}
@IBAction func action22(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[21] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[21] = false
    }
}
@IBAction func action23(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[22] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[22] = false
    }
}
```

---

```

    }
    @IBAction func action24(_ sender: UIButton) {
        if sender.backgroundColor != rosGreen3 {
            sender.backgroundColor = rosGreen3
            pressedButtons[23] = true
        } else {
            sender.backgroundColor = rosGreen2
            pressedButtons[23] = false
        }
    }
    @IBAction func überprüfenaction(_ sender: Any) {
        if pressedButtons == correct{
            correctAlert(vc: self, why: "Ihr habt absolut richtig
                zerlegt. Geht zur nächsten Zahl.")
            nextButton.isEnabled = true
            UserDefaults.standard.set(true, forKey: "Quiz6.5")
            überprüfenbutton.backgroundColor = rosGrey1

        } else {
            wrongAlert(vc: self)
        }
    }
}

```

## ViewController6\_6

```
//  
// ViewController6_6.swift  
// Maturitätsapp Rosina  
//  
// Created by Rosina Weiß on 03.12.18.  
// Copyright © 2018 Rosina Weiß. All rights reserved.  
//  
  
import UIKit  
  
class ViewController6_6: UIViewController {  
  
    @IBOutlet weak var button1: UIButton!  
    @IBOutlet weak var button2: UIButton!  
    @IBOutlet weak var button3: UIButton!  
    @IBOutlet weak var button4: UIButton!  
    @IBOutlet weak var button5: UIButton!  
    @IBOutlet weak var button6: UIButton!  
    @IBOutlet weak var button7: UIButton!  
    @IBOutlet weak var button8: UIButton!  
    @IBOutlet weak var button9: UIButton!  
    @IBOutlet weak var button10: UIButton!  
    @IBOutlet weak var button11: UIButton!  
    @IBOutlet weak var button12: UIButton!  
    @IBOutlet weak var button13: UIButton!  
    @IBOutlet weak var button14: UIButton!  
    @IBOutlet weak var button15: UIButton!  
    @IBOutlet weak var button16: UIButton!  
    @IBOutlet weak var button17: UIButton!  
    @IBOutlet weak var button18: UIButton!  
    @IBOutlet weak var button19: UIButton!  
    @IBOutlet weak var button20: UIButton!  
    @IBOutlet weak var button21: UIButton!  
    @IBOutlet weak var button22: UIButton!  
    @IBOutlet weak var button23: UIButton!  
    @IBOutlet weak var button24: UIButton!  
    @IBOutlet weak var überprüfenbutton: UIButton!  
    @IBOutlet weak var nextButton: UIButton!  
  
    var pressedButtons =  
        [false, false, false, false, false, false, false, false, false, false,  
          false, false, false, false, false, false, false, false, false, false,  
          false, false, false, false]  
    let correct =  
        [false, true, true, false, false, false, true, false, false, false, tr  
ue, false, false, false, false, false, false, false, false, false, fal  
se, false, false, false]  
}
```

---

```
override func viewDidLoad() {
    super.viewDidLoad()

    let quizSolved = UserDefaults.standard.bool(forKey:
        "Quiz6.6")
    nextButton.isEnabled = quizSolved

    button1.backgroundColor = rosGreen2
    button2.backgroundColor = rosGreen2
    button3.backgroundColor = rosGreen2
    button4.backgroundColor = rosGreen2
    button5.backgroundColor = rosGreen2
    button6.backgroundColor = rosGreen2
    button7.backgroundColor = rosGreen2
    button8.backgroundColor = rosGreen2
    button9.backgroundColor = rosGreen2
    button10.backgroundColor = rosGreen2
    button11.backgroundColor = rosGreen2
    button12.backgroundColor = rosGreen2
    button13.backgroundColor = rosGreen2
    button14.backgroundColor = rosGreen2
    button15.backgroundColor = rosGreen2
    button16.backgroundColor = rosGreen2
    button17.backgroundColor = rosGreen2
    button18.backgroundColor = rosGreen2
    button19.backgroundColor = rosGreen2
    button20.backgroundColor = rosGreen2
    button21.backgroundColor = rosGreen2
    button22.backgroundColor = rosGreen2
    button23.backgroundColor = rosGreen2
    button24.backgroundColor = rosGreen2

}

@IBAction func action1(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[0] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[0] = false
    }
}

@IBAction func action2(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[1] = true
    } else {
```

```
        sender.backgroundColor = rosGreen2
        pressedButtons[1] = false
    }
}
@IBAction func action3(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[2] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[2] = false
    }
}
@IBAction func action4(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[3] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[3] = false
    }
}
@IBAction func action5(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[4] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[4] = false
    }
}
@IBAction func action6(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[5] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[5] = false
    }
}
@IBAction func action7(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[6] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[6] = false
    }
}
```

```

}
@IBAction func action8(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[7] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[7] = false
    }
}
@IBAction func action9(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[8] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[8] = false
    }
}
@IBAction func action10(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[9] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[9] = false
    }
}
@IBAction func action11(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[10] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[10] = false
    }
}
@IBAction func action12(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[11] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[11] = false
    }
}
@IBAction func action13(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {

```

```
        sender.backgroundColor = rosGreen3
        pressedButtons[12] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[12] = false
    }
}
@IBAction func action14(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[13] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[13] = false
    }
}
@IBAction func action15(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[14] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[14] = false
    }
}
@IBAction func action16(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[15] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[15] = false
    }
}
@IBAction func action17(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[16] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[16] = false
    }
}
@IBAction func action18(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[17] = true
    } else {
```

---

```
        sender.backgroundColor = rosGreen2
        pressedButtons[17] = false
    }
}
@IBAction func action19(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[18] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[18] = false
    }
}
@IBAction func action20(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[19] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[19] = false
    }
}
@IBAction func action21(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[20] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[20] = false
    }
}
@IBAction func action22(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[21] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[21] = false
    }
}
@IBAction func action23(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[22] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[22] = false
    }
}
```

```
}
@IBAction func action24(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[23] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[23] = false
    }
}
@IBAction func überprüfenaction(_ sender: Any) {
    if pressedButtons == correct{
        correctAlert(vc: self, why: "Ihr habt absolut richtig
        zerlegt. Geht zur nächsten Zahl.")
        nextButton.isEnabled = true
        UserDefaults.standard.set(true, forKey: "Quiz6.6")
        überprüfenbutton.backgroundColor = rosGrey1
    } else {
        wrongAlert(vc: self)
    }
}
}
```

---

## ViewController6\_7

```
//
// ViewController6_7.swift
// Maturitätsapp Rosina
//
// Created by Rosina Weiß on 03.12.18.
// Copyright © 2018 Rosina Weiß. All rights reserved.
//

import UIKit

class ViewController6_7: UIViewController {

    @IBOutlet weak var button1: UIButton!
    @IBOutlet weak var button2: UIButton!
    @IBOutlet weak var button3: UIButton!
    @IBOutlet weak var button4: UIButton!
    @IBOutlet weak var button5: UIButton!
    @IBOutlet weak var button6: UIButton!
    @IBOutlet weak var button7: UIButton!
    @IBOutlet weak var button8: UIButton!
    @IBOutlet weak var button9: UIButton!
    @IBOutlet weak var button10: UIButton!
    @IBOutlet weak var button11: UIButton!
    @IBOutlet weak var button12: UIButton!
    @IBOutlet weak var button13: UIButton!
    @IBOutlet weak var button14: UIButton!
    @IBOutlet weak var button15: UIButton!
    @IBOutlet weak var button16: UIButton!
    @IBOutlet weak var button21: UIButton!
    @IBOutlet weak var überprüfenbutton: UIButton!
    @IBOutlet weak var nextButton: UIButton!

    var pressedButtons =
        [false, false, false, false, false, false, false, false, false, false,
        , false, false, false, false, false, false, false]
    let correct1 =
        [false, false, false, false, false, false, false, false, false, false,
        , false, false, false, false, false, false, true]
    let correct2 =
        [true, true, true, true, false, false, false, false, false, false, fal
        se, false, true, false, false, false, false]
    let correct3 =
        [true, true, true, true, false, false, false, false, false, false, fal
        se, false, true, false, false, false, true]

    override func viewDidLoad() {
        super.viewDidLoad()
    }
}
```

```
let quizSolved = UserDefaults.standard.bool(forKey:
    "Quiz6.7")
nextButton.isEnabled = quizSolved

button1.backgroundColor = rosGreen2
button2.backgroundColor = rosGreen2
button3.backgroundColor = rosGreen2
button4.backgroundColor = rosGreen2
button5.backgroundColor = rosGreen2
button6.backgroundColor = rosGreen2
button7.backgroundColor = rosGreen2
button8.backgroundColor = rosGreen2
button9.backgroundColor = rosGreen2
button10.backgroundColor = rosGreen2
button11.backgroundColor = rosGreen2
button12.backgroundColor = rosGreen2
button13.backgroundColor = rosGreen2
button14.backgroundColor = rosGreen2
button15.backgroundColor = rosGreen2
button16.backgroundColor = rosGreen2
button21.backgroundColor = rosGreen2

}

@IBAction func action1(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[0] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[0] = false
    }
}

@IBAction func action2(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[1] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[1] = false
    }
}

@IBAction func action3(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[2] = true
    } else {
        sender.backgroundColor = rosGreen2
    }
}
```

```

        pressedButtons[2] = false
    }
}
@IBAction func action4(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[3] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[3] = false
    }
}
@IBAction func action5(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[4] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[4] = false
    }
}
@IBAction func action6(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[5] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[5] = false
    }
}
@IBAction func action7(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[6] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[6] = false
    }
}
@IBAction func action8(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[7] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[7] = false
    }
}
}

```

```
@IBAction func action9(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[8] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[8] = false
    }
}
@IBAction func action10(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[9] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[9] = false
    }
}
@IBAction func action11(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[10] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[10] = false
    }
}
@IBAction func action12(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[11] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[11] = false
    }
}
@IBAction func action13(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[12] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[12] = false
    }
}
@IBAction func action14(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
```



## ViewController6\_8

```
//  
// ViewController6_8.swift  
// Maturitätsapp Rosina  
//  
// Created by Rosina Weiß on 03.12.18.  
// Copyright © 2018 Rosina Weiß. All rights reserved.  
//  
  
import UIKit  
  
class ViewController6_8: UIViewController {  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
  
        // Schaltet den nächsten Posten im Info View Controller  
        frei  
        UserDefaults.standard.set(true, forKey: "Posten7")  
        setNotes(notes: 6.81)  
        setNotes(notes: 6.82)  
        setNotes(notes: 6.83)  
        setNotes(notes: 6.84)  
        setNotes(notes: 6.85)  
        setNotes(notes: 6.86)  
        setNotes(notes: 6.87)  
        setNotes(notes: 6.88)  
        setNotes(notes: 6.89)  
    }  
}
```

---

## ViewController7\_0

```
//  
// ViewController7_0.swift  
// Maturitätsapp Rosina  
//  
// Created by Rosina Weiß on 16.10.18.  
// Copyright © 2018 Rosina Weiß. All rights reserved.  
//  
  
import UIKit  
  
class ViewController7_0: UIViewController {  
  
    var pos: Position!  
    override func viewDidLoad() {  
        super.viewDidLoad()  
  
        //Standortpositionen  
        let pos12 = Position(name: "Fussabdruck", lat:  
            47.26651662, long: 8.56027904)  
        pos = pos12  
  
    }  
  
    override func prepare(for segue: UIStoryboardSegue, sender:  
        Any?) {  
  
        if let dest = segue.destination as?  
            UINavigationController {  
            dest.pos = pos  
        }  
    }  
}
```

## ViewController7\_1

```
//  
// ViewController7_1.swift  
// Maturitätsapp Rosina  
//  
// Created by Rosina Weiß on 29.11.18.  
// Copyright © 2018 Rosina Weiß. All rights reserved.  
//  
  
import UIKit  
  
class ViewController7_1: UIViewController {  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
  
        setNotes(notes: 7.1)  
    }  
}
```

---

## ViewController7\_2

```
//  
// ViewController7_2.swift  
// Maturitätsapp Rosina  
//  
// Created by Rosina Weiß on 29.11.18.  
// Copyright © 2018 Rosina Weiß. All rights reserved.  
//  
import UIKit  
  
class ViewController7_2: UIViewController {  
    override func viewDidLoad() {  
        super.viewDidLoad()  
    }  
}
```

## ViewController7\_3

```
//
// ViewController7_3.swift
// Maturitätsapp Rosina
//
// Created by Rosina Weiß on 29.11.18.
// Copyright © 2018 Rosina Weiß. All rights reserved.
//

import UIKit

class ViewController7_3: UIViewController {

    @IBOutlet weak var button1: UIButton!
    @IBOutlet weak var button2: UIButton!
    @IBOutlet weak var button3: UIButton!
    @IBOutlet weak var button4: UIButton!
    @IBOutlet weak var überprüfenButton: UIButton!
    @IBOutlet weak var nextButton: UIButton!

    var pressedButtons = [false, false, false, false]
    let correct = [false, true, false, false]

    override func viewDidLoad() {
        super.viewDidLoad()

        let quizSolved = UserDefaults.standard.bool(forKey:
            "Quiz7.3")
        nextButton.isEnabled = quizSolved

        button1.backgroundColor = rosGreen2
        button2.backgroundColor = rosGreen2
        button3.backgroundColor = rosGreen2
        button4.backgroundColor = rosGreen2
        überprüfenButton.backgroundColor = rosGrey2

        setNotes(notes: 7.31)
        setNotes(notes: 7.32)
        setNotes(notes: 7.33)
        setNotes(notes: 7.34)
    }
    @IBAction func aciton1(_ sender: UIButton) {
        if sender.backgroundColor != rosGreen3 {
            sender.backgroundColor = rosGreen3
            pressedButtons[0] = true
        } else {
            sender.backgroundColor = rosGreen2
            pressedButtons[0] = false
        }
    }
}
```

```

}
@IBAction func action2(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[1] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[1] = false
    }
}
@IBAction func action3(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[2] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[2] = false
    }
}
@IBAction func action4(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[3] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[3] = false
    }
}
@IBAction func überprüfenAction(_ sender: Any) {
    if pressedButtons == correct {
        correctAlert(vc: self, why: "Dies lässt darauf
        schliessen, dass der Fussabdruck wahrscheinlich von
        einer weiblichen Person stammt. Damit könnte man
        alle Männer als Mörder ausschliessen und auch den
        Verdacht auf Selbstmord wiederlegen. Jedoch ist noch
        nicht festgestellt worden, ob der Fussabdruck
        wirklich vom Täter stammt oder einfach von einem
        Pasanten.")
        nextButton.isEnabled = true
        UserDefaults.standard.set(true, forKey: "Quiz7.3")
        überprüfenButton.backgroundColor = rosGrey1
    } else {
        wrongAlert(vc: self)
        button1.backgroundColor = rosGreen2
        button2.backgroundColor = rosGreen2
        button3.backgroundColor = rosGreen2
    }
}

```

```
        button4.backgroundColor = rosGreen2
        pressedButtons[0] = false
        pressedButtons[1] = false
        pressedButtons[2] = false
        pressedButtons[3] = false
    }
}
}
```

---

## ViewController7\_4

```
//  
// ViewController7_4.swift  
// Maturitätsapp Rosina  
//  
// Created by Rosina Weiß on 29.11.18.  
// Copyright © 2018 Rosina Weiß. All rights reserved.  
//  
  
import UIKit  
  
class ViewController7_4: UIViewController {  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
  
        setNotes(notes: 7.41)  
        setNotes(notes: 7.42)  
  
    }  
}
```

## ViewController7\_5

```
//  
// ViewController7_5.swift  
// Maturitätsapp Rosina  
//  
// Created by Rosina Weiß on 29.11.18.  
// Copyright © 2018 Rosina Weiß. All rights reserved.  
//  
  
import UIKit  
  
class ViewController7_5: UIViewController {  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
  
        // Schaltet den nächsten Posten im Info View Controller  
        frei  
        UserDefaults.standard.set(true, forKey: "Posten8")  
  
        setNotes(notes: 7.51)  
    }  
}
```

---

## ViewController9\_0

```
//  
// ViewController9_0.swift  
// Maturitätsapp Rosina  
//  
// Created by Rosina Weiß on 16.10.18.  
// Copyright © 2018 Rosina Weiß. All rights reserved.  
//  
  
import UIKit  
  
class ViewController9_0: UIViewController {  
  
    var pos: Position!  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
  
        //Standortpositionen  
        let pos13 = Position(name: "Tatort", lat: 47.29089781,  
            long: 8.55939123)  
        pos = pos13  
  
        setNotes(notes: 9.01)  
        setNotes(notes: 9.02)  
    }  
    override func prepare(for segue: UIStoryboardSegue, sender:  
        Any?) {  
        if let dest = segue.destination as?  
            UINavigationController {  
            dest.pos = pos  
        }  
    }  
}
```

## ViewController9\_1

```
//  
// ViewController9_1.swift  
// Maturitätsapp Rosina  
//  
// Created by Rosina Weiß on 22.10.18.  
// Copyright © 2018 Rosina Weiß. All rights reserved.  
//  
  
import UIKit  
  
class ViewController9_1: UIViewController {  
  
    @IBOutlet weak var nextButton: UIButton!  
    @IBOutlet weak var button1: UIButton!  
    @IBOutlet weak var button2: UIButton!  
    @IBOutlet weak var button3: UIButton!  
    @IBOutlet weak var button4: UIButton!  
    @IBOutlet weak var überprüfenButton: UIButton!  
  
    var pressedButtons = [false, false, false, false]  
    let correct = [false, false, true, false]  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
  
        button1.backgroundColor = rosGreen2  
        button2.backgroundColor = rosGreen2  
        button3.backgroundColor = rosGreen2  
        button4.backgroundColor = rosGreen2  
        überprüfenButton.backgroundColor = rosGrey2  
  
        let quizSolved = UserDefaults.standard.bool(forKey:  
            "Quiz9.1")  
        nextButton.isEnabled = quizSolved  
    }  
  
    @IBAction func action1(_ sender: UIButton) {  
        if sender.backgroundColor != rosGreen3 {  
            sender.backgroundColor = rosGreen3  
            pressedButtons[0] = true  
        } else {  
            sender.backgroundColor = rosGreen2  
            pressedButtons[0] = false  
        }  
    }  
  
    @IBAction func action2(_ sender: UIButton) {  
        if sender.backgroundColor != rosGreen3 {  
            sender.backgroundColor = rosGreen3  
        }  
    }  
}
```

```

        pressedButtons[1] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[1] = false
    }
}
@IBAction func action3(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[2] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[2] = false
    }
}
@IBAction func action4(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[3] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[3] = false
    }
}
@IBAction func ueberprüfenButton(_ sender: UIButton) {
    if pressedButtons == correct{
        correctAlert(vc: self, why: "Der Standort Y ist
richtig. Durch die Geschwindigkeit v und die Zeit t
mithilfe der Formel  $s = v * t$  kann man die
Entfernung zwischen dem Opfer und dem Täter
berechnen. Somit bekommt man dann:  $0.016s * 250m/s = 4m$ . Aufgrund der nun ausgerechneten Entfernung, kann
man mithilfe des Massstabes in der Graphik
abschätzen, welche der Punkte vier Meter vom Opfer
entfernt ist. Diese Zahl wird später noch eine Rolle
spielen und wird deswegen im Notzibuch
gespeichert.")
        nextButton.isEnabled = true
        UserDefaults.standard.set(true, forKey: "Quiz9.1")
        sender.backgroundColor = rosGrey1
    } else {
        wrongAlert(vc: self)
        button1.backgroundColor = rosGreen2
        button2.backgroundColor = rosGreen2
        button3.backgroundColor = rosGreen2
        button4.backgroundColor = rosGreen2
        pressedButtons[0] = false
        pressedButtons[1] = false
    }
}

```

```
        pressedButtons[2] = false  
        pressedButtons[3] = false  
    }  
}  
}
```

---

## ViewController9\_2

```
//  
// ViewController9_2.swift  
// Maturitätsapp Rosina  
//  
// Created by Rosina Weiß on 22.10.18.  
// Copyright © 2018 Rosina Weiß. All rights reserved.  
//  
  
import UIKit  
  
class ViewController9_2: UIViewController {  
    override func viewDidLoad() {  
        super.viewDidLoad()  
    }  
}
```

## ViewController9\_3

```
//
// ViewController9_3.swift
// Maturitätsapp Rosina
//
// Created by Rosina Weiß on 22.10.18.
// Copyright © 2018 Rosina Weiß. All rights reserved.
//

import UIKit

class ViewController9_3: UIViewController {

    @IBOutlet weak var bestätigenButton: UIButton!
    @IBOutlet weak var nextButton: UIButton!
    @IBOutlet weak var label: UILabel!

    var pressedButtons =
        [false, false, false, false, false, false, false]
    let correct = [true, false, false, false, false, false]

    override func viewDidLoad() {
        super.viewDidLoad()

        bestätigenButton.backgroundColor = rosGrey1
        label.backgroundColor = rosGrey1

        let quizSolved = UserDefaults.standard.bool(forKey:
            "Quiz9.3")
        nextButton.isEnabled = quizSolved
    }

    @IBAction func action1(_ sender: UIButton) {
        if sender.backgroundColor != rosGrey1 {
            sender.backgroundColor = rosGrey1
            pressedButtons[0] = true
        } else {
            sender.backgroundColor = rosDefault
            pressedButtons[0] = false
        }
    }

    @IBAction func action2(_ sender: UIButton) {
        if sender.backgroundColor != rosGrey1 {
            sender.backgroundColor = rosGrey1
            pressedButtons[1] = true
        } else {
```

---

```

        sender.backgroundColor = rosDefault
        pressedButtons[1] = false
    }
}
@IBAction func action3(_ sender: UIButton) {
    if sender.backgroundColor != rosGrey1 {
        sender.backgroundColor = rosGrey1
        pressedButtons[2] = true
    } else {
        sender.backgroundColor = rosDefault
        pressedButtons[2] = false
    }
}
@IBAction func action4(_ sender: UIButton) {
    if sender.backgroundColor != rosGrey1 {
        sender.backgroundColor = rosGrey1
        pressedButtons[3] = true
    } else {
        sender.backgroundColor = rosDefault
        pressedButtons[3] = false
    }
}
@IBAction func action5(_ sender: UIButton) {
    if sender.backgroundColor != rosGrey1 {
        sender.backgroundColor = rosGrey1
        pressedButtons[4] = true
    } else {
        sender.backgroundColor = rosDefault
        pressedButtons[4] = false
    }
}
@IBAction func action6(_ sender: UIButton) {
    if sender.backgroundColor != rosGrey1 {
        sender.backgroundColor = rosGrey1
        pressedButtons[5] = true
    } else {
        sender.backgroundColor = rosDefault
        pressedButtons[5] = false
    }
}
@IBAction func bestätigenAction(_ sender: UIButton)
{nextButton.isEnabled = true
    UserDefaults.standard.set(true, forKey: "Quiz9.3")
    sender.backgroundColor = rosGrey2
}
}
}

```

## ViewController9\_4

```
//
// ViewController9_4.swift
// Maturitätsapp Rosina
//
// Created by Rosina Weiß on 22.10.18.
// Copyright © 2018 Rosina Weiß. All rights reserved.
//

import UIKit

class ViewController9_4: UIViewController {

    @IBOutlet weak var button1: UIButton!
    @IBOutlet weak var button2: UIButton!
    @IBOutlet weak var button3: UIButton!
    @IBOutlet weak var button4: UIButton!
    @IBOutlet weak var button5: UIButton!
    @IBOutlet weak var button6: UIButton!
    @IBOutlet weak var button7: UIButton!
    @IBOutlet weak var button8: UIButton!
    @IBOutlet weak var nextButton: UIButton!
    @IBOutlet weak var bestätigenButton: UIButton!

    var pressedButtons =
        [false, false, false, false, false, false, false, false]
    let correct =
        [false, false, true, false, false, false, false, false]
    override func viewDidLoad() {
        super.viewDidLoad()

        // Schaltet den nächsten Posten im Info View Controller
        frei
        UserDefaults.standard.set(true, forKey: "Posten9")

        button1.backgroundColor = rosGreen2
        button2.backgroundColor = rosGreen2
        button3.backgroundColor = rosGreen2
        button4.backgroundColor = rosGreen2
        button5.backgroundColor = rosGreen2
        button6.backgroundColor = rosGreen2
        button7.backgroundColor = rosGreen2
        button8.backgroundColor = rosGreen2
        bestätigenButton.backgroundColor = rosGrey2

        let quizSolved = UserDefaults.standard.bool(forKey:
            "Quiz9.4")
        nextButton.isEnabled = quizSolved
    }
}
```

```

}

@IBAction func action1(_ sender: UIButton) {
    if sender.backgroundColor != UIColor.green {
        sender.backgroundColor = UIColor.green
        pressedButtons[0] = true
    } else {
        sender.backgroundColor = UIColor.lightGreen
        pressedButtons[0] = false
    }
}

@IBAction func action2(_ sender: UIButton) {
    if sender.backgroundColor != UIColor.green {
        sender.backgroundColor = UIColor.green
        pressedButtons[1] = true
    } else {
        sender.backgroundColor = UIColor.lightGreen
        pressedButtons[1] = false
    }
}

@IBAction func action3(_ sender: UIButton) {
    if sender.backgroundColor != UIColor.green {
        sender.backgroundColor = UIColor.green
        pressedButtons[2] = true
    } else {
        sender.backgroundColor = UIColor.lightGreen
        pressedButtons[2] = false
    }
}

@IBAction func action4(_ sender: UIButton) {
    if sender.backgroundColor != UIColor.green {
        sender.backgroundColor = UIColor.green
        pressedButtons[3] = true
    } else {
        sender.backgroundColor = UIColor.lightGreen
        pressedButtons[3] = false
    }
}

@IBAction func action5(_ sender: UIButton) {
    if sender.backgroundColor != UIColor.green {
        sender.backgroundColor = UIColor.green
        pressedButtons[4] = true
    } else {
        sender.backgroundColor = UIColor.lightGreen
        pressedButtons[4] = false
    }
}

@IBAction func action6(_ sender: UIButton) {

```

```
        if sender.backgroundColor != rosGreen3 {
            sender.backgroundColor = rosGreen3
            pressedButtons[5] = true
        } else {
            sender.backgroundColor = rosGreen2
            pressedButtons[5] = false
        }
    }
}
@IBAction func action7(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[6] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[6] = false
    }
}
@IBAction func action8(_ sender: UIButton) {
    if sender.backgroundColor != rosGreen3 {
        sender.backgroundColor = rosGreen3
        pressedButtons[7] = true
    } else {
        sender.backgroundColor = rosGreen2
        pressedButtons[7] = false
    }
}
}
@IBAction func bestätigenAction(_ sender: UIButton)
    {nextButton.isEnabled = true
    UserDefaults.standard.set(true, forKey: "Quiz9.4")
    sender.backgroundColor = rosGrey1
}
}
```

---

## ViewController10\_0

```
//  
// ViewController10_0.swift  
// Maturitätsapp Rosina  
//  
// Created by Rosina Weiß on 16.10.18.  
// Copyright © 2018 Rosina Weiß. All rights reserved.  
//  
  
import UIKit  
  
class ViewController10_0: UIViewController {  
    override func viewDidLoad() {  
        super.viewDidLoad()  
    }  
}
```

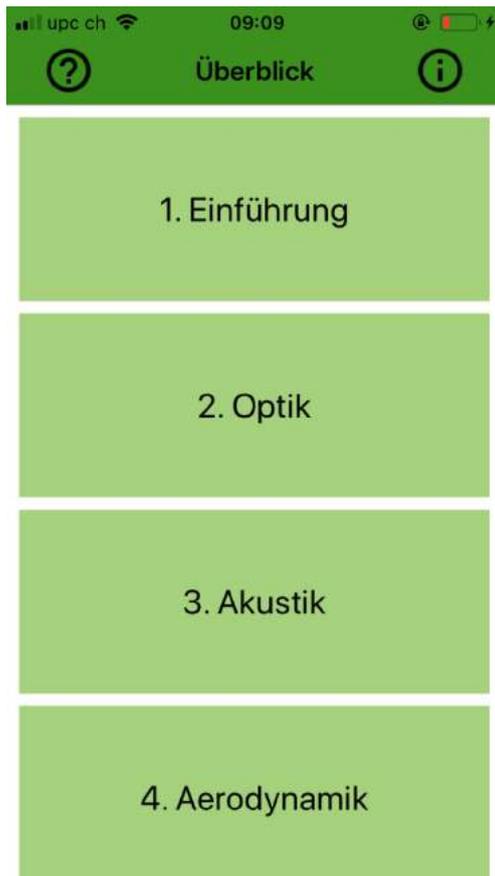
## ViewController10\_1

```
//  
// ViewController10_1.swift  
// Maturitätsapp Rosina  
//  
// Created by Rosina Weiß on 29.11.18.  
// Copyright © 2018 Rosina Weiß. All rights reserved.  
//  
  
import UIKit  
  
class ViewController10_1: UIViewController {  
    override func viewDidLoad() {  
        super.viewDidLoad()  
    }  
}
```

---

## 13.2. App UIViewController

Hier angefügt sind Screenshots aller UIViewController meiner App. Sie sind nach Auftreten innerhalb der App sortiert. Hierbei ist anzufügen, dass ich oft eine Scroll-Funktion gebrauche, welche es ermöglicht auf dem Screen herauf und herunter zu scrollen. Deshalb enden gewisse Texte auf den Screenshots unfertig, obwohl sie in der App noch weitergehen.



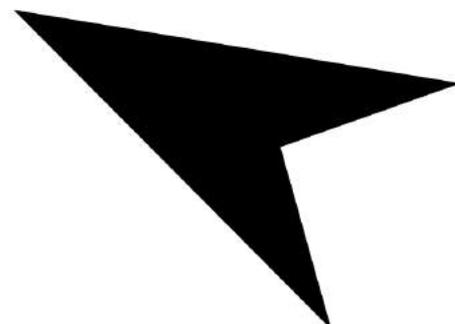
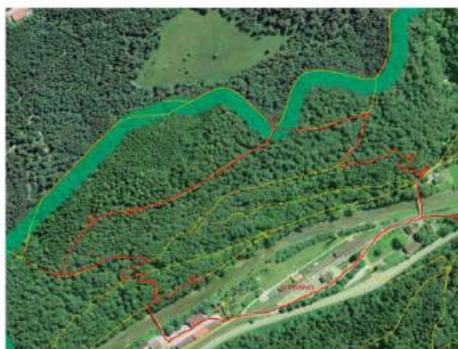
Diese App ist das Endprodukt einer Maturitätsarbeit von Rosina Weiß. Ich habe innerhalb meiner Arbeit einen Spaziergag im Sihlwald entwickelt, auf dem man durch die Hilfe von dieser App und von physikalischen Phänomenen einen Mordfall aufklärt.

Support unter:  
[dasnotizbuch@gmx.ch](mailto:dasnotizbuch@gmx.ch)

Version 1.0



Leichenfundort



→ 2641 m

47.27039417 N / 8.55732123 O

<+47.28930303,+8.53618853> +/- 10.00m (spe...



Herzlich Willkommen bei der Maturitätsarbeit von Rosina Weiß. Falls ihr mehr darüber wissen wollt, geht auf den Übersichtsscreen zurück und drückt auf den Infobutton.

Heute ist im Sihlwald ein Mord passiert und die Polizei ist aufgrund eures physikalischen Denkens auf eure Hilfe angewiesen.

Bevor ihr jedoch anfangt den Spuren hinterherzugehen, möchte ich euch eine kleine Einführung in meine App geben, damit ihr wisst wie sie funktioniert:

1. Überblick: Wie ihr zu Beginn der App schon gemerkt habt, gibt es am Anfang eine Übersicht aller Posten, die ihr heute lösen werdet. Wenn ihr mit einem Posten fertig seid, kommt man auf die Überblicksseite, auf der der nächste Posten freigeschaltet wurde und ausgeführt werden kann.



Wir befinden uns im Sihlwald des Wildnisparks Zürich und da gibt es einige Regeln zu beachten.

Bitte lest euch, die oben stehenden Sihlwaldregeln durch und haltet sie auch ein.

Zusätzlich ist hier anzumerken, dass der Weg meiner App nicht velo- noch



Startpunkt des Falls ist auf der Holzbrücke über der Sihl. Geht vom Bahnhof her nach norden und an den Fischottern und dem Spielplatz vorbei.

Um dort hinzukommen, könnt ihr auch auf den Knopf unten in der Mitte drücken.

Dieser führt euch zum Navigationspfeil.

Der Pfeil zeigt euch die Richtung und die Entfernung an, die ihr bis zum Ziel gehen müsst.

Euer Navigationsgerät ist bis auf ca. 15 Meter genau, also wird der Hintergrund grün, wenn ihr im richtigen Umfeld seit.

Hierbei ist anzufügen, dass der Pfeil immer nur die direkte Luftlinie anzeigt, das heisst bei grossen Entfernungen, geht man nicht immer sofort in die richtige Richtung, sondern dem Weg entlang.

Sollte man irgendwann nicht mehr



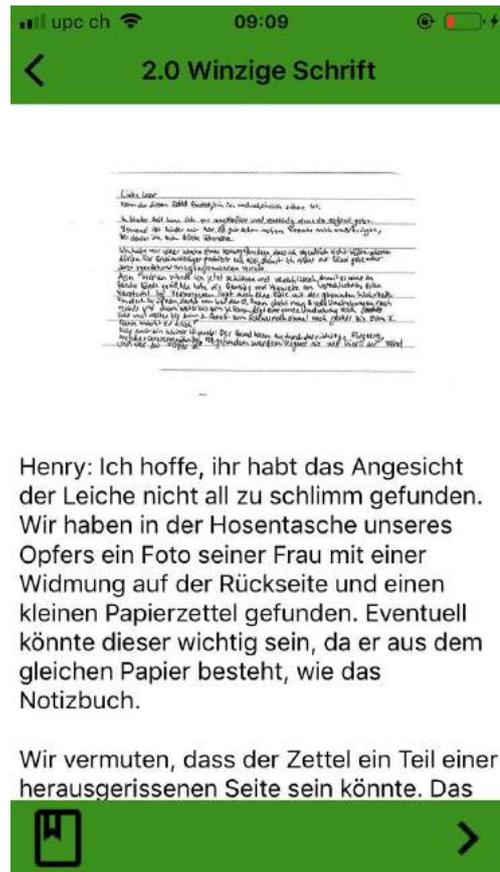
Henry: Guten Tag alle miteinander!

Mein Name ist Komissar Henry und ich leite die Untersuchung im Fall Gutenberg. Ich bin wirklich sehr froh, dass ihr gekommen seit. Jetzt könnt ihr Neulinge bei der Polizei mithelfen, die physikalischen Aspekte in dem Fall Gutenberg zu lösen.

Das Opfer ist ein Physiker namens Julian Gutenberg. Er wurde erschossen und trieb dann eine Zeit lang in der Sihl bis er am Ufer angeschwemmt wurde.

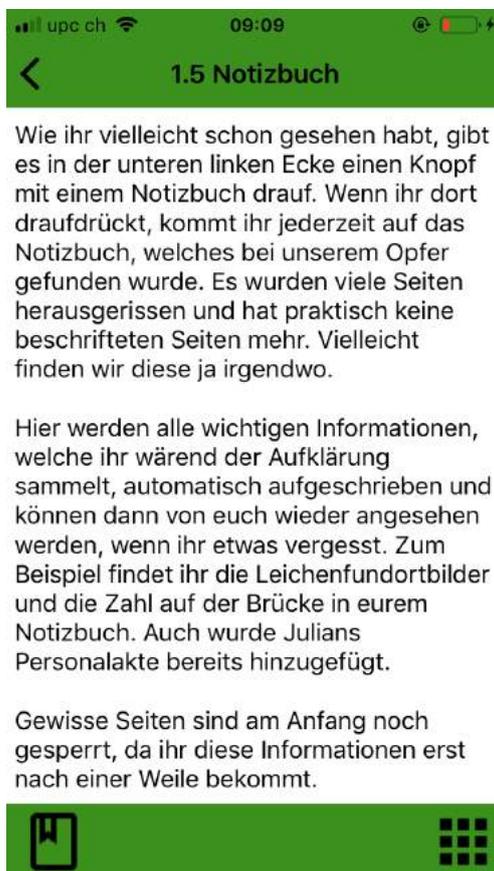
Am besten gehen wir mal zum Fundort der Leiche. Folget wieder eurem Navigationspfeil bis zur Sihl und geht auf die nächste Seite.





Henry: Ich hoffe, ihr habt das Angesicht der Leiche nicht all zu schlimm gefunden. Wir haben in der Hosentasche unseres Opfers ein Foto seiner Frau mit einer Widmung auf der Rückseite und einen kleinen Papierzettel gefunden. Eventuell könnte dieser wichtig sein, da er aus dem gleichen Papier besteht, wie das Notizbuch.

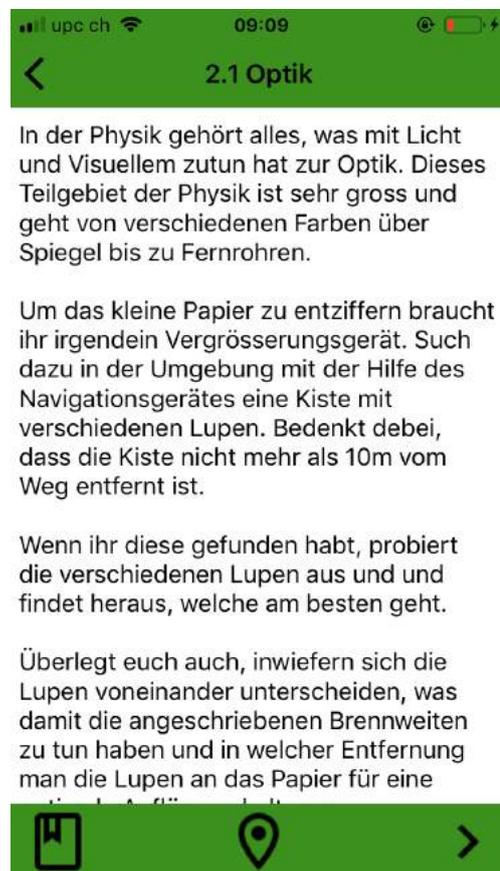
Wir vermuten, dass der Zettel ein Teil einer herausgerissenen Seite sein könnte. Das



Wie ihr vielleicht schon gesehen habt, gibt es in der unteren linken Ecke einen Knopf mit einem Notizbuch drauf. Wenn ihr dort draufdrückt, kommt ihr jederzeit auf das Notizbuch, welches bei unserem Opfer gefunden wurde. Es wurden viele Seiten herausgerissen und hat praktisch keine beschrifteten Seiten mehr. Vielleicht finden wir diese ja irgendwo.

Hier werden alle wichtigen Informationen, welche ihr während der Aufklärung sammelt, automatisch aufgeschrieben und können dann von euch wieder angesehen werden, wenn ihr etwas vergesst. Zum Beispiel findet ihr die Leichenfundortbilder und die Zahl auf der Brücke in eurem Notizbuch. Auch wurde Julians Personalakte bereits hinzugefügt.

Gewisse Seiten sind am Anfang noch gesperrt, da ihr diese Informationen erst nach einer Weile bekommt.

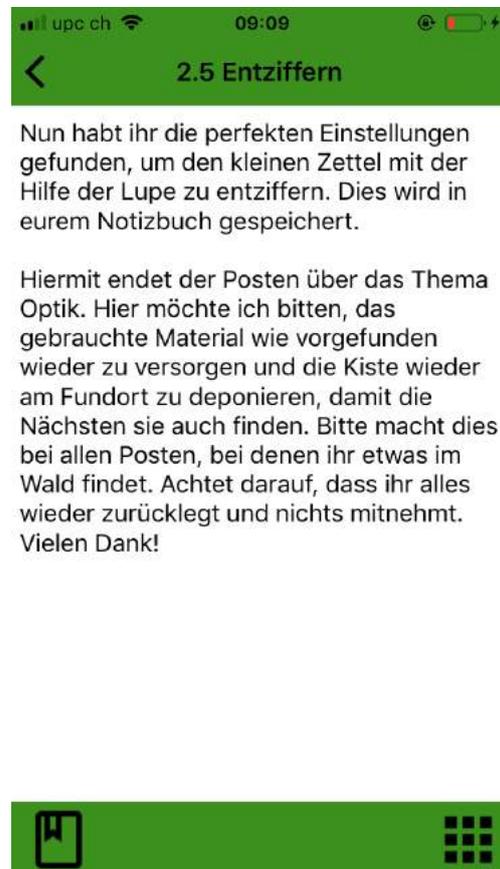
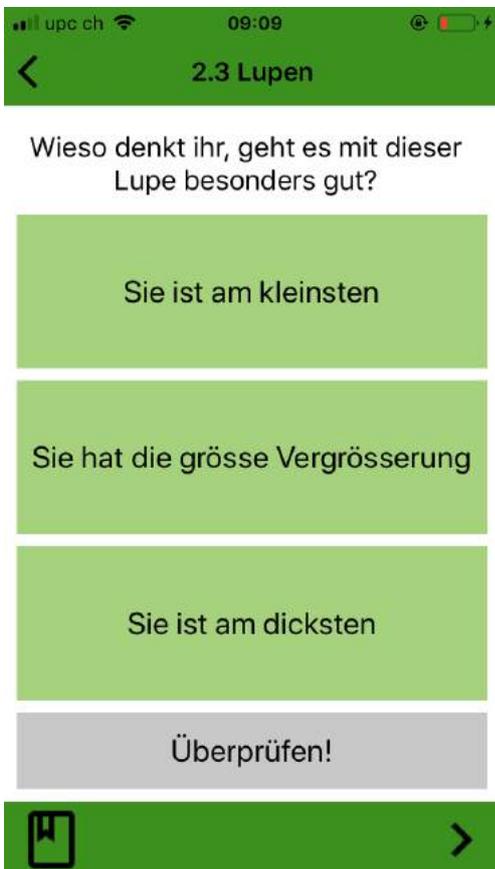
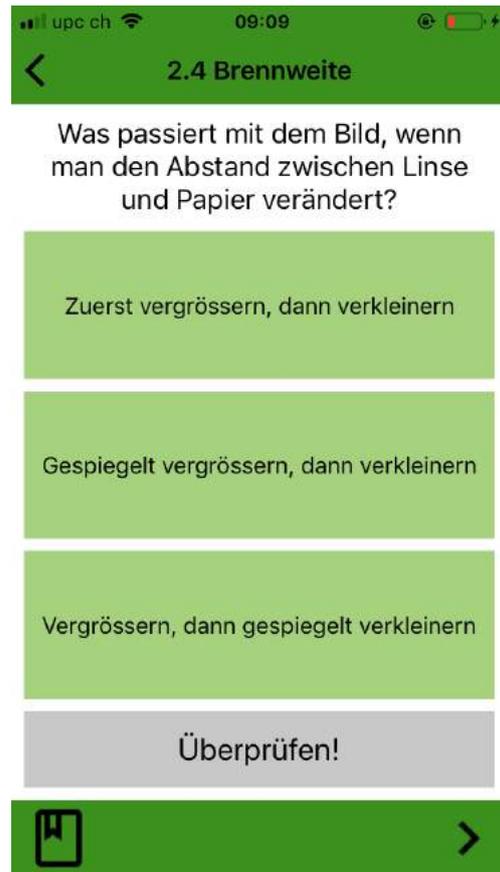
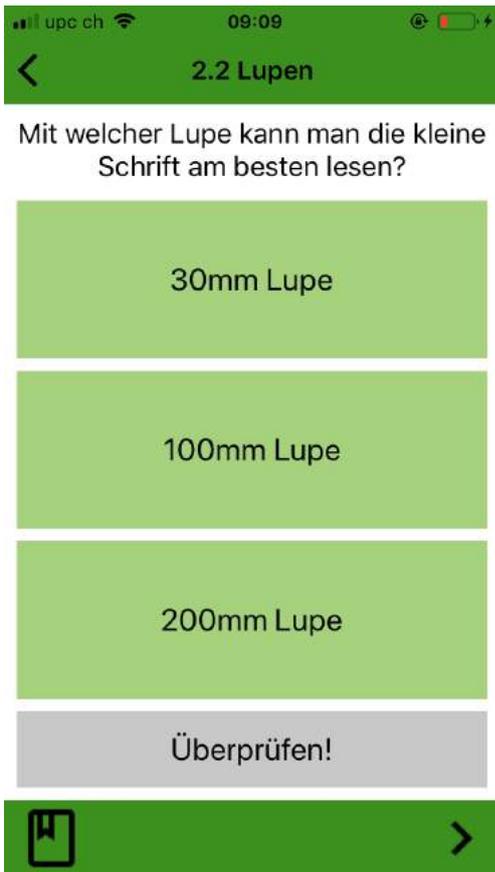


In der Physik gehört alles, was mit Licht und Visuellem zutun hat zur Optik. Dieses Teilgebiet der Physik ist sehr gross und geht von verschiedenen Farben über Spiegel bis zu Fernrohren.

Um das kleine Papier zu entziffern braucht ihr irgendein Vergrößerungsgerät. Sucht dazu in der Umgebung mit der Hilfe des Navigationsgerätes eine Kiste mit verschiedenen Lupen. Bedenkt dabei, dass die Kiste nicht mehr als 10m vom Weg entfernt ist.

Wenn ihr diese gefunden habt, probiert die verschiedenen Lupen aus und findet heraus, welche am besten geht.

Überlegt euch auch, inwiefern sich die Lupen voneinander unterscheiden, was damit die angeschriebenen Brennweiten zu tun haben und in welcher Entfernung man die Lupen an das Papier für eine





An dieser dritten Station ist vor allem Teamarbeit wichtig. Denn obwohl ihr gerade der Polizei erfolgreich geholfen habt, trauen sie euch immernoch nicht 100% -ig. Deswegen müsst ihr euch beweisen, damit die Polizei euch alle Informationen zugänglich macht. Deshalb wollt ihr 1. eure Zusammenarbeit und 2. euer physikalisches Wissen beweisen.

Geht deshalb bis kurz vor die Brücke zurück und schlagt dann nach rechts den Erlebnispfadweg ein. Geht den Weg bis ganz oben hinauf und folgt auf der Strasse wieder dem Navigationspfeil, um zum richtigen Ort zu kommen.



Um eure Teamarbeit zu stärken gibt es hier eine kleine Aufgabe für euch: Stellt euch alle in zwei Reihen auf, sodass man sich immer versetzt gegenübersteht. Nehmt einen der Stöcke vom Walderlebnispfadposten des Gleichgewichts (möglichst leicht und lang) und positioniert den auf euren Fingern. Achtung: Der Stock wird nur von je zwei Zeigefingern pro Person getragen, welche immer in Kontakt mit dem Stock sein müssen. Schaut euch auch das unten



Nun kommen wir zum eigentlichen Experiment. Dies hat mit dem physikalischen Thema Akustik zu tun. Davon habt ihr sicherlich schon alle einmal gehört. Dies beschäftigt sich mit allem, was mit dem Gehör zutun hat. Das Wort Schall spielt hier auch eine wichtige Rolle. Durch dieses Experiment wollen wir ein paar Eigenschaften vom Schall herausfinden.

Geht dazu zum nächsten Punkt von eurem Navigationsgerät. Ih solltet an einer Wegbiegung landen.

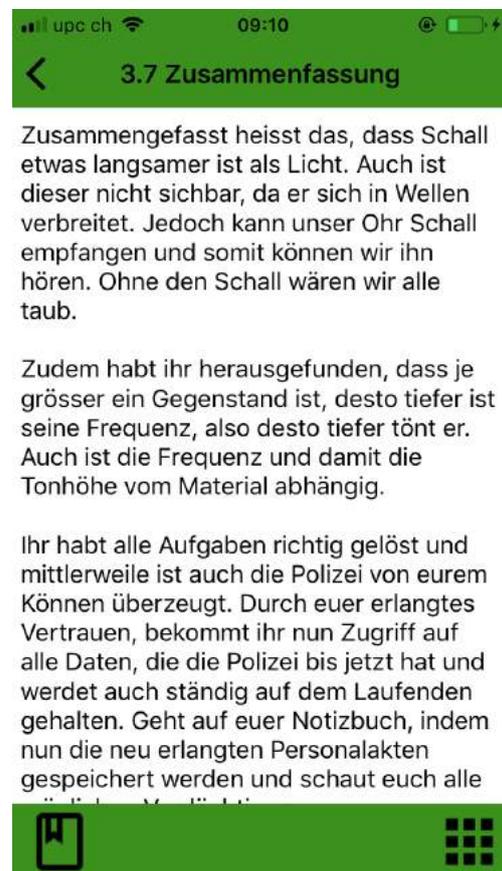
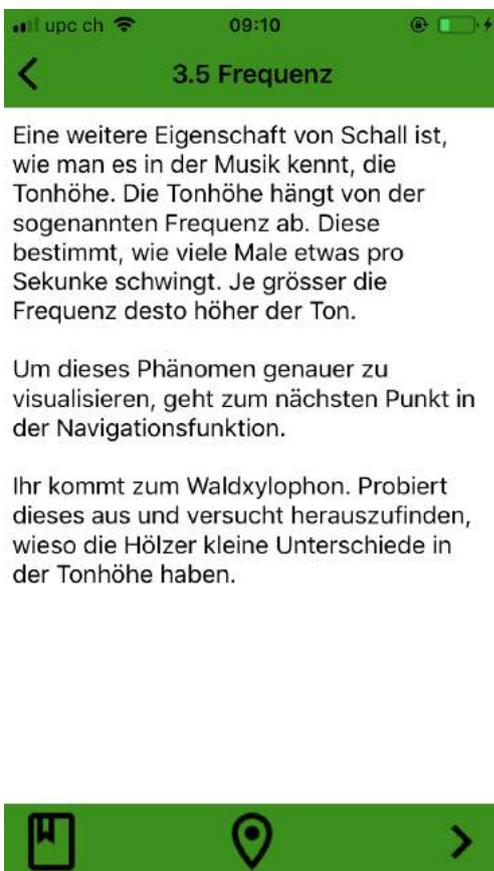
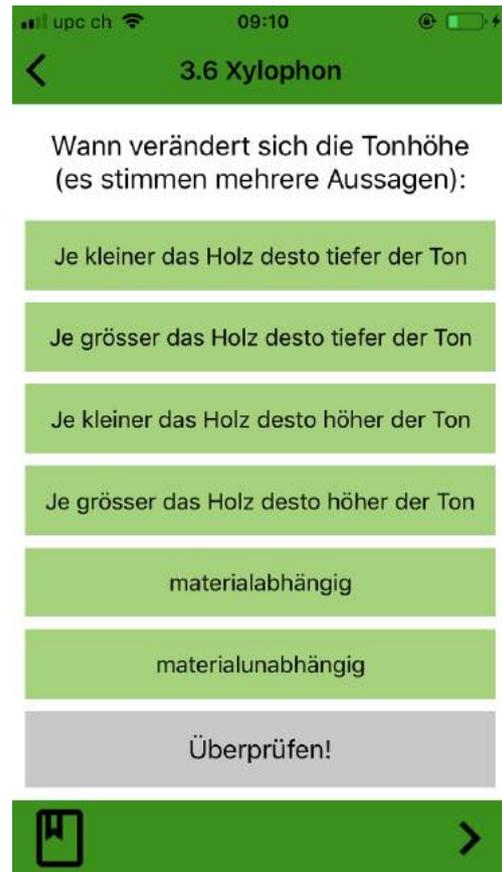


Zuerst werden wir den Schall in Zusammenhang mit Licht beobachten.

Dazu bleibt mindestens eine Person bei der Wegbiegung stehen, während sich die anderen um mindestens 80 Meter entfernen, wenn möglich sogar noch mehr. Dafür gehen diese den Weg weiter hinauf. Schliesslich klatscht die Person an der Wegbiegung über dem Kopf fest in die Hände und wiederholt dies ein paar mal.

Was fällt den anderen (Beobachtern) auf, wenn man genau hinschaut und hinhört?

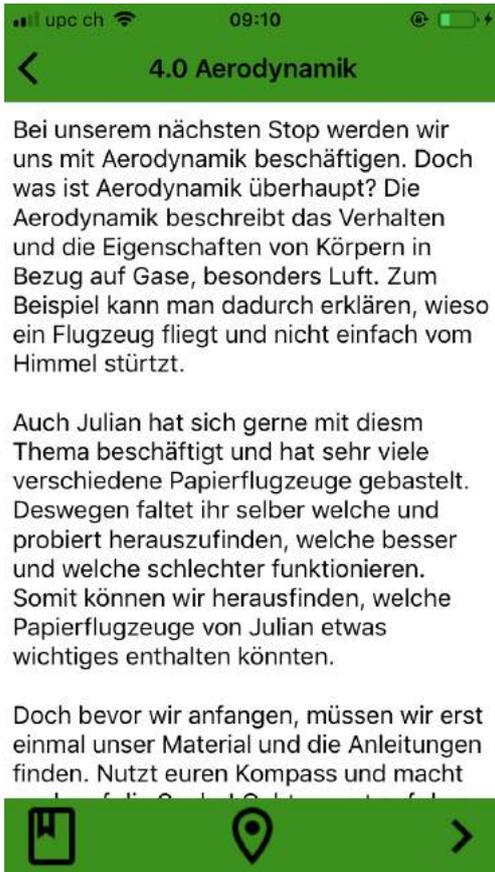




Bei unserem nächsten Stop werden wir uns mit Aerodynamik beschäftigen. Doch was ist Aerodynamik überhaupt? Die Aerodynamik beschreibt das Verhalten und die Eigenschaften von Körpern in Bezug auf Gase, besonders Luft. Zum Beispiel kann man dadurch erklären, wieso ein Flugzeug fliegt und nicht einfach vom Himmel stürzt.

Auch Julian hat sich gerne mit diesem Thema beschäftigt und hat sehr viele verschiedene Papierflugzeuge gebastelt. Deswegen faltet ihr selber welche und probiert herauszufinden, welche besser und welche schlechter funktionieren. Somit können wir herausfinden, welche Papierflugzeuge von Julian etwas wichtiges enthalten könnten.

Doch bevor wir anfangen, müssen wir erst einmal unser Material und die Anleitungen finden. Nutzt euren Kompass und macht



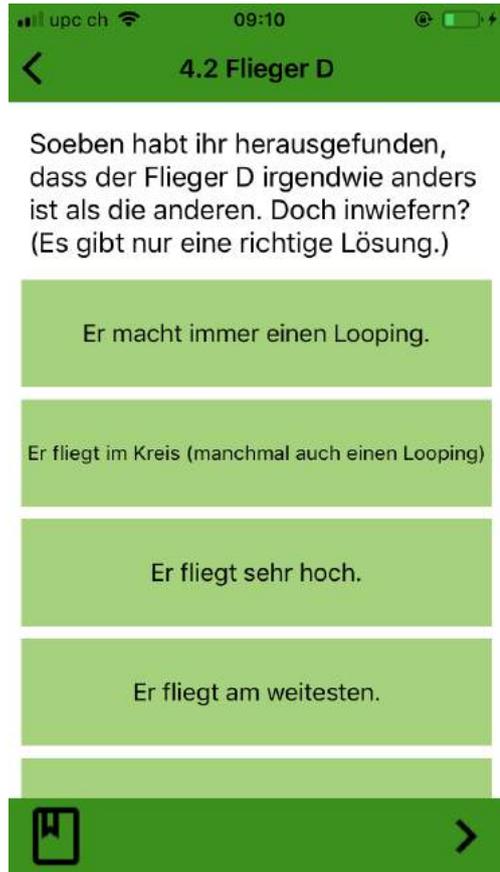
Soeben habt ihr herausgefunden, dass der Flieger D irgendwie anders ist als die anderen. Doch inwiefern? (Es gibt nur eine richtige Lösung.)

Er macht immer einen Looping.

Er fliegt im Kreis (manchmal auch einen Looping)

Er fliegt sehr hoch.

Er fliegt am weitesten.



Bei welchem eurer Papierflieger fällt euch etwas spezielles auf?

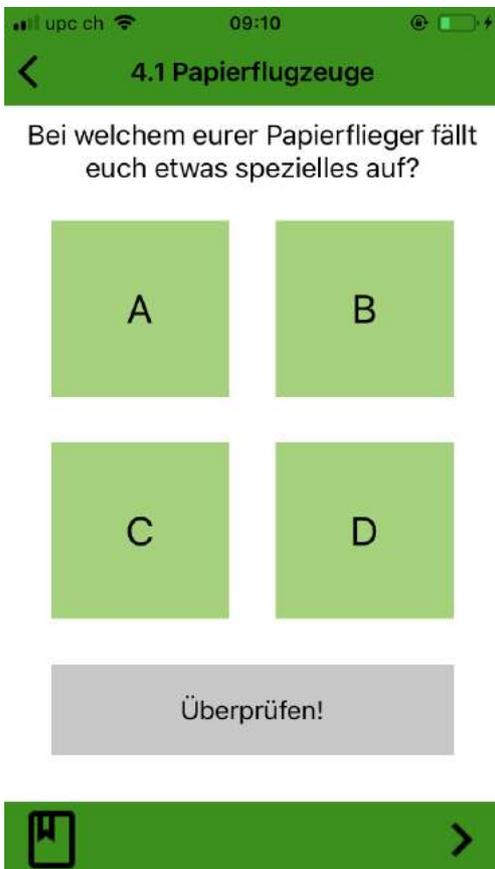
A

B

C

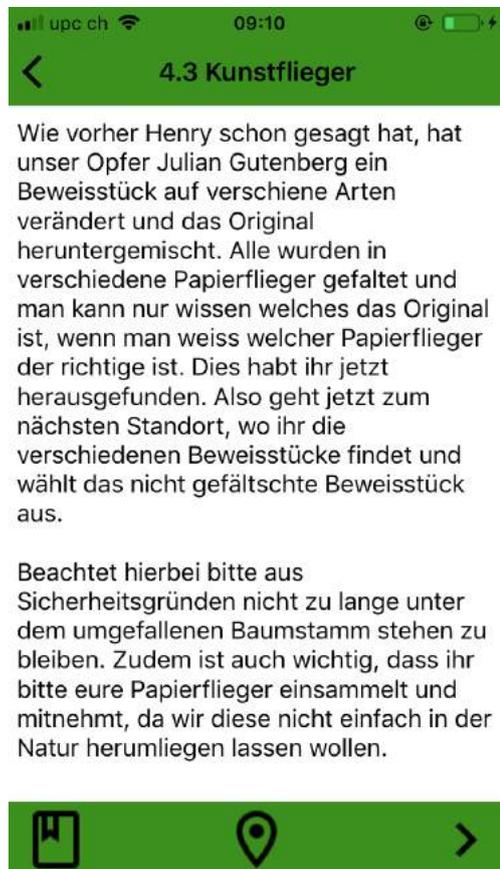
D

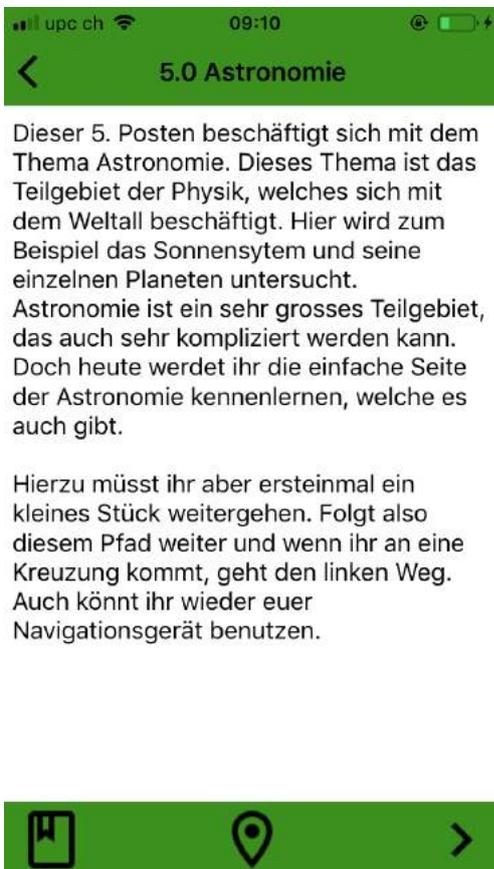
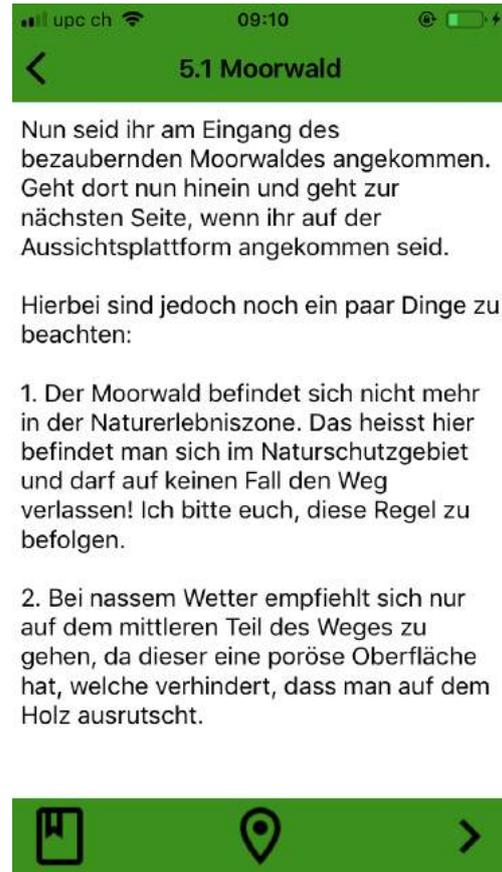
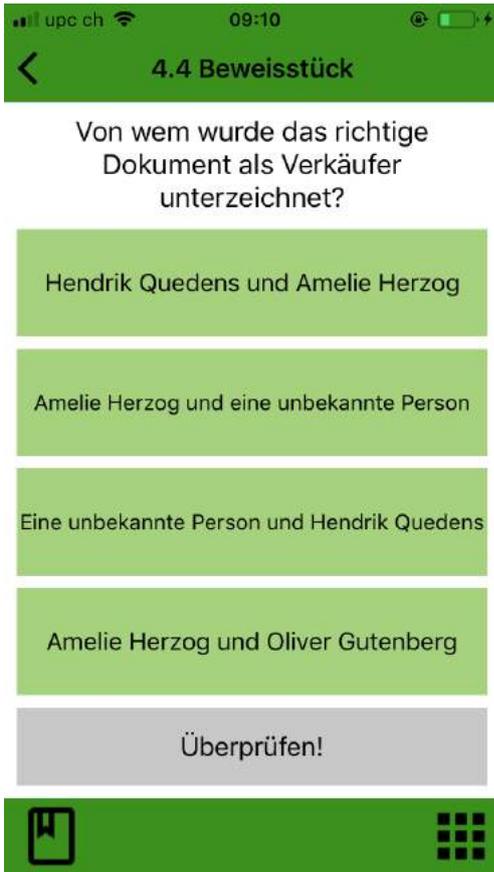
Überprüfen!



Wie vorher Henry schon gesagt hat, hat unser Opfer Julian Gutenberg ein Beweisstück auf verschiedene Arten verändert und das Original heruntergemischt. Alle wurden in verschiedene Papierflieger gefaltet und man kann nur wissen welches das Original ist, wenn man weiss welcher Papierflieger der richtige ist. Dies habt ihr jetzt herausgefunden. Also geht jetzt zum nächsten Standort, wo ihr die verschiedenen Beweisstücke findet und wählt das nicht gefälschte Beweisstück aus.

Beachtet hierbei bitte aus Sicherheitsgründen nicht zu lange unter dem umgefallenen Baumstamm stehen zu bleiben. Zudem ist auch wichtig, dass ihr bitte eure Papierflieger einsammelt und mitnehmt, da wir diese nicht einfach in der Natur herumliegen lassen wollen.





upc ch 09:10

### 5.3 Tatzeitpunkt

Um wieviel Uhr wurde der Schuss gelöst? Und wo ist in der Abbildung der Süden?

Überprüfen!

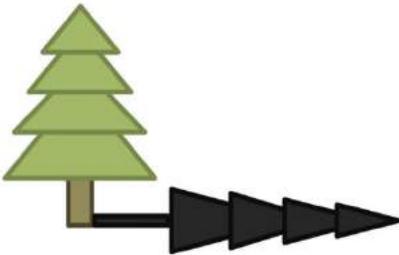


Tabelle zur Tatzeit

Winkel	Uhrzeit

upc ch 09:10

### 5.4 Süden?

Oben

Unten

Rechts

Links

Vorne

Hinten

Überprüfen!

upc ch 09:10

### 5.5 Alibis

Da man jetzt aufgrund eurer Hilfe weiss, wann das Opfer erschossen wurde, konnte man die einzelnen Verdächtigen nach einem Alibi fragen. Die Personalakten in eurem Notizbuch werden nun mit den entsprechenden Aussagen ergänzt.

Die Polizei hat mittlerweile mit der Ehefrau Emma Gutenberg gesprochen. Sie haben jedoch nicht so viel herausgefunden. Nur ist bekannt geworden, dass Emma vor dem Unfall des Ehepaars an Quantencomputern geforscht hat. Ihr Mann Julian hat, als sie im Koma lag, ihre theoretischen Berechnungen genommen und ohne ihre Erlaubnis umzusetzen. Jedoch empfand das Emma als nicht so schlimm, da es ein grosser Verlust für die Wissenschaft gewesen wäre, wenn ihr Quantencomputer einfach vergessen gegangen wäre.

An diesem Punkt möchte ich euch

upc ch 09:11

### 6.0 Quantencomputer

In den vorherigen Posten habt ihr herausgefunden, dass Julian Gutenberg und Amelie Herzog zusammen an einem Quantencomputer gearbeitet haben. Dies ist eines der neusten Forschungsprojekte. Da diese in unserm Krimifall eine bedeutende Rolle spielen, wollen wir uns in diesem Posten etwas genauer mit diesem Thema befassen und die Grundzüge verstehen.

Um weiterzufahren, folgt dem Naturerlebnispfad bis zum nächsten Punkt im Navigationsgerät. Geht dafür ein grosses Stück weiter, bis zur Walderlebnisstation der Baumringe.



Ein Quantencomputer ist ein Hochleistungsrechner. Im Gegensatz zu einem normalen Computer arbeitet ein Quantencomputer nicht nach normalen physikalischen Gesetzen. Auch werden die normalen Bits (Rechen- und Speichereinheit) bei Quantencomputern zu Qubits, deren Rechenleistung viel grösser sind. Dadurch kann man Probleme, welche exponentiell komplexer werden, besser lösen und sehr genaue Annäherungen machen.

Hier ein Beispiel im Vergleich zum Menschen, der wie ein normaler Computer funktioniert:  
Stellt euch vor man hat vier Spielkarten vor sich verdeckt liegen. Man weiss, dass unter einer Karte ein König versteckt ist. Wenn man jetzt herausfinden will, unter welcher Karte sich der König versteckt, dann bleibt einem nichts anderes übrig, als eine Karte nach der anderen



Diese grosse Leistung wird besonders bei verschlüsselten Übertragungen benötigt. Deswegen wird auch sehr fleissig daran geforscht, einen Quantencomputer zu entwickeln.

Solche Verschlüsselungen funktionieren oft durch die Primfaktorzerlegung, welche exponentiell immer schwerer wird zu lösen.

Anscheinend hat unser Opfer Julian es geliebt zu verschlüsseln, denn die Polizei hat hier, wo ihr jetzt seid, Seiten aus seinem Notizbuch mit verschlüsselten Nachrichten gefunden.

Findet die verschlüsselten Seiten und probiert die Zahlen darauf in ihre Primzahlen zu zerlegen, damit unsere Verschlüsselungsprofis mithilfe der Primzahlen die Nachrichten entschlüsseln können. Beachtet, dass das versteckte



Welche Zahlen stehen auf den gefundenen Blättern drauf?

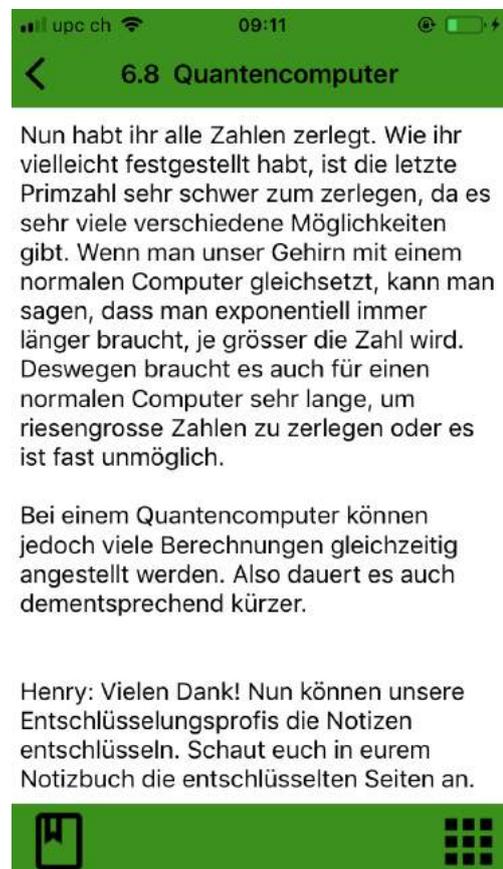
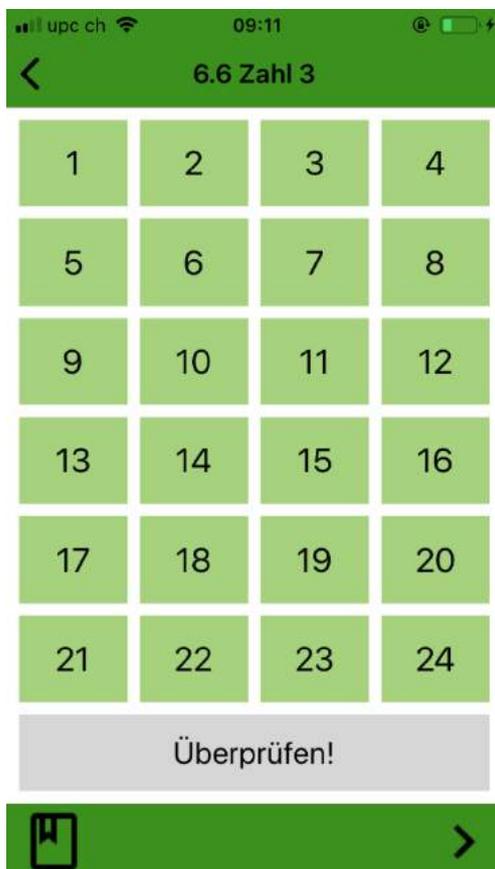
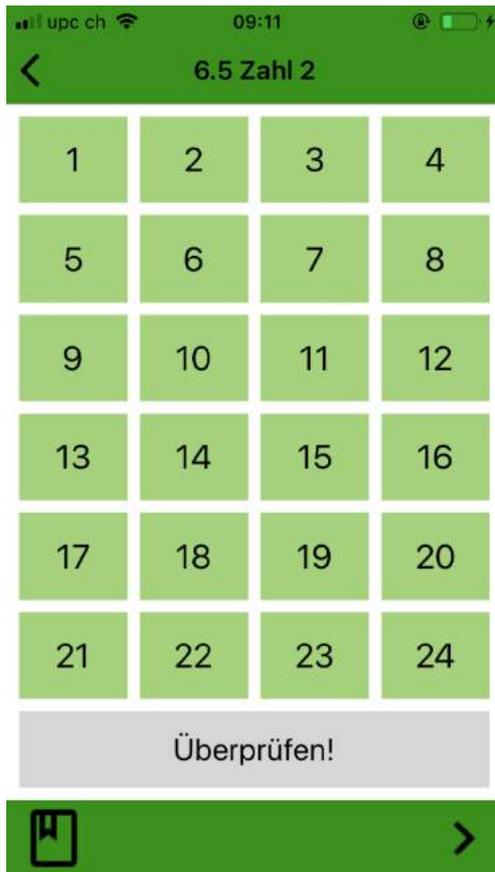
Zerlegt nun die Zahlen in ihre Primzahlen. Am besten schreibt ihr diese irgendwo auf. Wenn ihr alle zerlegt habt, könnt ihr diese auf den nächsten Seiten korrigieren. Pro Zahl gibt es eine Seite. Wählt aus den verschiedenen Zahlen die richtigen aus und kontrolliert, ob sie richtig sind.



1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16
17	18	19	20
21	22	23	24

Überprüfen!







Henry: Die Spurensicherung hat gerade in der Nähe der Strasse einen Fussabdruck im Boden gefunden, der zeitlich her möglicherweise vom Täter stammen könnte.

Der Abdruck stammt von einem Schuh mit gutem Profil, vielleicht einem Winterschuh.

Die Schuhgrösse 39 konnte schon bestimmt werden, doch ich glaube wir könnten eure Hilfe bei der Bestimmung des Gewichtes benötigen.

Das hat doch irgendetwas mit dem Druck oder so zu tun, nicht?

Am besten ihr geht gleich mal hin und schaut euch den Abdruck selber an.



Drucke, vor allem Schweredrucke, entstehen immer durch eine Kraft, die irgendwo gedrückt.

Bei einem Fussabdruck ist nicht nur die Gewichtskraft, die auf die Unterlage drückt, verantwortlich, wie der tief Abdruck wird, sondern auch das Material der Unterlage. Wenn man zum Beispiel auf eine normale Steinplatte steht, kann man ewig lange darauf stehen bleiben, ohne dass es einen Abdruck gibt. Denn Stein ist ein sehr hartes Material.

Bei Schlamm oder Waldboden sieht das jedoch schon ganz anders aus. Weicher Boden probiert dem Gewichtsdruck standzuhalten, gibt dann jedoch nach und die Teilchen müssen Platz machen und auf die Seite gehen. So entstehen die Vertiefungen, welche zusammengesetzt einen Schuhabdruck ergeben.



Durch die Analyse des Bodens konnte die unten abgebildete Tabelle erstellt werden. Diese gibt die Masse der Person im Vergleich zur Tiefe des Eindruckes an der tiefsten Stelle an.

Schätzt etwa, wie tief der Abdruck maximal ist und findet somit das Gewicht des möglichen Täters heraus.

**Tabelle zum Schuhabdruck**

Vertiefung	Gewicht
1 cm	50 - 60 kg
3 cm	60 - 70 kg
5 cm	70 - 80 kg
7 cm	80 - 90 kg



Was ist die Masse des potentiellen Mörders?

50 - 60 kg

60 - 70 kg

70 - 80 kg

80 - 90 kg

Überprüfen!





Henry: Ich habe noch drei weitere neue Infos für euch.

1. Wisst ihr noch, dass die Leiche nur mit einem Schuh gefunden wurde? Wir haben nun den zweiten gefunden. Er lag angeschwemmt am Ufer der Sihl. Leider hat uns dieser Schuh nicht viel weitergeholfen, aber ich habe gedacht, vielleicht interessiert es euch ja.



Henry: Wir vermuten, dass Julian hier erschossen wurde und dann von der Brücke hinunter in die Sihl gestossen wurde. Der Bericht der Ballistik sagt, dass unser Opfer mit einem .45 ACP Kaliber getroffen wurde. Danach muss er sich an die Brust gefasst haben und sich schliesslich mit der blutigen Hand am Geländer abgestützt haben. So sind die verschmierten Blutspuren vermutlich entstanden.

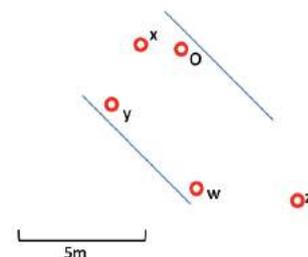


2. Endlich haben wir den Beschluss für die Hausdurchsuchung bekommen und ein Team war schon vor Ort. Das spannendste war, dass der Fussabdruck, welcher im Wald gefunden wurde, im Haus von unser Spurensicherung auch aufgenommen wurde.

Und schliesslich 3. Wir haben nun endlich den endgültigen Tatort gefunden, indem Julian Gutenberg erschossen wurde. Am besten ihr beendet sogleich diesen Posten und kommt mit zum Tatort.

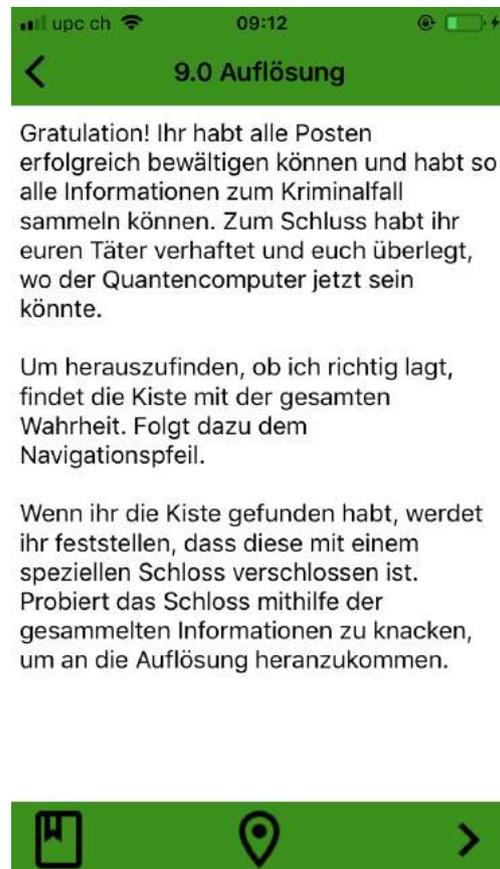
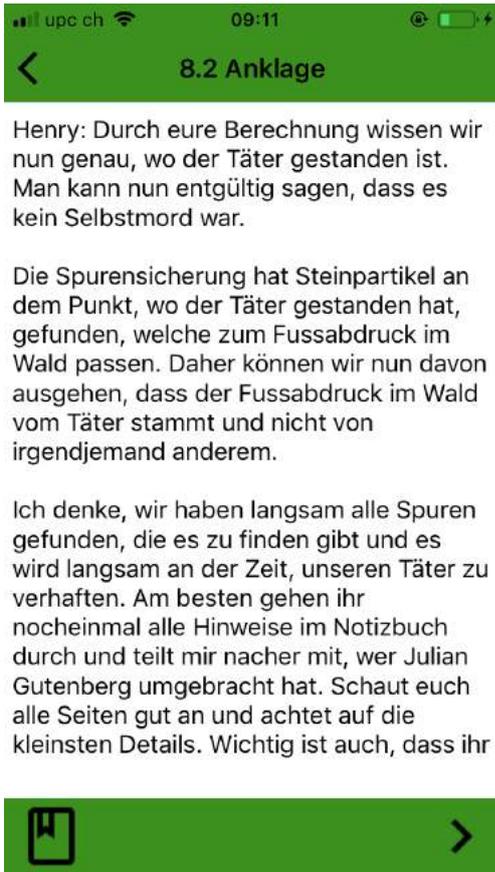


Mit dem Thema Kinematik werdet ihr euch als nächstes und letztes beschäftigen. Kinematik ist die Bewegungslehre. Das heisst sie beschäftigt sich mit allen, sich bewegenden Objekten. So kommt sie auch bei Schusswaffen vor. Ein .45 ACP Kaliber hat eine



W	X	Y	Z
Überprüfen!			







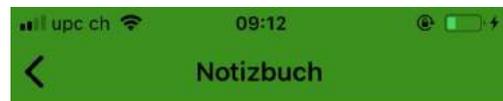
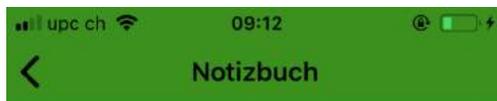
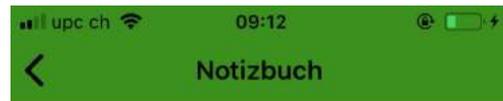
Mit der Auflösung des Falles, kommt ihr ans Ende dieser App. Ihr habt nun alle Rätsel gelöst.

Ich hoffe, ihr hattet viel Spass und habt etwas auf eurem Spaziergang gelernt.

Ich würde mich sehr freuen, über ein kurzes Feedback zu meiner Arbeit auf die Mailadresse [dasnotizbuch@gmx.ch](mailto:dasnotizbuch@gmx.ch), falls jemand Lust hätte, mir dabei zu helfen, diese App zu verbessern. Vielen Dank!

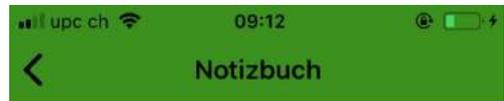
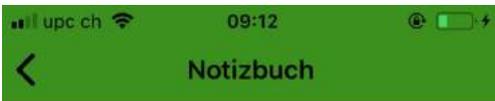
Nun wünsche ich euch einen schönen Nachhauseweg.

(Die Züge nach Zürich fahren jeweils einmal in der Stunde um \*\*:45 Uhr)

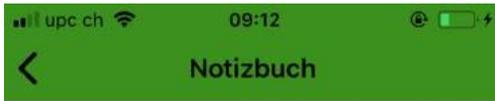


$x = 28$



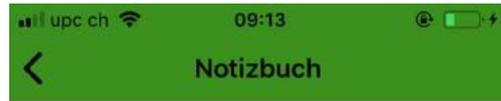






Für meinen  
geliebten  
Julian!

In Liebe ♥  
deine Emma



### Personalakte Verdächtige

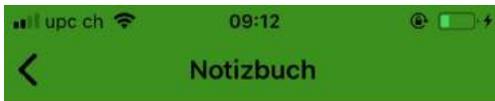


Name: Herzog  
Vorname: Amelie  
Geburtsdatum: 12.11.1989  
Geburtsort: Zürich  
Nationalität: Schweiz  
Beruf: Physikerin bei Omnotitia  
Körpergröße: 172cm

Schuhgröße: 39-40 Gewicht: 63kg  
Augenfarbe: blau Haarfarbe: blond  
Familienmitglieder: ledig, keine von Interesse  
Spezielles: Wollte Juliana und ihren QC an Teleshophaus verkaufen, Affäre mit Juliana nachdem Emma wieder nach Rückzug von Julian

Verdacht auf: Mord an Julian Gutenberg  
Beweise: Behauptung mit Verkauf von QC an Teleshophaus  
Fussabdruck könnte passen  
Alibi: mit freigeschrieben, konnte gar nicht sein

Schriftprobe:  
*Emma hat im komplett verwirrtsten Taxi quer durch Bayern, während zwölf Boxkämpfer Viktor quer über den ganzen grossen Silber Döck jagen.*  
Unterschrift: *Amelie Herzog*



### Personalakte Opfer



Name: Gutenberg  
Vorname: Julian  
Geburtsdatum: 27.1.1984  
Geburtsort: Zürich  
Nationalität: Schweiz  
Beruf: Physiker bei Omnotitia

Körpergröße: 178cm Schuhgröße: 45 Gewicht: 74kg  
Augenfarbe: blau Haarfarbe: graubraun

Familienmitglieder: verheiratet mit Emma, keine Kinder

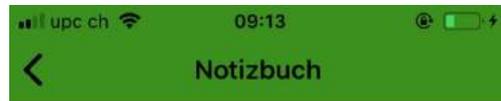
Todesursache: Schuss in die Brust mit .45ACP Kaliber,

danach tiefer Sturz und mehrere Stunden im Wasser bis Fund

Spezielles: Fund eines Notizbuchs bei Leichenfundort, Leiche hatte nur einen

Schick beim Fund an, in Entwicklungsprozess von Quantencomputer, Affäre mit

Amelie Herzog



### Personalakte Verdächtige



Name: Gutenberg  
Vorname: Emma  
Geburtsdatum: 6.8.1988  
Geburtsort: Hamburg  
Nationalität: Schweiz, Deutsch  
Beruf: Physikerin bei Omnotitia  
Körpergröße: 169cm

Schuhgröße: 38-39 Gewicht: 59kg  
Augenfarbe: blau Haarfarbe: blond

Familienmitglieder: Ehemann Julian, keine Kinder

Spezielles: Nach Unfall mit Ehemann am Steper 2 Jahre im Koma, vor 2

Monaten wieder aufgewacht, theoretischer QC eigentlich von ihr, Mann hat ihn

„gestohlen“, nicht schlüssig, da nur für Wissenschaft

Verdacht auf: Mord an Julian Gutenberg

Beweise: Fussabdruck könnte passen, Abdruck in ihrem Haus auch gefunden

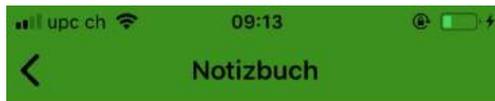
Alibi: Telefonat mit Praktikant in der Firma, sowie Telefonat nachgeholfen

Schriftprobe:

*Frage hat im komplett verwirrtsten Taxi quer durch Bayern, während zwölf Boxkämpfer Viktor quer über den ganzen grossen Silber Döck jagen.*

Unterschrift: *Emma Gutenberg*





**Personalakte Verdächtige**



Name: Gutenberg  
 Vorname: Oliver  
 Geburtsdatum: 5.2.1991  
 Geburtsort: Zürich  
 Nationalität: Schweiz  
 Beruf: Bobfahrer an der ETH Zürich  
 Körpergröße: 181cm

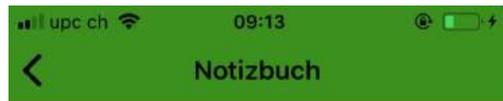
Schuhgröße: 45-46 Gewicht: 69kg  
 Augenfarbe: braun Haarfarbe: braun  
 Familienmitglieder: Bruder von Julian, ledig  
 Spezielles: ist Leiter der Terroristenfirma Techgeniosus, wollte somit QC kaufen

Verdacht auf: Mord an Julian Gutenberg  
 Beweise: Fussabdruck passt nicht

Alibi: war zuhause ein Buch am lesen  
 Schriftprobe:

*Frank jagt im komplett verrückten Taxi quer durch Bayern während zwölf Bombenplaner Viktor quer über die ganze, grossen Sylter Bül jagt.*

Unterschrift: *O. Gutenberg*



**Kaufvertrag**

zwischen  
 Verkäufer: Amelie Herzog und [redacted]  
 und  
 Käufer: Techgeniosus

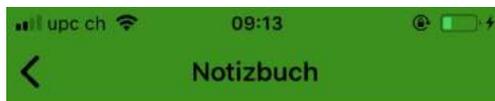
- §1. Die Verkäufer veräußern den von ihnen entwickelten und erbauten Quantencomputer QCA100-1.
- §2. Der Kaufpreis in Höhe von 125 Mio Dollar wird im Zeitraum von fünf Monaten in drei Teilen gezahlt.
- §3. Die Parteien vereinbaren eine absolute Schweigepflicht über den Kaufgegenstand, diesen Vertrag und alle damit in Zusammenhang stehenden Umstände.
- §4. Die weiteren Abwicklungsmodalitäten wie Lieferart, Lieferzeitpunkt, Zahlung und Haftung regelt ein und ein weiterer Vertrag.
- §5. Ein Rücktritt vom Vertrag ist ausdrücklich ausgeschlossen.

Verkäufer 1: *Amelie Herzog*

Verkäufer 2: [redacted]

Käufer:

Zürich, 25 September 2018



**Personalakte Verdächtige**



Name: Quaden  
 Vorname: Hendrik  
 Geburtsdatum: 25.6.1968  
 Geburtsort: Wijk aan Zee  
 Nationalität: Schweiz, Deutsch  
 Beruf: Physik Abteilungsleiter bei CERN  
 Körpergröße: 183cm

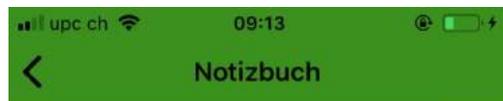
Schuhgröße: 48 Gewicht: 89kg  
 Augenfarbe: blau Haarfarbe: blau  
 Familienmitglieder: Chef von Julian, Ehefrau und 3 Töchter  
 Spezielles: hat von Verbleib von QC mitbekommen und will diesen verhindern, möchte so selber haben und dafür Geld und Leuten anfordern

Verdacht auf: Mord an Julian Gutenberg  
 Beweise: Fussabdruck passt nicht

Alibi: war mit Hund im Wald spazieren  
 Schriftprobe:

*Frank jagt im komplett verrückten Taxi quer durch Bayern, während zwölf Bombenplaner Viktor quer über die ganze, grossen Sylter Bül jagt.*

Unterschrift: *Hendrik Quaden*



*Julians Zahlen:*

*Mo: 1.21*

*Di: 2.78*

*Fr: 3.462*

*Sa: 4.8610*

*Blau:  $y = Mo \cdot 1 - 1$*



?Jr afxr im demlv V3mg@  
 :co <clqco Wk zco z  
 Gokrvitcpozyky l-|ocylt  
 m/h. kexqx c-sr afompix  
 8lhgv Efszgv4 ?Jr mot  
 wvi 'j' jrsp gwit0 z3wo  
 fl dshkwih Wds c-sr J -  
 vipx m5/0 Ziml [cx d] vqvo-  
 suA Mkr d'w l' Wop z\_x  
 fviiereo M qc >h l' z3z  
 [Jx lix cc] [c] copniociv  
 2a/l iigat j' rsp Cxwelkn  
 Kqgidvciv\$ ?Jr Xcx h'paw4  
 l'Wp -Jr p0, Fimlio JripC  
 z=Entfernung ~~der~~ ~~Spiter~~  
 durch 2

1

Mu b/lw - Gzu y\|Xqo| -  
 kv6 IG dkr:Urv X/lx  
 10 @Tuyoydo XS Hfzu  
 Jotkyky -G > @l mKz  
 nyk IG X mklmumx  
 not CK Mkh q' mKxp -  
 xjmlzd cx d[X]x 'Xuung  
 go 12qz [Gu dorluunwgg  
 -Gfuu8 & akjuu =G  
 | 1Rz  
 io 'l<IG K [K Hfz w]-  
 otkyysyky Ja j0|x  
 -Gzuu no  
 13 1S [Oz Clroxmx l]B  
 [G >k + q[X]x4

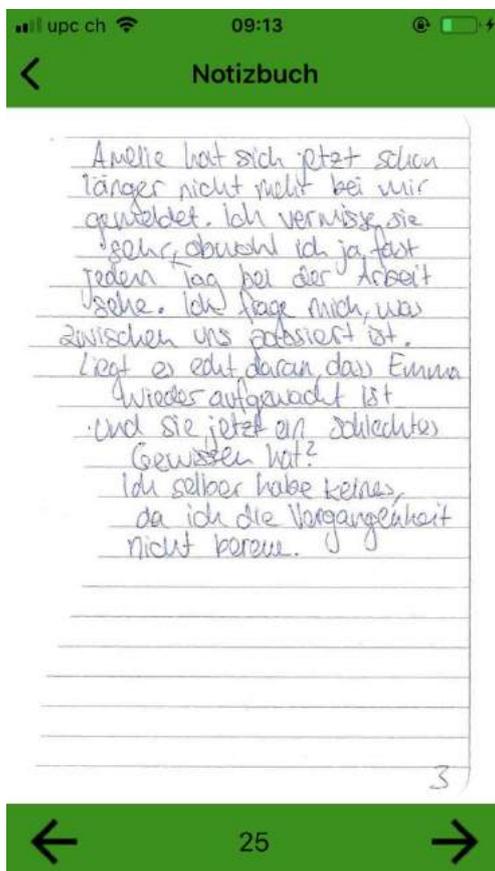
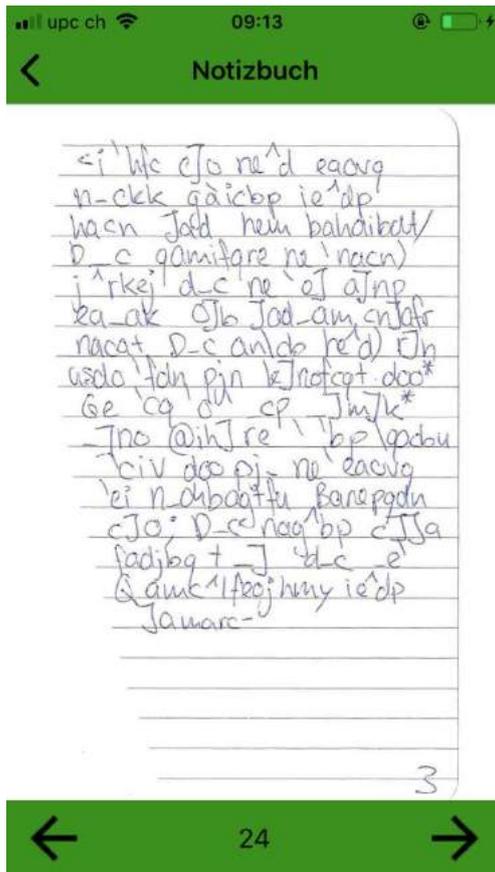
2

Ich kann es nicht fassen:  
 Die Firma, an die das Quantencomputer  
 verkauft wird, leitet mein kleiner  
 Bruder Oliver. Ich wusste ja schon  
 immer, dass er reichlich auf  
 mein Talent war, doch  
 ein Terrorist?  
 Wann kam er auf den  
 illegalen Weg?  
 Hat er das recht von mir?  
 Vielleicht hat er sogar  
 schon Menschen umgebracht!  
 Ich bin ratlos.  
 Soll ich zur Polizei gehen?

1

Vor kurzem habe ich  
 erfahren, dass mein Chef  
 Hendrik auch ein Interesse  
 am Quantencomputer  
 hat. Ich vermutete, dass er  
 versuchsweise den  
 Verkauf zu verhindern.  
 Er muss auch irgendeine  
 etwas davon mitbekommen  
 haben. Auf jeden Fall  
 glaube ich, dass er den  
 Quantencomputer für  
 sich haben will, um  
 die Lorbeeren und das  
 Geld zu ernten.

2





## 13.3. Dokumente der App

### 13.3.1. Personalakten

Jeder Charakter in meinem Mordfall hat eine eigene Personalakte. Diese ist am Anfang noch sehr leer und füllt sich dann im Verlauf meiner Krimi-App mit den Informationen, die man sammelt. Diese Personalakten kann man bei Bedarf im Notizbuch immer wieder anschauen. Hier angefügt sind jeweils die letzten, also kompletten Versionen der Personalakten, da es mir nicht sinnvoll erschien, die anderen auch noch zu integrieren, wenn deren Informationen sowieso in der Endversion enthalten sind.

---

## Personalakte Opfer



Name: Gutenberg

Vorname: Julian

Geburtsdatum: 27.1.1984

Geburtsort: Zürich

Nationalität: Schweiz

Beruf: Physiker bei Omnotitia

Körpergrösse: 178cm Schuhgrösse: 45 Gewicht: 74kg

Augenfarbe: blau Haarfarbe: graubraun

Familienmitglieder: verheiratet mit Emma, keine Kinder

Todesursache: Schuss in die Brust mit .45 ACP Kaliber,  
danach tiefer Sturz und mehrere Stunden im Wasser bis Fund

Spezielles: Fund eines Notizbuches bei Leichenfundort, Leiche hatte nur einen  
Schuh beim Fund an, in Entwicklungsprozess von Quantencomputer, Affäre mit  
Amelie Herzog

## Personalakte Verdächtige



Name: Herzog

Vorname: Amelie

Geburtsdatum: 12.11.1989

Geburtsort: Zürich

Nationalität: Schweiz

Beruf: Physikerin bei Omnotitia

Körpergrösse: 172cm

Schuhgrösse: 39-40 Gewicht: 63kg

Augenfarbe: blau Haarfarbe: blond

Familienmitglieder: Ledig, keine von Interesse

Spezielles: wollte Julians und ihren QC an Techgeniosus verkaufen; Affäre mit Julian; nachdem Emma wieder wach, Rückzug von Julian

Verdacht auf: Mord an Julian Gutenberg

Beweise: Dokument mit Verkauf von QC an Techgeniosus

Fussabdruck könnte passen

Alibi: Hat ferngesehen, konnte genau sagen was

Schriftprobe:

franz jagt im komplett verwalteten Saxi quer  
durch Bayern, während zwölf Boxkämpfer Viber  
quer über den pauen, posse Siles Reich jagt.

Unterschrift:

Amelie Herzog

## Personalakte Verdächtige



Name: Gutenberg

Vorname: Emma

Geburtsdatum: 6.8.1988

Geburtsort: Hamburg

Nationalität: Schweiz, Deutsch

Beruf: Physikerin bei Omnotitia

Körpergrösse: 169cm

Schuhgrösse: 38-39 Gewicht: 59kg

Augenfarbe: blau Haarfarbe: blond

Familienmitglieder: Ehemann Julian, keine Kinder

Spezielles: Nach Unfall mit Ehemann am Steuer 2 Jahre im Koma, vor 2

Monaten wieder aufgewacht, theoretischer QC eigentlich von ihr, Mann hat ihn

„gestohlen“, nicht schlimm, da gut für Wissenschaft

Verdacht auf: Mord an Julian Gutenberg

Beweise: Fussabdruck könnte passen, Abdruck in ihrem Haus auch gefunden

Alibi: Telefonat mit Praktikant in der Firma, 30min Telefonat nachgewiesen

Schriftprobe:

Franz jagt im komplett verwahrlosten Taxi quer  
durch Bayern, während zwölf Boxkämpfer Viktor quer  
über den ganzen, grossen Sylter Datch jagen.

Unterschrift: Emma Gutenberg

## Personalakte Verdächtige



Name: Gutenberg

Vorname: Oliver

Geburtsdatum: 5.2.1991

Geburtsort: Zürich

Nationalität: Schweiz

Beruf: Doktorand an der ETH Zürich

Körpergrösse: 181cm

Schuhgrösse: 45-46 Gewicht: 69kg

Augenfarbe: braun Haarfarbe: braun

Familienmitglieder: Bruder von Julian, ledig

Spezielles: Ist Leiter der Terroristenfirma Techgeniosus, wollte somit QC kaufen

Verdacht auf: Mord an Julian Gutenberg

Beweise: Fussabdruck passt nicht

Alibi: war Zuhause ein Buch am lesen

Schriftprobe:

Franz jast im komplett veruhrlosk Taxi quer  
durch Bayern, während zwölf Boxkämpfer Viktor  
quer über de ganzen, giossen Sylter Diel josen.

Unterschrift:

*O. Gutenberg*

## Personalakte Verdächtige



Name: Quedens

Vorname: Hendrik

Geburtsdatum: 25.6.1968

Geburtsort: Wyk auf Föhr

Nationalität: Schweiz, Deutsch

Beruf: Physik Abteilungsleiter bei Omnotitia

Körpergrösse: 189cm

Schuhgrösse: 48 Gewicht: 89kg

Augenfarbe: blau Haarfarbe: blau

Familienmitglieder: Chef von Julian, Ehefrau und 3 Töchter

Spezielles: Hat von Verkauf von QC mitbekommen und will diesen verhindern, möchte QC selber haben und dafür Geld und Lorbeeren ernten

Verdacht auf: Mord an Julian Gutenberg

Beweise: Fussabdruck passt nicht

Alibi: war mit Hund im Wald gassigehen

Schriftprobe:

Franz jagt im komplett verhalten Kosten Tätigwer  
durch Bayern, während zwölf Barkämpfer Viktor  
quer über den ganzen grossen Syden Reich jagen.

Unterschrift: Hendrik Quedens

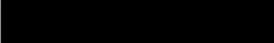
### 13.3.2. Kaufverträge (mit Fälschungen)

In meiner App gibt es einen Kaufvertrag, in dem bestätigt wird, dass ein Verkauf vom Quantencomputer geplant ist. Damit man nicht von Anfang an weiss, wer bei diesem Verkauf beteiligt ist, gibt es vier verschiedene Kaufverträge, unter denen einer der Richtige ist. Die anderen drei Exemplare sind Fälschungen. In der Reihenfolge, in der sie hier abgebildet sind, ist der erste Kaufvertrag der Richtige und die drei folgenden die Fälschungen.

---

## Kaufvertrag

zwischen

Verkäufer: Amelie Herzog und 

und

Käufer: Techgeniosus

§1. Die Verkäufer veräußern den von ihnen entwickelten und erbauten Quantencomputer QCA100-1.

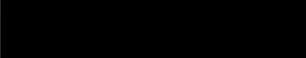
§2. Der Kaufpreis in Höhe von 125 Mio Dollar wird im Zeitraum von fünf Monaten in drei Teilen gezahlt.

§3. Die Parteien vereinbaren eine absolute Schweigepflicht über den Kaufgegenstand, diesen Vertrag und alle damit in Zusammenhang stehenden Umstände.

§4. Die weiteren Abwicklungsmodalitäten wie Lieferart, Lieferzeitpunkt, Zahlung und Haftung regelt ein und ein weiterer Vertrag.

§5. Ein Rücktritt vom Vertrag ist ausdrücklich ausgeschlossen.

Verkäufer 1: 

Verkäufer 2: 

Käufer:

Zürich, 25 September 2018

## Kaufvertrag

zwischen

Verkäufer: Hendrik Quedens und Amelie Gutenberg

und

Käufer: Techgeniosus

§1. Die Verkäufer veräußern den von ihnen entwickelten und erbauten Quantencomputer QCA100-1.

§2. Der Kaufpreis in Höhe von 125 Mio Dollar wird im Zeitraum von fünf Monaten in drei Teilen gezahlt.

§3. Die Parteien vereinbaren eine absolute Schweigepflicht über den Kaufgegenstand, diesen Vertrag und alle damit in Zusammenhang stehenden Umstände.

§4. Die weiteren Abwicklungsmodalitäten wie Lieferart, Lieferzeitpunkt, Zahlung und Haftung regelt ein und ein weiterer Vertrag.

§5. Ein Rücktritt vom Vertrag ist ausdrücklich ausgeschlossen.

Verkäufer 1: 

Verkäufer 2: 

Käufer:

Zürich, 25 September 2018

---

## Kaufvertrag

zwischen

Verkäufer: [REDACTED] und Hendrik Quedens

und

Käufer: Techgeniosus

§1. Die Verkäufer veräußern den von ihnen entwickelten und erbauten Quantencomputer QCA100-1.

§2. Der Kaufpreis in Höhe von 125 Mio Dollar wird im Zeitraum von fünf Monaten in drei Teilen gezahlt.

§3. Die Parteien vereinbaren eine absolute Schweigepflicht über den Kaufgegenstand, diesen Vertrag und alle damit in Zusammenhang stehenden Umstände.

§4. Die weiteren Abwicklungsmodalitäten wie Lieferart, Lieferzeitpunkt, Zahlung und Haftung regelt ein und ein weiterer Vertrag.

§5. Ein Rücktritt vom Vertrag ist ausdrücklich ausgeschlossen.

Verkäufer 1: [REDACTED]

Verkäufer 2: *Hendrik Quedens*

Käufer:

Zürich, 25 September 2018

## Kaufvertrag

zwischen

Verkäufer: Amelie Gutenberg und Oliver Gutenberg

und

Käufer: Techgeniosus

§1. Die Verkäufer veräußern den von ihnen entwickelten und erbauten Quantencomputer QCA100-1.

§2. Der Kaufpreis in Höhe von 125 Mio Dollar wird im Zeitraum von fünf Monaten in drei Teilen gezahlt.

§3. Die Parteien vereinbaren eine absolute Schweigepflicht über den Kaufgegenstand, diesen Vertrag und alle damit in Zusammenhang stehenden Umstände.

§4. Die weiteren Abwicklungsmodalitäten wie Lieferart, Lieferzeitpunkt, Zahlung und Haftung regelt ein und ein weiterer Vertrag.

§5. Ein Rücktritt vom Vertrag ist ausdrücklich ausgeschlossen.

Verkäufer 1: 

Verkäufer 2: 

Käufer:

Zürich, 25 September 2018

---

### 13.3.3. Papierflieger Anleitungen

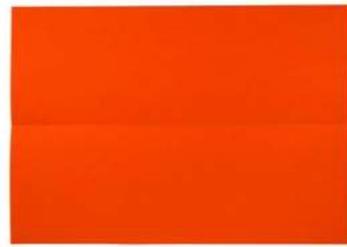
Bei dem Posten Aerodynamik werden vier verschiedene Papierflieger gebastelt. Dafür sucht man im Wald eine Dose mit den Materialien und den Faltanleitungen. Damit man nicht aus dem Namen des Fliegers schliessen kann, welche Eigenschaften der Papierflieger besitzt, habe ich die Namen weggenommen und Buchstaben zur Erkennung verteilt. Der Flieger A trägt eigentlich den Namen Pfeil.[34] Gleiter heisst der Papierflieger B.[35] Der Flieger C nennt sich Weltrekord.[36] Und der Papierflieger D heisst Kunstflieger.[37]

### Flieger A – Anleitung

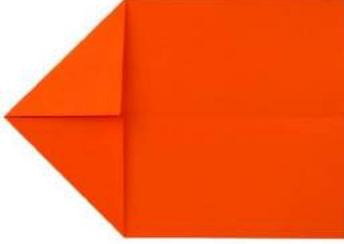
Lege das Papier hochkant vor dich hin.



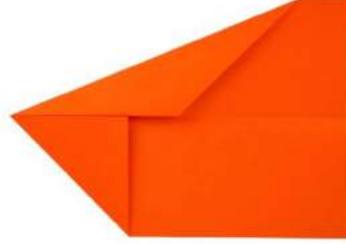
Falte das Papier nun in der Hälfte und klappe es anschließend wieder auf.

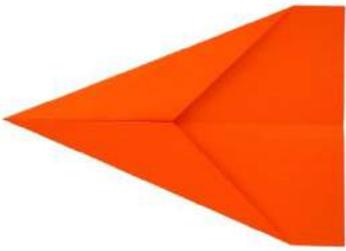


Daraufhin nimmst du die oberen Ecken und knickst diese zur Mittelfalz.

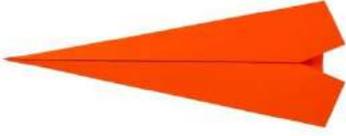


Falte die bereits gefalteten Kanten erneut bis zur Mitte.





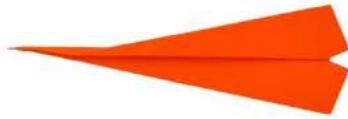
Im nächsten Schritt musst du die Kanten ein weiteres Mal zum Mittelteil falten.



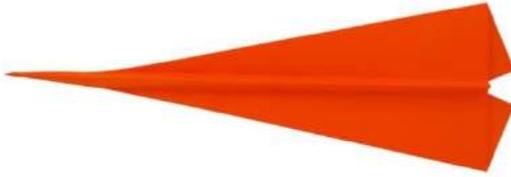
Falte den Papierflieger in der Hälfte.



Knicke nun die Flügel ab. Die Griffkante sollte etwa einen Zentimeter groß sein.



Klappe den Papierflieger auf und fertig ist er.

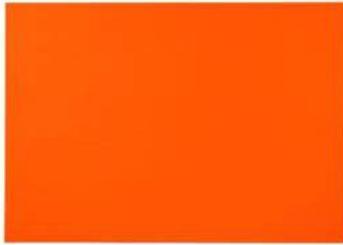


Werfe diesen Flieger mit mehr Kraft, als andere Papierflieger. Aufgrund der kleinen Tragflächen hat dieses Flugzeug nicht allzu viel Auftrieb und muss dies durch die Schnelligkeit, mit der der Flieger geworfen wird ausgleichen.

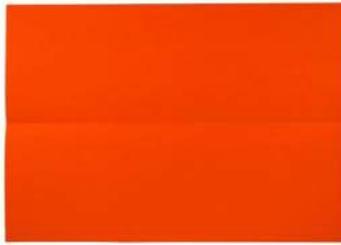


## Flieger B – Anleitung

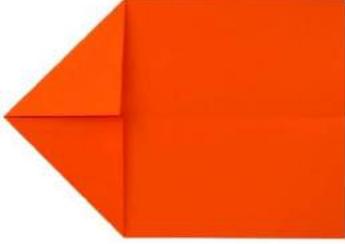
Lege das Papier hochkant vor dich.



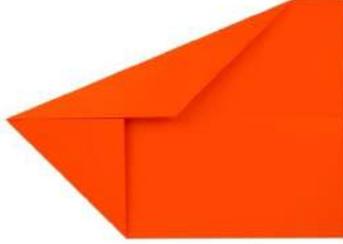
Falte das Papier im Anschluss in der Hälfte.



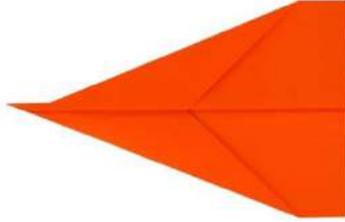
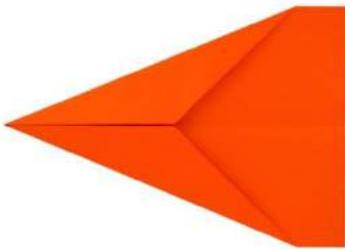
Nehme nun die oberen Ecken und falte diese zur Mitte des Papierfliegers.



Im Anschluss faltest du die oberen Kanten ebenfalls hin zur Mitte.



Lasse an der unteren Seite des Papierfliegers (Hier die rechte Kante) etwa 1,5 Zentimeter stehen und falten ab dort den Flügel um.



Knicke nun den Papierflieger an der Mittelkante.



Wiederhole den Schritt auf der anderen Seite. Achte darauf, dass die Flügel des Papierflugzeugs deckungsgleich sind. Das sorgt für eine besseren Flugbahn.



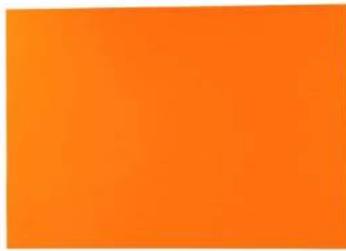


Falte nur noch etwa 1-1,5 Zentimeter der äußeren Flügelseiten rechtwinklig nach oben und fertig ist dein Papierflieger.

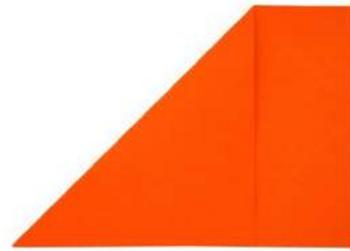


### Flieger C – Anleitung

Lege das Papier zu Anfang hochkant vor dich.



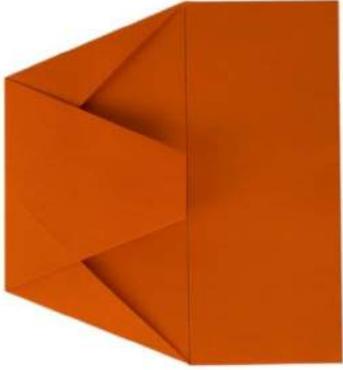
Knicke daraufhin die rechte obere Ecke zur linken Seite und umgekehrt.  
Falte das Papier anschließend wieder auf.



Falte nun die linke und rechte Kante an die diagonalen Falze heran.



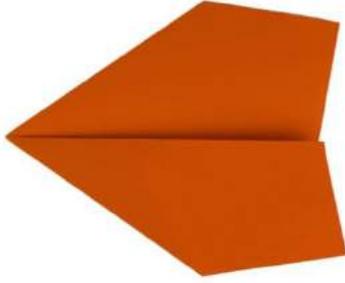
Im nächsten Schritt musst du die obere Spitze auf die Höhe des Striches falten. Der Strich wurde hier als Hilfslinie gezeichnet.



Die oberen Kanten werden nun zur Mitte gefaltet. Entlang der bereits bestehenden Falze, die ein wenig nachgezogen werden müssen.



Im Anschluss werden nun die Tragflächen entlang des Bleistiftstriches gefaltet.



Falte den Papierflieger nun in der Hälfte, sodass die zur Mitte gefalteten Seiten außen liegen.



Falte deinen Papierflieger auf und fixiere nur noch mit kleinen Tesafilmstreifen die abstehenden Flügelhälften an der Tragfläche.



Dein Gleiter ist nun fertig. Viel Spaß damit!



## Flieger D – Anleitung

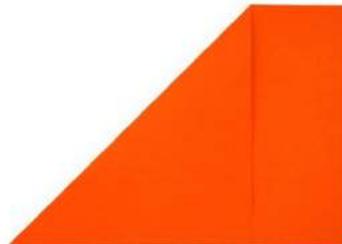
Lege das Papier zu Anfang Hochkant vor dich.



Wiederhole diesen Schritt ebenfalls auf der anderen Seite, und falte den Papierflieger wieder auf.



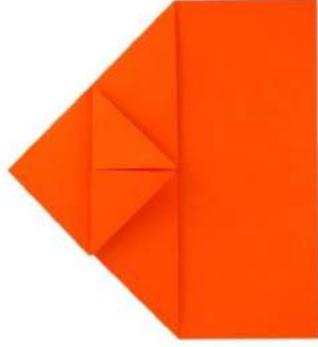
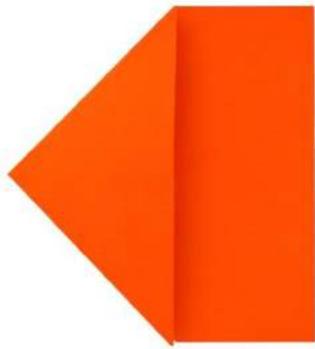
Knicke im Anschluss die rechte obere Ecke zur linken Seite.



Falte den oberen Teil des Blattes mithilfe der bestehenden Falze so zusammen, dass daraus eine dreieckige Spitze wird.



Falte die nach oben gefalteten Ecken im nächsten Schritt nach unten.



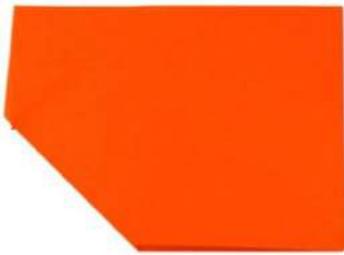
Nehme nun die unteren Ecken des entstandenen Dreiecks und falte diese zur oberen Spitze des Gleiters.



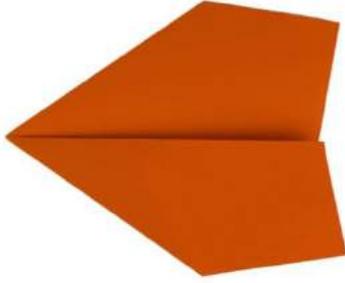
Wiederhole diesen Schritt auch mit der Spitze des Papierfliegers.



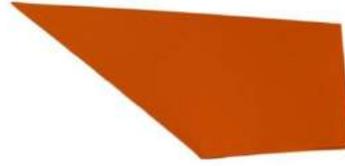
Der Papierflieger muss nun in der Hälfte gefaltet werden.



Im Anschluss werden nun die Tragflächen entlang des Bleistiftstriches gefaltet.



Anschließend musst du die Flügel umknicken. Lasse etwa 1-1,5 Zentimeter Abstand von der Unterkante (hier rechte Seite) als Griffkante.





Dein Papierflieger ist nun fertig. Viel Spaß beim Fliegen lassen!



## 13.4. Wildnispark Zürich

### 13.4.1. Exposé

Damit ich meine App im Sihlwald durchführen konnte, musste ich anfangs ein Exposé zu Händen des Wildnisparks Zürich einreichen. Darin beschreibe ich meine Arbeit kurz und erkläre genauer, inwiefern ich gewisse Infrastrukturveränderungen plane vorzunehmen.

---

# Exposé Maturitätsarbeit Rosina Weiss

## 03.06.2018

### Einleitung

Mein Name ist Rosina Weiss und ich wohne in Langnau am Albis. Ich bin 16 Jahre alt und gehe an die Kantonsschule Wiedikon in die 5. Klasse im musischen Profil. Im Rahmen meiner Maturitätsarbeit möchte ich eine App programmieren, die den Nutzer oder die Nutzerin durch den Sihlwald leitet, während man durch das Lösen physikalischer Phänomene einen virtuellen Mordfall aufklärt.

Mit meiner Arbeit möchte ich zur Weiterentwicklung des Wildnisparks Zürich einen Beitrag leisten.

Ich bin auf dieses Thema gekommen, da unsere Familie schon immer viele Spiele gespielt hat. In letzter Zeit waren auch einige Krimispiele darunter. Zusätzlich haben wir auf Wanderungen oder Spaziergängen Geocaches gesucht und das hat mir sehr viel Spass gemacht. Da ich später etwas im wissenschaftlichen Bereich studieren möchte und ich mich sehr für Physik und Informatik interessiere, habe ich mich dann entschieden, eine App zu programmieren und die Physik zum Auflösen des Falles zu gebrauchen. Meine Maturitätsarbeit wird von Herrn Fabio Torriani, einem Physiklehrer der KWI, betreut.

### Ziel

In meiner Maturitätsarbeit möchte ich kleinere Gruppen ab 15 Jahren ansprechen und diesen einen Einblick in die Physik bieten, während sie gleichzeitig Spass haben, selber Kommissar spielen dürfen und einen spannenden Mordfall auflösen dürfen. Vor allem möchte ich in meiner Maturitätsarbeit den Menschen zeigen, dass die Physik nicht so kompliziert ist, wie man denkt. Sondern ich möchte visualisieren, dass unsere ganze Natur aus Physik besteht, man sie überall antreffen kann und dass sie einfacher gebaut sein kann, als man es sich vorstellt. Ich möchte den Menschen zeigen, wie spannend die Natur neben den vielen faszinierenden Tieren noch sein kann. Ich finde, dass in der heutigen Zeit der Digitalisierung, die reale Welt oft vernachlässigt wird, da sich oft nur alles um Computer und Handys dreht. Also möchte ich einen Mittelweg finden, wie man die Natur besser in unsere heutige Welt miteinbeziehen kann. Somit würde ich gerne ein kleines Puzzleteil zur Erhaltung und zum Bekanntheitsgrad des Sihlwaldes beitragen. Ich werde meiner Nutzergruppe nach der Rallye die Besichtigung des Besucherzentrums und der Visite der Fischotteranlage empfehlen.

### Zeitplan

Bis zu den Sommerferien möchte ich den theoretischen Entwicklungsteil der Rallye abgeschlossen haben. Dies schliesst die Festlegung der physikalischen Experimente,

Phänomene und Aufgaben ein, sowohl auch die Kreierung der Hintergrundstory, des Mordfalles. Auch möchte ich bis dann die einzelnen Stationen festlegen.

Innerhalb der Sommerferien werde ich mich dann auf die praktische Arbeit stürzen. Also auf die Entwicklung und Programmierung der App und auch die Umsetzung meiner Rallye. Darin wäre inbegriffen das Fotografieren von Opfer, Verdächtigen, speziellen Hinweisen im Wald, etc. Auch würde ich in dieser Zeit, wenn dies möglich wäre, gewisse Gegenstände, kleine Schilder oder anderes im Sihlwald platzieren.

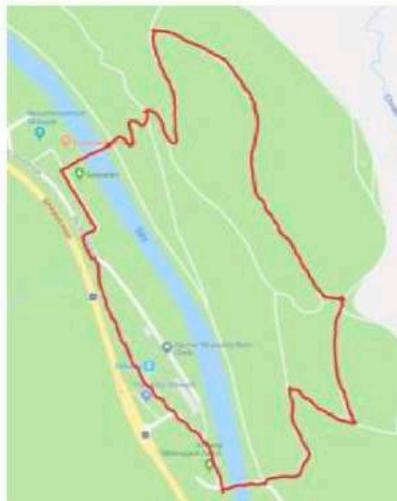
Nach den Sommerferien werde ich mich grösstenteils mit meiner schriftlichen Arbeit beschäftigen und der Verbesserung des praktischen Teils. Ich werde mit Testpersonen die Rallye ablaufen und mir Verbesserungsvorschläge holen oder Fehler beheben.

Ende Dezember werde ich meine Arbeit abgeben. Und im Januar 2019 halte ich dann meine Maturitärbeitspräsentation.

Von der Länge her würde ich sicherlich ein Jahr lang die Rallye aktiv halten, doch ich könnte mir auch vorstellen, je nachdem wie gross der Erfolg ist, dies auch noch weiterzuführen und länger zugänglich zu machen.

### Umgebung

Meine Rallye würde ungefähr den Walderlebnispfad mit kleinen Änderungen ablaufen.



Auf dem obigen Bild sieht man in rot eingezeichnet meine Route. Ich habe mir die Umgebung des Sihlwaldes ausgesucht, da erstens der Walderlebnispfad vorhanden ist und ich diesen sehr gut integrieren kann. Zweitens wohne ich in Langnau am Albis und habe früher viel Zeit im Sihlwald verbracht, kenne mich dadurch dort gut aus und finde auch die Umgebung sehr schön. Drittens gibt es im Sihlwald zwei Brücken und man

---

somit innerhalb einer angemessenen Zeit auf beiden Flussufern spazieren gehen kann. Dies ist zum Beispiel an der Sihl bei Wildpark-Höfli nicht der Fall.

### **Infrastrukturveränderungen**

Um meine Rallye realer zu machen und auch besser zu gestalten, möchte ich nicht nur alles mit der App machen. So würde ich sehr gerne gewisse Dinge in der Natur deponieren, die man dann finden oder entdecken kann.

Im Sihlwald gibt es schon vier bis fünf aktive Geocaches. Daher könnte ich sicherlich ebenfalls kleinere Boxen im Wald mit zum Beispiel einem Informationsblatt, das man braucht, um eine Aufgabe zu lösen, verstecken.

Ich würde mich sehr freuen, wenn ich eventuell kleine Schilder (etwa in A5/A6 oder kleiner) aufstellen könnte. Auf diesen Schildern würden auch physikalische Informationen oder Hinweise zum Auffinden des Täters stehen.

Hier sind noch ein paar andere Ideen, die ich hatte, was man möglicherweise anbringen oder deponieren könnte:

- An der Autobrücke an einem Pfeiler etwas rote Farbe anbringen, welches Blut vom Opfer sein könnte (ca. 2-3 cm<sup>2</sup>).
- Im Wald einen Stein deponieren, der einen Schuhabdruck aufzeigt, der dem Täter gehören könnte.
- An einem Schild des Erlebnispfads oder einem Wegweiser auf der Rückseite ein Blatt wiederum mit Informationen zu dem Fall oder der Physik zu montieren.
- Kleinere, nicht gefährliche Experimente im Wald geschützt deponieren, wie zum Beispiel einen Flaschenzug.
- ...

Natürlich würde ich darauf achten, dass alles gut und sicher deponiert wird, sodass sich weder Tiere verletzt werden könnten, noch Bäume oder andere Pflanzen einen Schaden nehmen. Etwaige Unkosten würde ich selbstverständlich übernehmen.

Schlussendlich stellt sich für mich noch die Frage, ob es auch möglich wäre, im Sihlwaldzentrum oder am Start der Rallye ein kleines Schild anzubringen, welches die Spaziergänger auch darauf hinweist, dass es diesen physikalischen Mordfall gibt und auf die App hinweisen würde.

Hier möchte ich noch anfügen, dass die App und alles andere für die Nutzer gratis sein soll. Ich möchte den Menschen Spass bereiten und ihnen etwas Lehrreiches vermitteln. Profit für mich ist nicht das Ziel meiner Maturitätsarbeit. Ich denke, die App wird eine Bereicherung für die Spaziergänger.

### 13.4.2. Koordinationsblatt

Nachdem mein Exposé geprüft und meine Arbeit bewilligt wurde, musste ich zur offiziellen Aufnahme eines Projektes noch ein Koordinationsblatt ausfüllen.



## Koordinationsblatt für sozialwissenschaftliche Forschungsarbeiten im Wildnispark Zürich

Bitte das Koordinationsblatt (Seite 1 bis 6) digital als PDF-Formular ausfüllen, speichern und zurücksenden.  
Kontakt für die Rücksendung des Koordinationsblattes und bei Fragen zum Koordinationsblatt ist:  
Ronald Schmidt, Stiftung Wildnispark Zürich, Alte Sihltalstrasse 38, 8135 Sihlwald  
Email: [ronald.schmidt@wildnispark.ch](mailto:ronald.schmidt@wildnispark.ch), Telefon: 044 722 55 21

1 Angaben zur Person	
Ausführende/r ForscherIn	<input type="text" value="Rosina Weiss"/>
Institution/Organisation	<input type="text" value="Kantonsschule Wiedikon"/>
Adresse	<input type="text" value="Birkenstrasse 46"/> <input type="text" value="8135 Langnau am Albis"/>
E-Mail	<input type="text" value="weissrosina@kwi.ch"/>
Telefon (Büro/Mobil)	<input type="text" value="078 838 41 40"/>
ProjektleiterIn/BetreuerIn	<input type="text" value="Fabio Torriani"/>

2 Angaben zum Projekt	
Projekttitel/Arbeitstitel	<input type="text" value="Krimiapp im Wildnispark Zürich zum Thema Physik"/>
Art der Forschungsarbeit	<input type="checkbox"/> Semesterarbeit <input type="checkbox"/> Bachelorarbeit <input type="checkbox"/> Masterarbeit <input type="checkbox"/> Dissertation <input type="checkbox"/> Langzeit- / Dauerforschungsprojekt <input type="checkbox"/> Auftragsforschung, AuftraggeberIn <input type="text"/>
	<input checked="" type="checkbox"/> Sonstiges, nämlich <input type="text" value="Maturitätsarbeit"/>
Projektbeginn	<input type="text" value="Juli 2018"/>
Projektende (voraussichtlich)	<input type="text" value="1 Jahr +"/>

3 Detailangaben zum Projekt	
Kurzbeschreibung des Projektes	Ich werde im Rahmen meiner Maturitätsarbeit eine App programmieren, die den Nutzer oder die Nutzerin durch den Sihlwald leitet, während man durch das Lösen physikalischer Phänomene einen virtuellen Mordfall aufklärt.
Detailbeschreibung der Fragestellung	Ziel ist es die Physik in der Natur genau zu untersuchen und sie den Nutzern und Nutzerinnen spielerisch näher zu bringen. Man soll einen Einblick in die Welt der Physik erhalten und erkennen, dass die Physik nicht nur in weissen Kitteln in Laboren stattfindet, sondern in unserem Alltag überall vorkommt.
Detailbeschreibung der Methodik	Ich werde eine IOS-App mit der Programmiersprache Swift programmieren. Diese App, welche den Nutzer oder die Nutzerin leitet, kann man, wenn die Entwicklungsphase fertig ist, im Apple-Store gratis herunterladen und benutzen.
Führen Sie Begehungen oder Feldarbeiten im Wildnispark Zürich durch? (Kartierungen, Beobachtungen, Zählungen, Befragungen, usw.)	<input type="radio"/> Ja (weiter mit Block 4) <input type="radio"/> Nein (weiter mit Block 8 auf Seite 5)

4 Angaben zu Feldarbeiten	
Zeitraum der geplanten Feldaufnahmen, bzw. genaue Daten und Anzahl der Begehungen	In der Zeit zwischen Juli und Oktober werde ich Teile des Erlebnispfades ablaufen, Fotos aufnehmen und eventuell Dinge nach Absprache mit Frau Aebli montieren.
Ort der Feldaufnahme (bitte zusätzlich Karte beilegen, Koordinaten angeben oder Link zum GIS-Browser einfügen)	Siehe Karte in meinem Exposé unter Umgebung.
Art der Feldaufnahme	<input checked="" type="checkbox"/> Neue Aufnahme <input type="checkbox"/> Aufnahme auf einer Langzeitforschungsfläche / Dauerforschungsfläche nämlich: <input type="text"/> <input type="checkbox"/> Folgeaufnahme einer früheren Arbeit, nämlich: <input type="text"/>
Beschreibung der geplanten Interaktion mit den Besuchenden (z.B. Befragungen, Zählungen usw.)	Eine Interaktion mit den Besuchenden ist via App geplant. Die Besuchenden sollen schlussendlich die App benutzen und die "Rallye" ablaufen.
Beschreibung der geplanten Einrichtungen im Feld (z.B. Zählgeräte, Posten mit Fragebögen, usw.)	Einrichtungen werden noch mit Frau Aebli abgesprochen.

### 5 Dokumentation des Untersuchungsgebietes

Ist die Dokumentaion der räumlichen Ausprägung Ihres Untersuchungsgebietes sichergestellt?

- Ja  
 Nein

Falls ja, wie wurde die Dokumentation durchgeführt, bzw. wie ist sie geplant (z.B. einzeichnen auf einer Karte, GPS-Vermessung, Fotos usw.)

Mein Vorgehen wird in meiner Maturitätsarbeit dokumentiert, teilweise auch durch Fotos, Karten, etc.

Falls nein, wünschen oder benötigen Sie Unterstützung bei der Dokumentation durch den Wildnispark?

- Ja  
 Nein

### 6 Fahrzeugbenutzung

Sind Sie bei Ihrer Feldkampagne auf ein Fahrzeug angewiesen?

- Ja  
 Nein

Falls ja, Begründung und Angabe von Anzahl Fahrten und Zeitraum, bzw. Datum

Es werden nur Materialtransporte zugelassen, jedoch keine Personentransporte. Nach Möglichkeit sollten die Transporte mit dem Rangerteam und mit Fahrzeugen des Wildnispark Zürich durchgeführt werden.

### 7 Begleitung während der Feldarbeiten

Werden Sie bei der Feldarbeit durch eigene Begleitpersonen unterstützt?

- Ja  
 Nein

Falls ja, Begründung und Namen der Begleitpersonen

Wünschen oder benötigen Sie bei der Feldarbeit Begleitung durch Ranger des Wildnispark Zürich?

- Ja  
 Nein

Falls ja, Beschreibung, wofür, wann und wo

8 Geodaten	
Benötigen Sie für Ihr Projekt Geodaten aus dem Wildnispark Zürich?	<input type="radio"/> Ja <input type="radio"/> Nein
Falls ja, welche Geodaten? Datendokumentation unter <a href="http://www.parcs.ch/wpz">http://www.parcs.ch/wpz</a>	Ich benutze für die Geodaten meine Navigationsapp auf meinem iPhone.

9 Unterstützung	
Benötigen Sie sonstige Unterstützung durch den Wildnispark Zürich?	<input type="radio"/> Ja <input type="radio"/> Nein
Falls ja, Beschreibung wofür, wann und wo?	Ich führe noch Abklärung mit Frau Aebli, inwiefern ich Sachen im Wildnispark montieren darf. Dafür bin ich sehr dankbar!

10 Bezüge und Kontakte zu anderen Arbeiten und Projekten mit ähnlichen Fragestellungen und Methoden oder in der Nachbarschaft	
Zu welchen Arbeiten oder Projekten wurden bereits Bezüge hergestellt oder bestehen bereits Kontakte?	Keine
Zu welchen Arbeiten und Projekten sind Kontakte erwünscht?	Keine

Seite 6 von 6

**11 Sonstige Mitteilungen und Anmerkungen für den Wildnispark Zürich**

An dieser Stelle möchte ich auf mein Exposé verweisen, dass ich bereits per Email verschickt habe.  
Vielen Dank für Ihre Kooperationsbereitschaft!  
Ich denke, dass meine App vielen Menschen den Zugang zur Physik erleichtert und ich in Sachen MINT-Förderung der Schweiz etwas beitragen kann.

Formular von der Forscherin / vom Forscher ausgefüllt am

14.07.18

**Der Inhalt des folgenden Blocks wird vom Wildnispark Zürich ausgefüllt!**

Projekt-ID

**Bemerkungen des Wildnispark Zürich zum Projekt**

Formular ist vollständig am

sco 20150904

---

## 13.5. Interviews

In den folgenden Kapiteln werde ich die Inhalte der drei Interviews, die ich geführt habe, aufschreiben. Ich habe alle Interviews aufgenommen und wollte diese eigentlich wortgemäss abtippen, doch nach dem ersten Interview musste ich feststellen, dass dies sehr zeitaufwendig ist. Da die Interviews nicht eine entscheidende Rolle für meine Arbeit spielen, habe ich mich dazu entschieden, die Zeit für Wichtigeres zu nutzen und die restlichen Interviews nur zusammengefasst anzufügen.

### 13.5.1. Interview mit Heike Riel

Interviewpartner: Dr. Heike E. Riel, IBM-Fellow, Director IoT Technology and Solutions IBM Research

Datum: 9. Juni 2018

Dauer: 45 min

Per Videoanruf

Rosina Weiß: Herzlichen Dank, dass ich mit Ihnen ein Interview für meine Maturitätsarbeit führen darf.

Heike Riel: Sehr gerne! Du darfst mich übrigens gerne duzen.

(Am Anfang habe ich die Idee meiner Arbeit erklärt und wie ich dazu gekommen bin.)

Was gibt es für aktuelle Forschungen oder Entdeckungen, die der Mörder machen könnte?

Wenn du jetzt dir die Story überlegst, denkst du da eher an was, dass die Welt bedroht oder an was, das der Welt ein Benefit bringen kann, aber derjenige dann den Bennefit für sich alleine haben möchte?

Ich denke, darauf kommt es nicht so drauf an. Natürlich wäre es schon spannend, wenn es auch die Welt bedrohen würde, doch ich denke, dass das Ganze dann einfach viel zu gross und wichtig werden würde. Also wenn man dann zum Beispiel eine Atomwaffe oder etwas Ähnliches nehmen würde, dann würde das zu gross und zu gefährlich werden, als dass der Fall dann von speziellen Kommissaren oder Agenten aufgeklärt wird. Aber es sollte schon etwas sein, das negativ gegen Leute verwendet werden kann, wenn es nicht von der Regierung gesteuert wird. Also wenn es wie in die falschen Hände gerät kann es negative Auswirkungen haben

Hmm, also du stellst dich da vor grosse Herausforderungen.

Ja, also es muss jetzt auch nicht so etwas Bahnbrechendes Verwendung finden.

Ja, also Themen, die sehr aktuell sind, sind künstliche Intelligenz. Das wäre ein Thema und das andere Thema ist Quantencomputing.

Stimmt, da habe ich letzstens einen Artikel darüber gelesen.

Genau, also das sind zwei Themen, die jetzt sozusagen noch in den Anfangsschuhen stecken. Wo aber jetzt unheimlich viel Ressourcen und Investment verplant werden und auch viele Leute sich dem Thema widmen, sodass man in den nächsten paar Jahren erwartet, dass da sich noch viel mehr entwickeln wird. Also dass da grosse Entwicklungssprünge auch stattfinden werden, bei denen man zum Teil auch noch gar nicht beide Komponenten hat. Also wenn du jetzt die künstliche Intelligenz anschaust, hat es sozusagen eine negative Komponente, aber auch eine positive Komponente. Je nachdem wie die Mechanismen und Algorithmen eingesetzt werden, können sie eben sehr positive Beiträge zur Gesellschaft ergeben, aber auch negative Beiträge. Und das ist auch schon was, was von den Wissenschaftlern, aber von der Regierung, den politischen Parteien und auch von der Gesellschaft selber in die entsprechenden Richtungen gelenkt wird. Quantencomputer stecken im Moment noch in den Anfangsschuhen. Also das ist ein Thema, was es schon seit den 80er Jahren gibt und das ist auch ein sehr schwieriges technisches Thema zum Realisieren. Aber man hat nicht nur Probleme vom Verständnis her, sondern auch von den ganzen physikalischen Gesetzen, die dahinterstecken. Ich weiss halt natürlich nicht, wie viel du schon darüber weisst. Habt ihr das schon behandelt in der Schule?

Nein, leider nicht, wir sind jetzt grad bei Elektromagnetismus, also noch nicht so weit.

Ja, wenn man sozusagen in die Quantenphysik geht, gibt es dort andere Gesetzmässigkeiten als man es aus der klassischen Mechanik, klassischen Physik kennt. Und die sind zum Teil eben auch nicht so intuitiv, wie wir es aus unserem Leben kennen. Das Leben, wie wir es sozusagen kennen und auch jeden Tag empfinden, ist eben durch die klassische Mechanik, die klassische Physik abgedeckt. Und erst als man näher hingeschaut hat und andere Mechanismen fand, konnte man die eben nicht mehr mit der klassischen Physik erklären und musste dann die Quantenphysik nehmen. Da gibt es ganz viele Dinge die auf dich zukommen werden, wenn du Physik studierst.

Und diese neuen Gesetzmässigkeiten kann man im Quantencomputer ausnutzen. Dadurch kann man mathematische Probleme lösen, die exponentiell komplexer werden, wenn man mehr Komponenten hat. Die kann man auch heute auf den Riesencomputern nicht lösen. Was der Physiker macht sind dann Annäherungen und man muss dann schauen, ob diese Annäherungen auch wirklich zutreffen. Je grösser der Computer wird, umso besser ist die Annäherung, aber wenn es eben exponentiell grösser wird, bereitet es sehr grosse Probleme und irgendein Computer, den wir jemals bauen werden, kann uns nicht helfen. Doch der Quantencomputer kann das, weil der Quantencomputer durch die Mechanismen in der Quantenphysik eben ein anderes Skalierungsverhalten hat. Er skaliert sozusagen auch exponentiell, wenn ich ein weiteres QBit hinzugebe. Im klassischen Verfahren hat man ja ein Bit und im Quantencomputer hat man dann QBits. Also wenn ich ein QBit hinzufüge, bekomme ich eine Verdoppelung der Leistungsfähigkeit. Und dadurch hat man auch ein exponentielles Verhalten der Leistungsfähigkeit und damit kann man einige dieser sehr komplexen mathematischen Probleme, die man nicht lösen kann, lösen. Ein Problem davon ist die Primzahlzerlegung. Das habt ihr in der Schule sicher auch gehabt, das ist kein einfaches Problem. Das ist zwar einfach sozusagen zwei Zahlen miteinander zu multiplizieren, aber dann eine Primzahl zu zerlegen, ist eben nicht einfach. Wenn eben die Zahlen ganz gross werden, kann die Rechenzeit die man dafür braucht mit den Computern in die

---

300 Jahre gehen. Und das sind natürlich Zahlen die sozusagen unrealistisch sind und deshalb benutzt man diese Primzahlen auch zur Verschlüsselung von Datenübertragung. Das hast du vielleicht auch schon einmal gehört. Wenn wir jetzt aber einen Quantencomputer hätten, dann wäre die Verschlüsselung der Datenübertragung belanglos. In der Realität braucht man für die Entwicklung eines Quantencomputers, den man für diese Art des Problems herstellen kann, sicher noch 20 bis 30 Jahre. Also es liegt noch weit in der Zukunft. Deshalb ist diese Gefahr auch nicht gegeben und auch deshalb nicht gegeben, da die Leute erkannt haben: „He, das ist ein Problem“ und bis dann haben sie schon wieder an etwas gearbeitet, das dem entgegenwirkt. Also sie arbeiten daran, neue Verschlüsselungsmethoden zu finden, die auch durch einen solchen Quantencomputer nicht gelöst werden. Jetzt ein Szenario, das du dir überlegen könntest, wäre ja: He, da hat jetzt wirklich jemand einen Quantencomputer, der jetzt so gross ist und in der Lage wäre, dieses Verschlüsselungsproblem zu lösen. Und das wäre auch im Rahmen, dass die Regierungen das kontrollieren wollen. Wenn die Amis oder die Russen das hätten, dann wäre das schon so ein Szenario, wo das einer für Machtzwecke nutzen kann. Oder eben auch ein Gangster, der dann die Bankkonten leerräumt oder auch Delikte in der Wirtschaftskriminalität. Das kann ich mir auch vorstellen, dass die sich die Wirtschaft an sich reissen wollen und die dann entsprechen verkaufen wollen. Ja, das wäre eine Möglichkeit.

Wenn man in die künstliche Intelligenz geht, ist das Ganze komplexer, denke ich, weil es dort viel mehr Szenarien gibt. Ein interessantes Thema, was ich letztens gesehen habe in Einstein, das kennst du vielleicht auch. Das ist vom Schweizer Rundfunk eine Sendung. Und da war glaub im März oder im April oder sogar Mai eine Sendung über künstliche Intelligenz. Und da waren auch so Szenarien enthalten zum Beispiel, welche Roboter man heute schon bauen kann, und die kann man dann natürlich auch als Waffen einsetzen.

Ein anderes Szenario war, dass es ganz kleine Minidrohnen gibt und diese Minidrohnen, die dann sozusagen die Technologie enthalten, dass sie auf Menschen losgehen können, dass sie dann auch ein Bild machen können, eine Bilderkennung haben und dann auch erkennen, wer dieser Mensch ist. Die hätten dann drei Gramm Sprengstoff und diese kleine künstliche Biene könnte auf den Menschen losgehen, zum Beispiel auf den Kopf losgehen und dann sozusagen mit diesen drei Gramm Sprengstoff einen Kopfschuss generieren. Und das wäre auch als Szenario denkbar. Es ist scheinbar auch im Umlauf auf YouTube und sogar hat sich das Schweizer Militär das dann auch genau angeschaut und überprüft, ob das ideal wäre. Und sie mussten leider feststellen, dass diese drei Gramm Sprengstoff wirklich ausreichen würden, um jemanden zu töten. Die würden dann wirklich den Kopf sozusagen durchbohren. Ja, da war ich auch sehr schockiert, als ich das gesehen habe. Und da überlegt man sich dann auch Szenarien, wie man sich davor schützen kann. Künstliche Intelligenz ist eben, also da gibt es so viele Szenarien, dass es vielleicht ein bisschen schwieriger wird, ein passendes zu finden.

Wir beschäftigen uns hauptsächlich mit positiven Sachen und versuchen eben auch Probleme, die es da gibt, zu beheben, weil bei künstlicher Intelligenz ist es ja heute so, dass viel von den Daten abhängt, die zum Trainieren des künstlichen Netzwerks verwendet werden. Je nach Daten kann das zu einer gewissen Einseitigkeit führen, zum Beispiel wenn du Daten hast, die nicht den Gesamtbereich abdecken, dann bekommst

du eine Antwort, die nur in eine Richtung geht und nicht die richtige Antwort sein muss oder zu einer Bios führen kann. Hmm, was ist das deutsche Wort für Bios. Also nicht Spannung, sondern wenn man einen einseitigen Blick hat auf ein Thema. Sozusagen dass du einen gewissen Vorzug hast für weisse Männer oder so was, weil die Daten grundsätzlich so aufgebaut sind oder dass die Daten so aufgebaut sind, dass alle Banker weisse Männer sind. Dann kann man gewisse Bios führen, die wir eigentlich nicht haben oder die eigentlich nicht da sein sollten. Und da gibt es eben auch Forschungsthemen, mit denen wir uns beschäftigen und sich auch die Universitäten damit beschäftigen, dass sie diese Bios aus den Daten herausnehmen.

Andere Themen wären im Bereich von Erfindungen. Dass du zum Beispiel sagst, dass jemand eine einzigartige Erfindung gemacht hat, wie die Batterie um einen Faktor 10'000 verbessern zu können. Heutzutage sind ja Autobatterien ein riesiges Thema und auch für den zukünftigen Verkehr, der sich ja ändern wird. Das sind in dem Bereich die wichtigen Themen und Batterien sind eine wichtige Komponente für die zukünftigen Autos. Ich meine, wir wissen, dass die Elektromobilität immer wichtiger wird, nur schon aufgrund des Ausstosses von Öl und Benzin, also bei der Verbrennung von Stickoxiden, CO<sub>2</sub> und alles Möglichem. Und dass man eben mehr Greenenergy verwenden kann. Aber es braucht immer die Batterie, die noch besser werden muss, damit die Autos leichter werden und wenn die leichter werden brauchen diese weniger Energie und so weiter. Oder auch für unsere mobilen Geräte, für unsere Handys und alles möchten wir ja auch bessere Batterien haben, sodass wir es nur noch einmal im Monat aufladen müssen. Wenn jemand da eine Lösung hat, dann kann er damit auch unheimlich viel Geld verdienen.

**Nochmals zur künstlichen Intelligenz, die hat ja eher mehr mit Informatik zu tun, als mit Physik, oder?**

Künstliche Intelligenz ist eigentlich sehr viel Computerscience, aber auch Mathematik. Weil die Algorithmen dahinter viel mit Mathematik zu tun haben.

**Also so sehr physikalisch ist das nicht.**

Ne, da würde dann eher Quantencomputing besser passen. Du hast bei der artificial intelligence schon mehr die Computerscience und die Mathematik, die dahintersteckt. Was man natürlich da von der Physikseite miteinbringt, sind die Rechner. Unsere heutigen Rechner, die wir verwenden, die sind nicht optimiert für künstliche Intelligenz also für die Berechnungen, die man in der künstlichen Intelligenz braucht. Um künstliche Intelligenz, sinnvoll zu verwenden, braucht man auch eine unheimliche Rechnerleistung, weil man diese neuronalen Netzwerke mit Daten trainieren muss. Also muss du Daten hindurchschicken und sagen, das war richtig, das war falsch, das war richtig, damit der Rechner wie ein Kind auch lernt. Und hierfür benötigt man ganz viel Computerleistung. Dieses Training wird zum Teil auf Supercomputern gemacht. Und jetzt überlegen wir uns als Physiker und Elektroingenieure: Wie können wir neue Computer bauen, um dieses Training zu beschleunigen? Dass man es schneller machen kann, denn heutzutage kann es sein, dass das Training auf dem Supercomputer eine Woche dauern kann, was sehr teuer ist. Und deshalb kann man neue Mikroprozessoren, also neue Computer bauen, wo das Ganze schneller geht und damit dann auch günstiger wird und auch weniger Energie verbraucht und auch nochmal eine Beschleunigung der

---

künstlichen Intelligenz geben wird, weil man mehr in weniger Zeit machen kann oder auch grössere Aufgaben lösen kann.

Und da arbeiten wir auch an neuen Computerchips. Das sind dann auch neue Materialien und neue Bauelemente, also der Transistor den wir heutzutage verwenden, wird da nicht mehr das Bauelement sein. Da kommt dann die Physik hinein. Doch die ist eher die Grundvoraussetzung, damit der Rest obendrauf funktioniert.

Was noch ein weiterer Bereich wäre „internet of the things“, das hast du sicherlich auch schon mal gehört. Das ist es dann auch, dass jedes mobile Device, was wir besitzen, und das ganze Haus und die ganze Umgebung miteinander vernetzt ist. Der Kühlschrank oder deine Lampe in deinem Zimmer oder dein Mobiltelefon, deine Uhr oder was weiss ich sind alle miteinander vernetzt. Die Gefahr, die man da im Augenblick sieht, ist, dass man die Sicherheit in jeder einzelnen Komponente haben muss. Denn so Hackerangriffe, hast du wahrscheinlich auch gehört, wenn dann sozusagen jemand über deine Uhr man dann in diese Netze reinkommt. Und was eben häufig vergessen wird ist, dass man bei diesen kleineren Elementen eben weniger auf die Sicherheit achtet, weil das der Computer tut. Dadurch, dass alles vernetzt ist, können so Lecks gefunden werden, bei diesen anderen Devices. Und dann kann man einen Hackerangriff durchführen. Da ist man auch sehr an der Forschung dran, dass man da diese Lecks stopft und dass man auch die Leute drauf aufmerksam macht, dass wenn sie sich selber solche Dinge anschaffen, dass sie dann die Passwörter ändern, sodass diese nicht mehr die gleichen Passwörter sind, wie die gegebenen vom Hersteller. Weil sonst ein Angriff möglich ist.

Also da könnte man sich dann auch Szenarien vorstellen. Zum Beispiel wenn jemand etwas erfindet, wie er das alles sofort mit ganz einfachen Möglichkeiten sicher macht, dass manche vielleicht nicht haben wollen, weil sie es ausnutzen wollen. Dann würde es auch einen Konflikt geben. Also so in die Richtung. Ja, also Quantencomputer wäre sicherlich eine spannende Sache, weil du da dann auch über Physik schreiben kannst. Da kannst du dann auch etwas über diese veränderte Skalierung schreiben und du kannst sozusagen ein Szenario bilden, dass jemand schon einen Quantencomputer hat, der wirklich ganz viele Quantenbits hat um diese Primzahlenzerlegung durchzuführen.

Neben der Hintergrundstory entwickle ich einzelne Stationen, bei denen man dann stückchenweise was von der Hintergrundstory herausfindet. Bei diesen Stationen möchte ich auch die Physik hineinbringen durch Experimente oder ähnliches. Grundsätzlich geht das nur mit Mechanik und Optik, da man im Wald nicht so gut etwas mit zum Beispiel Strom machen kann. Vielleicht weisst du da auch ein bisschen mehr als ich, aber was für Experimente kennst du, die man gut in der Natur durchführen kann?

Da stellst du mich vor eine grosse Herausforderung. Typischerweise haben wir Laborzeug usw.

Ja, eben. Es muss aber überhaupt nicht kompliziert sein, also eher einfache Sachen.

Ja, also da weiss ich grad nicht so viel. Ich glaube, da müsste ich mir länger Gedanken dazu machen, weil es spannend wäre, das dann auch mit dem Quantencomputer zu verbinden. Es gibt vielleicht Dinge, die man mit seinem Handy machen kann, was ja jeder dabei hat.

Ich habe eh vor, eine entsprechende App zu programmieren, dass man dann auf dem Handy irgendwelche Aufgaben hat, die man dann lösen muss.

Ah, okay. Also die IBM hat irgend so ein Quantenspiel, also ich muss selber mal nachschauen, aber die haben eine App gemacht für ein Spiel, wo Quantencomputing hineinkommt. Es gibt eine Sache zum Beispiel, wenn du vier Karten hast und du hast da einen König und du möchtest den König finden, dann macht der klassische Computer wie jeder Mensch. Also er dreht jede Karte und dann findet er so den König. Und statistisch gesehen sind das glaube ich 1.4 mal oder irgendwie so, wie oft er die Karte umdrehen muss. Und der Quantencomputer kann das eben aufs Mal. Weil der mehr oder weniger unter alle Karten gleichzeitig runterschauen kann und genau weiss, wo die liegt. So ein ähnliches Spiel, glaube ich, haben wir auch als App zur Verfügung gestellt. Also das wäre vielleicht auch etwas, was man sozusagen verdeutlichen kann was ein Unterschied ist beim Quantencomputer. Was sind so klassische Beispiele für Quantenphänomene? Da müsste man vielleicht nochmal nachschauen. Also so Beispiele, die man immer hernimmt um Quantenphänomene darzustellen könnte man damit auch etwas mit einfachen Mitteln machen. Das Doppelspaltexperiment, das wäre vielleicht eine Möglichkeit. Kennst du das Doppelspaltexperiment?

Nein, das kenne ich nicht.

Na, dann schreibe dir das mal auf und schau das bei Wikipedia nach. Da geht es eben darum, dass du zwei Spalten hast und da geht ein Licht durch und das soll den Elektronencharakter beschreiben. Das wäre vielleicht eine Möglichkeit, die man sich noch anschauen könnte. Darüber müsste ich mir noch Gedanken machen, da kann ich jetzt aus dem Stehgreif nicht so viel sagen und Beispiele geben.

Nein, es ist schon gut. Ich glaube, da werde ich keine Probleme haben, genügend Sachen für meine Arbeit zu finden. Ich bin mir auch noch nicht sicher, was das Hintergrundwissen dafür sein soll. Gewisse Menschen haben noch keine Ahnung von Physik und ich muss dann einen guten Mittelweg finden. Aber ich glaube, ich baue meine Inhalte auf dem Schulunterricht auf und werde sicherlich etwas finden.

Hattet ihr schon Spektralzerlegung?

Nein, das Thema haben wir noch nicht behandelt.

Dabei handelt es sich um Licht, das aus verschiedenen Farben besteht. Dieses Phänomen kann man auch bei der Entstehung von Regenbögen beobachten.

Achja, jetzt erinnere ich mich, das hatten wir bereits.

Das Licht könnte über ein Prisma zerlegt werden oder ähnliches.

Das ist eine gute Idee, die man sicherlich sehr gut einbauen könnte. Man könnte gut ein Prisma deponieren, womit die Menschen dann ihre Erfahrungen machen könnten.

Dabei könnte man beobachten, wie das Licht vom Tageslicht aussieht. Und ob es einen Unterschied gibt, wenn man es mit der Handytaschenlampe nachahmt. Dann könnte man der Frage nachgehen, ob es ein unterschiedliches Spektrum mit verschiedenen Farben hat.

---

Darf ich dir noch eine Frage stellen bitte. Du hast ja, soweit mir bekannt ist, eine Ausbildung als Tischlerin gemacht.

Ja, als Schreinerin.

Wie bist du von diesem Handwerk auf die Naturwissenschaft Physik gekommen?

Oh, wie bin ich auf die Physik gekommen? Ab der ersten Klasse hat mich die Mathematik und die Physik interessiert und ich habe diese Themen sehr spannend gefunden. Die Physik kam als Schulfach zwar später, aber die Mathematik hat auf mich eine grosse Anziehung ausgeübt und es hat mir sehr viel Spass gemacht. Die gesamte Schulzeit hindurch waren das meine Lieblingsfächer. Deutsch hat mich genervt, aber Mathe und Physik habe ich geliebt. Dann ging ich in Deutschland auf die Realschule und dort war ich supergut, aber ich wollte nicht auf das Gymnasium, weil mein Bruder ein schlechtes Beispiel war, denn er quälte sich mit Latein ab. Das wollte ich auf keinen Fall. Ich hatte mir viele Gedanken gemacht, was ich danach machen wollte. Meine Eltern hatten eine Schreinerei und ich wollte eigentlich in Richtung Innenarchitektur gehen. Das wäre auch eine gute Kombination gewesen, so als Schreinerin und Innenarchitektin. Als Voraussetzung für das Innenarchitekturstudium erschien mir eine Schreinerlehre optimal, damit ich weiss, was ich alles so bauen könnte. Es hilft immer, wenn man weiss, wie etwas funktioniert. Man hat dann die Kompetenz darin. Dann ist man unabhängig, weiss selber, wie es geht, und muss sich von niemandem etwas sagen lassen. Und so habe ich die Ausbildung gemacht. In der 10. Klasse vor der Mittleren Reife kam jemand zu Besuch in die Klasse und hat erzählt, dass man mit einem Berufschulabschluss auch studieren könnte. Und diesen Weg habe ich schlussendlich eingeschlagen und nach der Lehre studiert. Ich wollte schon immer auf die Fachhochschule oder eben Innenarchitektur studieren. Also habe ich die Lehre begonnen und bin dann an die Fachoberschule gegangen. Dann habe ich zwischendurch die Information erhalten, dass ich das Abitur nach der Lehre nachholen kann. Dies war dann eher das, was ich wollte. Und dann kam der Wunsch, Physik zu studieren. Das hat sich einfach so entwickelt. Eines kam nach dem anderen. Ich habe dann die BOS gemacht und konnte das Abitur nachholen und dann Physik studieren. Später stellte sich die Frage, wo ich meine Masterarbeit (damals hiess das noch Diplomarbeit) machen würde. Und dann habe ich die Gelegenheit gehabt in die USA für ein halbes Jahr zu gehen. Dort habe ich bei einem Intership ganz viel gelernt und hatte viel Spass und Zugang zu ganz vielen verschiedenen Geräten. Wenn man Experimentalphysiker ist, ist es spannend mit vielen Geräten umzugehen. Man kann ganz viel aufbauen und testen und probieren und machen und tun. Z. B. kann man Nanofabrikationen aufbauen, kleine Bauelemente machen, diese charakterisieren und verbessern, dabei verstehen, was eigentlich passiert. Nach diesen Erfahrungen habe ich meine Diplomarbeit bei IBM Rüslikon in der Schweiz geschrieben, weil ich industrielle Forschungsumgebungen sehr spannend finde. Da hat man ein gemeinsames Ziel und man ist interessiert an dem, was man macht. Die haben mich danach gefragt, ob ich nicht eine Doktorarbeit bei ihnen schreiben möchte. Anschliessend haben sie mir dann eine Stelle angeboten. Und so hat sich das Ganze entwickelt. Bei einer Firma ändert man auch seine Forschungsthemen, weil sich die Zeiten auch weiterentwickeln.

Das kann ich mir durchaus vorstellen.

Dann habe ich zum einen auf Displays gearbeitet, das nennt sich Organic LED. Das sind Displays, die mittlerweile in den Mobiltelefonen verwendet werden und in den Fernsehern vorkommen. An den Displays habe ich Ende der 90er Jahre und Anfang 2000 gearbeitet. Danach habe ich neue Forschungsprojekte mit zukünftigen Bauelementen für neue Computer angefangen.

### Jetzt bist du ein IBM-Fellow? Was ist das? Was bedeutet das? Was machst du genau?

Vor 55 Jahren hat der Gründer von IBM den IBM-Fellow ins Leben gerufen: als technisches Gewissen der IBM, aber auch die höchste Auszeichnung im Bereich der technischen Kompetenz der IBM. Er wollte diese Gruppe von Leuten zum einen herausheben, eine Vernetzung zwischen diesen Leuten bilden, so dass diese sich austauschen, und dass auch eine kritische Stimme innerhalb der IBM da ist, die auch die Freiheit hat, dem Management eine ehrliche Meinung zu geben. Das ist somit das technische Gewissen des Unternehmens, welches geschützt und gefördert wird. Das heisst innerhalb der IBM gibt es etwa Hundert IBM-Fellows. Das ist relativ wenig, wenn man sieht, dass IBM 400'000 Menschen beschäftigt. Das heisst sozusagen, dass sich die Fellows technisch ausgezeichnet haben. Sie befinden sich in der höchsten Stufe der Leute in der IBM. Die Fellows werden auch direkt vom CEO in diese Rolle befördert und haben dadurch eine grosse Verantwortung gegenüber der Firma und auch den technischen Mitarbeitern der Firma. Für die technischen Kompetenzen innerhalb der Firma sind die Fellows auch verantwortlich und auch dafür, dass sich die Leute austauschen und die richtige technische Strategie verfolgen und nehmen dadurch auch die Position des Ratgebers ein.

### Was forschst du jetzt gerade?

Man hat als IBM Fellow eine grössere Verantwortung über einen Forschungsbereich, d. h. die eigene Forschung tritt manchmal auch in den Hintergrund, weil man bei grösseren Projekten die Richtung vorgeben muss, in die man gehen sollte. Und dann arbeitet man zusammen mit den Menschen zusammen, die im Labor die eigentliche Arbeit verrichten. Man schaut, welche Dinge sinnvoll sind und welche nicht. Einen Teil arbeitet man mit Partner zusammen, um zu sehen, was für sie wichtig ist. Dabei ist auch die Kommunikation mit dem technischen Personal und dem Marketing wichtig.

### Zum Schluss habe ich noch eine Frage: Wie definierst du Physik?

Ach, wie definiere ich die Physik? Das ist eine schwierige Frage... Für mich ist die Physik die Gesetzmässigkeiten der physikalischen Welt, zu verstehen, wie was warum funktioniert. Und dadurch ergeben sich die physikalischen Gesetzmässigkeiten, die wir als Physiker aufstellen. Zum Teil sind ja welche auch noch nicht erwiesen, weil wir manche physikalischen Phänomene noch nicht widerspruchsfrei erklären und beweisen können. Aber über die letzten paar hundert Jahre hat sich viel entwickelt und man hat viel entdeckt und konnte auch viel aufklären. Für mich ist eigentlich Physik die Lösung der Frage „Wie funktioniert die Welt?“ Viele der Gesetzmässigkeiten findet man in der Biologie und Chemie wieder. Aber ich muss auch sagen, dass die Physik ein sehr breites Thema abdeckt, z. B. von der Astronomie über die Kernphysik bis zur Quantenphysik über die Konstruktion von Displays und Mikrochips, aber die Themen verschmieren auch mit denen der Elektroingenieure, mit den Chemikern und den Biologen. Das

---

Spannende an der Physik ist auch, dass man nicht auf ein Thema fokussiert ist, sondern ein grundsätzliches Verständnis der Mathematik bekommt und dann mit diesen mathematischen Instrumenten versucht, die Natur und die Umgebung zu erklären. Und wenn du dieses Verständnis hast, kannst du auch dein analytisches Denken prägen. Mit dem kannst du in verschiedene Berufe gehen. Deswegen habe ich nie bereut, Physik studiert zu haben, weil dies für mich das Erweitern der Frustrationstoleranz war. Du stellst fest, dass du nicht alle Aufgaben lösen kannst, denn es gibt immer noch schwierigere Aufgaben, da gibt es nach oben kaum keine Grenze. Aber das analytische Denken wird geschult. Dies ist für alle Bereiche, wo es darum geht, Probleme zu lösen, essentiell. Egal, ob es das in der Wirtschaft ist oder woanders. Es gibt auch viele Physiker in der Wirtschaft.

(Sie hat noch Interesse an meiner Arbeit gezeigt und ich habe ihr das weitere Vorgehen erläutert.)

Vielen Dank für das Gespräch.

### 13.5.2. Interview mit Jan-Peter Meyn

Interviewpartner: Prof. Dr. Jan-Peter Meyn, Professor für Didaktik der Physik, Universität Erlangen-Nürnberg

Datum: 10. Juni 2018

Dauer: 1 h

Per Telefon

Rosina Weiß: Grüezi, erstmals möchte ich mich ganz herzlich bei Ihnen bedanken, dass ich ein Interview mit Ihnen führen darf.

Jan-Peter Meyn: Ja gerne, doch bis jetzt habe ich ja noch nichts gemacht...

Ist es für Sie in Ordnung, wenn ich das Gespräch aufzeichne?

Ja, das ist kein Problem.

(Ich habe dann kurz erklärt, was ich innerhalb meiner Maturitätsarbeit entwickeln und erreichen möchte.)

Was für Experimente oder physikalische Phänomene gibt es, die man in der Natur gut darstellen oder ausführen kann?

Ich denke, da gibt es viele. Ich habe mal, ist auch schon viele Jahre her, eine Fortbildung gemacht für Kindergärtnerinnen. Und mit denen bin ich auch in den Wald gegangen, weil die ja die Vorstellung haben, dass wenn sie Physik im Kindergarten machen, sich einen weissen Kittel anziehen müssen und nur dann sind die Eltern zufrieden. Hmm, das muss ich jetzt mal geizig hervorkramen, was ich mit denen gemacht habe. So der

Klassiker ist der Sonnentaler. Ich weiss nicht, ob du das kennst. Wenn die Sonne durchs Blätterdach scheint, dann sind ja die Blätter, die in der Nähe vom Boden sind, noch relativ scharf umrandet. Und je höher die Blätter kommen, so unschärfer werden die Schatten, und die Löcher werden auch immer gleichmässiger. Wenn man durch eine Hecke oder durch Blätter durchguckt, dann sind die Löcher erstmal sehr unregelmässig. Und in einem grossen Abstand, werden diese Löcher zu Kreisen und diese Kreise sind dann Sonnenbilder, die hervorgerufen werden durch eine Kamera-obscura. Also durch eine mehr oder weniger rundförmige oder kleinere Öffnung im Blätterdach, die klein ist gegenüber den Abständen. Also wir haben hier so Jalousien im Haus, da wo die Bänder, die die Jalousien zusammenhalten, durchgehen, sind auch so kleine Löcher. Also haben wir die Sonnentaler auch im Haus. Also wenn man die mal gesehen hat, dann sieht man die immer wieder. Ich wüsste jetzt zwar nicht, wie man da einen Kriminalfall draus macht, aber das ist sozusagen der Klassiker an Phänomenen im Wald, die man sehen kann. Da muss man dann zwar die richtige Stelle suchen und das geht nur im Sommer, aber ich denke, dass man da schnell fündig wird. Eine andere Sache, an die ich mich noch erinnern kann, ist die mit der Blütenfarbe. Es gibt ja viele blühende Pflanzen im Wald. Und viele davon sind ja im rötlichen bis violetten Bereich von der Blühfarbe her. Es gibt auch Weisse und wenige Gelbe, doch es gibt fast keine Blauen oder schon gar nicht Grüne. Die rötliche Blütenfarbe ist ziemlich häufig und dies stimmt gut mit dem überein, dass Rot und Grün Komplementärfarben sind. Denn der Wald ist ja wegen seinen vielen Blättern grün und die roten Blüten werden durch die Komplementärfarbe besonders interessant. Ob das jetzt biologisch korrekt ist, weiss ich nicht, aber das ist mir aufgefallen. Klar haben Insekten wie zum Beispiel Bienen ein ganz anderes Farbsehen, also kann man das nicht verallgemeinern und eine Wissenschaft daraus machen, aber zumindest mal eine Beobachtung, dass bei vielen Pflanzen, auch nicht nur denen im Wald, die Blütenfarbe in der Regel deutlich von der Pflanzenfarbe selbst abweicht.

Wahrscheinlich wird mein endgültiges Zielpublikum Menschen ab 15 Jahren sein, dass ein gewisses Hintergrundwissen vorhanden sein sollte. Man sollte Begriffe wie Geschwindigkeit und Beschleunigung kennen. Denn ich persönlich finde, dass man dann eher in die spannenderen Teile der Physik kommt.

Da muss ich aus der Sicht der Physikdidaktik etwas anfügen. Denn wie du ja vermutlich schon weisst, mache ich Lehrerausbildungen für Schulphysik und so. Da ist es so, dass wenn man sich selber für Physik interessiert, sich immer schwer vorstellen kann, wie schwer das „normalen Menschen“ fällt, Physik zu verstehen. Und das ist schon auch seit Jahrzehnten wissenschaftlich untersucht. Da spricht man von sogenannten Präkonzepten, dass Menschen die jetzt ganz normal zur Schule gehen und nicht eine spezielle Antenne für Physik haben, sich manchmal komische Sachen darunter vorstellen. Jüngstes Beispiel, dass ich hier in der Familie gehabt habe: Ich schreibe im Moment ein Buch über Infrarotstrahlung und dann frage ich meine Frau, wie das eigentlich ist, wenn man vor dem Feuer sitzt, dann ist das ja warm. Das nennt man dann Strahlungswärme, das kann man besonders bei grossen Feuer gut spüren. Und weisst du, wie das bei einem Kachelofen ist? Und da hat sie mir geantwortet, dass es dort auch Strahlungswärme gibt, weil da ja eine Glut drin ist. Dann habe ich sie gefragt, was passiert, wenn das Feuer ausgeht. Und darauf hat sie geantwortet, dass es dann noch ein bisschen nachstrahlen würde, aber nicht lange. Und das ist so ein typisches

---

Präkonzept, weil mit einem physikalischen Sachverstand sagt man als Physiker: „Ja ich weiss, es gibt Infrarotstrahlung und ein Körper der vielleicht 200°C heiss ist, strahlt nur im Infrarot und das sieht man nicht und zwischen einem heissen Bügeleisen, einer Glühbirne und einem Kachelofen ist da kein Unterschied.“ Du verstehst das jetzt vielleicht auch, da du entsprechendes Hintergrundwissen hast, doch „normale Menschen“, selbst solche, die mit einem Physiker seit 30 Jahren unter einem Dach wohnen, die denken so nicht. Und hier wäre mein Rat, dass du das berücksichtigen solltest, dass die Physik möglichst ohne Vorkenntnisse ist. Sonst richtet sich das an ein Spezialistenpublikum. Nun gibt es unter den Menschen, die sich für Physik interessieren, einige die auch mal in den Wald gehen, Sport machen oder so. Aber der Überlapp ist sehr klein. Es gibt auch viele, die sich eher für Computer und andere leblose Sachen interessieren und dann gar nicht erst in den Wald gehen. Die Nerds kommen gar nicht erst, um diesen Wanderweg zu machen. Deswegen würde ich das so ausrichten, dass das ganz normale Leute auch machen können.

Ich will ja eine App dazu programmieren und da habe ich überlegt, dass ich so eine Art Theoriebuch mache und in diesem Theoriebuch kann man sich das benötigte Wissen aneignen. Aber so Leute wie mein Papa brauchen dieses Theoriebuch nicht. Dieses könnte freiwillig genutzt werden.

Das ist eine gute Idee mit diesem Hintergrundwissen, aber es darf nicht dazu führen, dass Leute die dieses Wissen noch nicht haben, dort immer nachlesen müssen. Dann verlieren sie die Lust und machen die App gar nicht fertig. Es braucht auch einfache Aufgaben, die jeder mit ein bisschen gesundem Menschenverstand lösen kann. Ich sehe das natürlich aus der Perspektive des Physiklehrers. Ich glaube, bei deiner App ist es wichtiger, dass die Leute Spass haben, als dass sie etwas über Physik lernen.

Ich habe mir auch überlegt, ob ich zwei Versionen mache, also eine spassige Version und eine eher schwierigere für die, die auch was lernen wollen.

Ich will mich ja nicht zu stark einmischen, aber das würde ich nicht machen, weil das einfach nur doppelte Arbeit ist.

Das stimmt schon.

Ich würde lieber nur eine Version machen, dass man sich in ein Thema einarbeitet. Man kommt bei der Geschichte ja auch an Dinge, die man jetzt nicht so im Physikunterricht macht. Das Kriterium ist ja nicht, wie viel Physik man dabei lernt. Es wird ja eher so geschaut, ob das alles physikalisch richtig ist und was ist der Hintergrund dazu, also was hat sich die Autorin dabei gedacht und das ist kreativ. In der Elementarphysik ist es ja nicht immer so, dass die Dinge so einfach sind zu bestimmen, ob das richtig ist. In der Schulphysik ist das oftmals nicht klar, weil die Begriffe oft nicht ganz stimmen. Ein Beispiel ist das mit dem Sonnentaler, das kann man ja rein geometrisch erklären. Es wird ja immer von Transport von Licht geredet und das Licht bewegt sich mit Lichtgeschwindigkeit. Da frage ich meine Studenten immer, ob sie mir ein Instrument sagen können, mit dem man genau das messen kann. Und das können sie nie. Es gibt ein einziges und das ist der Drehspiegelversuch, bei dem man die Lichtgeschwindigkeit messen kann, aber der wird in der Schule oft nicht gezeigt. Man kann an so einem einzigen Versuch wie beim Sonnentaler viel erklären und da viel reinstecken. Bei der

Auswahl der Experimente ist das sehr schwierig im Wald, denn da hat man keinen Strom, also fallen schon mal all diese Experiment weg und auch alle die irgendeinen Antrieb haben. Optik kann man gut in der freien Natur machen. Und man kann auch was mit Mechanik machen.

Mechanik habe ich mir auch überlegt. Und ich habe mir gedacht, dass das Opfer erschossen wird und man dann die Geschwindigkeit der Kugel berechnet. Und die Frage ist, ob das dann nicht zu schwer ist.

Bei so einer Rallye kann man ja auch eine Aufgabe machen, die ein bisschen schwieriger ist. Das darf halt nicht dazu führen, dass jemand, der die Aufgabe nicht lösen kann, trotzdem die App löst. Man kann das ja auch mit Pfeil und Bogen machen. Der Pfeil fliegt dann auch näher zum Boden und nicht dorthin, wo man hingezielt hat. Das passiert auch bei Gewehren, besonders bei grossen Distanzen.

Ich kann ja auch die Formel angeben.

Ja, das kann man schon machen. Einfach nicht immer, weil das Umformen nicht allen so leichtfällt. Das kannst du ja mit einer Freundin Probe laufen lassen.

Ja, das mach ich sowieso.

Du hast dir ja das Thema ausgesucht und dann bist du gleich in der Rolle der Physiklehrerin. Du musst ja die App an die Menschen auch anpassen. Bau lieber auf dem gesunden Menschenverstand auf, als dass du jetzt im Lehrplan nachschaust, was z.B. ein 14-Jähriger in Physik können muss. Konzentrier dich auf den Spass.

Wie definieren Sie Spass?

Wenn die Leute daran Freude haben und das weiterempfehlen. Wenn jetzt jemand Fremdes diese Rallye macht und sie abschliesst, dann kann man ja davon ausgehen, dass die Person Spass hatte. Sobald da jemand positiv drauf reagiert, hatte die Person Spass.

Wie sind Sie denn auf Ihren Beruf gekommen? Hatten Sie schon immer gerne Physik?

Also für wissenschaftliche Dinge habe ich mich schon lange interessiert. Auf den Chemie- und Physikunterricht habe ich mich gefreut. Ich habe auch immer gerne diese Was ist Was-Bücher gelesen und hatte auch viele von diesen Büchern. Ich habe mich auch eher zuerst für Chemie interessiert. Ich hatte dann auch ein kleines Labor in der Garage. Man konnte sich damals auch noch in der Apotheke diese Chemikalien kaufen. Und dann habe ich mal ein Praktikum in einer pharmazeutischen Firma gemacht und habe so vier Wochen als Laborpraktikant gearbeitet. Es war sehr spannend, aber das hat mich so ein bisschen von der Chemie weggerückt und mehr zu der Physik hin. Und dann wollte ich studieren nach der Schule. So etwas zwischen Chemie und Maschinenbau. Ich habe mich dann informiert und so; und ich habe mich dann doch für Physik entschieden. Aber nicht so eindeutig, eher so aus dem Bauch heraus, aber es war immer das Richtige und ich würde es auch nochmal machen. Physiklehrer war eigentlich nur mein Plan B. Das hat sich halt dann so entwickelt. Ich habe halt immer viel herumgeschaut und mir die Möglichkeiten offengehalten.

Und was machen Sie jetzt so?

---

Jetzt bin ich an der Uni und bin zuständig für die Physiklehrerausbildung, und in der Lehre muss ich denen Didaktik der Physik beibringen. Also so, dass die Schüler das auch verstehen. Die lernen auch die Experimente richtig aufzubauen. Heutzutage besteht ja auch die Tendenz, auch mal lieber ein YouTube-Video zu zeigen. Das ist für mich aber nicht das Gleiche. Ich finde es schade, dass Versuche, die man problemlos in der Schule machen kann, dann als Video oder Animation gezeigt werden. Wir machen auch ein bisschen Forschung und versuchen neue Experimente zu entwickeln, z. B. zur Quantenphysik, dass man dieses Thema auch im Unterricht zeigen kann. Aber das ist halt sehr schwierig und aufwändig.

**Was gefällt Ihnen an der Physik am besten?**

Dass es immer etwas zu denken gibt. Auch bei einfachen Problemen gibt es oft eine Einsicht, wenn man die Lösung gefunden hat. Und Physik ist überall in der Welt. Und es gibt so viele verschiedene Gebiete, die alle miteinander zu tun haben und dass man nie fertig ist mit der Lösung. Ja, und auch, dass man Dinge ausprobieren kann.

**Wie definieren sie Physik?**

Physik ist erst mal eine Naturwissenschaft. Spontan könnte man das an den Teilgebieten zu definieren. Man muss das ja von der Biologie und der Chemie abgrenzen. Eine Wissenschaft, die versucht Gesetze aufzustellen, die die Natur möglichst gut beschreiben.

### 13.5.3. Interview mit Fritz Gassmann

Interviewpartner: Dr. Fritz Gassmann, Vorsitzender der Naturforschenden Gesellschaft Zürich

Datum: 20. Juni 2018

Dauer 2 h

Ort: Kantonsschule Wiedikon

**Rosina Weiß: Vielen Dank, dass Sie sich Zeit nehmen und zu mir an die KWI gekommen sind, damit ich Sie interviewen kann! Darf ich das Gespräch aufzeichnen?**

Fritz Gassmann: Ja sicher! Ich bin sehr gerne hier und helfe dir bei deiner Maturitätsarbeit.

**(Ich habe erzählt, was ich in meiner Arbeit genau mache.)**

**Was gibt es für einfache Experimente, die man in der Natur machen kann?**

Haben Sie sich schon etwas überlegt? So in welche Richtung?

**Man kann viel mit Optik machen und es gibt das Wasser in der Nähe. Und ich habe mir überlegt, dass das Opfer mit einer Pistole umgebracht wird und man könnte dann die**

Austrittsgeschwindigkeit berechnen. Das würde dann einfach schon zu den schweren Aufgaben gehören.

Ja, aber allgemein mit Geschwindigkeit würde das schön gehen. Denn jeder weiss, was auf dem Tachometer vom Auto steht und dann kann man dort anfangen, dass es Weg pro Zeit ist und so die Formel herleiten. Das kennt jeder.

Was ich mir auch überlegt habe, ist auch, dass ich die Formel der Geschwindigkeit auch angeben könnte, wenn man sich z. B. einen Tipp holt.

Ja, das könnte man auch machen.

Ich möchte auch die Stationen vom Erlebnispfad mit einbeziehen. Und im Sihlwald darf man auch abseits der Wege wandern. Dann könnte man auch dort Dinge verstecken. Kennen Sie vielleicht noch Experimente, die man in der Natur durchführen könnte?

Also ich weiss jetzt nicht, wie man das einbauen könnte, aber es ist ja so, dass Geräusche an Mauern und manchmal auch Bäumen reflektiert werden. Und da Bäume rund sind, werden nur die höheren Frequenzen reflektiert und das hört sich dann manchmal seltsam an. Die Leute wissen oft nicht, was das ist. Im Auto hört man das gut, wenn man durch eine Allee von Bäumen fährt und dann das Fenster herunterlässt. Es gibt sogar Blinde, die diese Technik entwickelt haben. Im Internet gibt es da auch Videos. Da fährt er mit dem Fahrrad auf einem Parkplatz mit vielen Autos und macht die ganze Zeit klack klack klack. Und dadurch kann er hören, wo sich Hindernisse befinden. Also wie eine Fledermaus.

Was mir noch einfällt ist, dass wenn es feucht ist im Wald und die Sonne so schräg scheint, dann sieht man den Strahl. Man sieht die Reflexion der Wassertropfen. Bei einem Wasserfall geht das natürlich auch, aber dafür muss ja die Sonne scheinen. Mit einer Taschenlampe könnte man es auch sehen, aber das müsste man dann probieren. Und eine Sache kann man sicher machen, und zwar einen Stein irgendwo fallen lassen und dann hören, bis es Plumps macht, um die Tiefe zu bestimmen.

Es gibt auch eine Brücke, wo man den Stein herunterwerfen kann. Mit dem Ausrechnen ist das dann einfach ein bisschen zu schwer.

Wenn man eine Kugel eine schiefe Ebene herunterrollen lässt und dann schaut, wie viel sie in einer, zwei, drei Sekunden zurücklegt. Dann kann man sehen, dass es immer das Doppelte ist. So könnte man es herleiten. Dann muss man keine Formel angeben.

Etwas mit Schallgeschwindigkeit könnte man auch machen. Dass man etwas anderes hört, als man sieht, halt eben versetzt. Dann kann man auch die Distanz berechnen.

Ich möchte auch einen Fussabdruck in Beton herstellen. Und dann kann man über die Tiefe berechnen, wie schwer die Person ist, und so entscheiden, ob Mann oder Frau. Und die Schuhgrösse kann man auch herausfinden.

Ich wollte auch etwas mit Auftrieb im Wasser machen.

Was in den Krimis vorkommt, ist ja, dass man die Leiche im Wasser findet. Dann kann man mögliche Tatorte herausfinden, wo die Person ins Wasser gefallen ist.

Das Opfer wird auf der Brücke erschossen und fällt dann ins Wasser.

Ah ja, dann muss man sowieso einen Spaziergang machen, das passt.

---

Mit Seifenblasen könnte man vielleicht etwas machen. Ich habe gelesen, dass sie unten anders sind als oben, weil sie herunterfließen. Und man sieht alles verkehrt herum.

Optische Sachen sind auch sehr einfach transportierbar. Mit einer Linse kann man auch etwas anzünden. Etwas mit Farben wäre vielleicht auch keine schlechte Idee. Wieso ist der Himmel blau? Und der See ist blau, weil er den Himmel reflektiert.

Man kann ja auch Wissensfragen einbauen und nur wenn man sie löst, bekommt man neue Information zum Fall.

(Wir haben weiter über verschiedene Experimente diskutiert, welche man ausprobieren könnte. Auch haben wir über einige irrelevante Themen für meine Arbeit gesprochen.)

Vielen Dank für das Gespräch!

### 13.6. Arbeitsjournal

Anschliessend folgt mein Arbeitsjournal, das ich im Laufe meiner Arbeit von Dezember 2017 bis Dezember 2018 geführt habe.

## Arbeitsjournal Maturitätsarbeit Rosina Weiß

Datum	Dauer	Thema	Was habe ich gemacht	Bemerkung
Dezember		Themenwahl	Ich habe mein Thema festgelegt und mir einen entsprechenden Betreuer gesucht. Mir war schnell klar, dass ich etwas in der Natur machen möchte und dass ich gerne eine App programmieren möchte.	Ich wollte mir ein Thema aussuchen, an dem ich wirklich Spass habe, mich ein ¼ Jahr damit zu beschäftigen.
Februar	9 h	Virtual Machine	Ich wollte eine App für iOS programmieren. Dafür brauche ich aber ein MacOS-Betriebssystem mit Xcode. Also habe ich auf meinem Windows PC eine Virtual Machine eingerichtet, auf der nun MacOS läuft.	Ich hatte einige Schwierigkeiten dabei, um zu den Dateien entsprechende Anleitungen zu finden, da ich keine Ahnung hatte, wie man das genau macht. Auch war es schwer, die passenden Tools zu finden, und ich hatte viel Wartezeit währenddessen alles installiert wurde.
14. März	1 ½ h	Bücher	Ich bin an die PH Zürich gegangen und habe mir Bücher zu dem Thema Physik im Unterricht ausgeliehen.	Ich war das erste Mal in der PH und habe ein Weilchen gebraucht, um mich gut zurechtzufinden.
25. März	1 ½ h	Location	Ich bin an der Sihl um die Zugstation Wildpark-Höfli spazieren gegangen und habe nach einer möglichen Rundtour für meine Rallye gesucht.	Um eine Tour zu haben, die dort endet, wo sie anfängt, bräuchte ich zwei Brücken, die in einer gewissen Nähe sind, damit man die Sihl überqueren kann. Dies war in Wildpark-Höfli nicht der Fall.
8. April	2 h	Location	Ich bin im Sihlwald spazieren gegangen und bin dort den Walderlebnispfad entlanggelaufen. Auch habe ich hier für	Ich habe eine ungefähre Route gefunden. Der Start und der Schluss meiner Tour sind beide am Bahnhof

			eine mögliche Tour für meine Arbeit gesucht. Ich habe zudem auch Fotos von möglichen Orten für Stationen gemacht. Schliesslich habe ich mich für diese Location entschieden.	Sihlwald. Zudem gibt es zwei Brücken in der Nähe, ein Naturmuseum im Besucherzentrum im Sihlwald mit einem Restaurant. Im Wald gibt es den Walderlebnispfad, den ich gut in meine Rallye integrieren kann.
21. April	1 ½ h	Design schriftliche Arbeit	Ich habe Formatvorlagen auf Word erstellt und das Seitenlayout eingerichtet.	Frage: Fussnoten oder Verweise beim Zitieren? Habe ich ein Maximum oder Minimum an Wörtern oder ist das egal? Zählt der Anhang da hinein?
24. April	3 h	Bücher	Ich habe das Buch Physik in der Natur grob durchgearbeitet, um mir Inspiration für die Physikstationen zu holen.	Leider kann man viele interessante Experimente oder Phänomene nicht so gut in der Natur durchführen, da sie spezielle Geräte brauchen oder auch Strom, der in der Natur nicht vorhanden ist.
26. April	3 ½ h	Bücher	Ich habe die restlichen Bücher aus der PH durchgearbeitet und mir Notizen zu den möglichen Physikexperimenten gemacht.	Hier war das Problem, dass die Bücher entweder viel zu einfach waren, also für die Primarschule, oder sie waren viel zu schwer, also für die Universität. Ausserdem ist es recht kompliziert zu wissen, was mit der Kriminalfall schlussendlich kompatibel sind.
27. April	1 h	Programmieren	Ich habe mich ein bisschen in die Programmiersprache Swift durch Xcode eingearbeitet und ein bisschen herumexperimentiert.	Ich musste zuerst ein komplettes Systemupdate machen, um die aktuellste Version von Xcode zu benutzen.

28. April	1 ½ h	Regeln im Sihlwald	Ich habe mich über das Internet erkundigt, welche Sachen man im Sihlwald machen darf, z. B. ob man die offiziellen Wege verlassen darf, und mir die allgemeinen Regeln des Sihlwaldes angeschaut. Auch habe ich mich genauer zum Walderlebnispfad erkundigt.	Ich habe herausgefunden, dass man die Wege in der Naturerlebniszone (keine Kernzone oder Naturschutzgebiet) verlassen darf, was ich selber sehr cool finde. Denn so kann man z.B. auch Verstecke abseits der Wege finden etc. Jedoch muss ich noch herausfinden, ob es möglich ist, Schilder oder Ähnliches aufzustellen.
29. April	1 ½ h	Bücher	Im Internetrechercheportal der ZB habe ich weitere Bücher herausgesucht. Speziell habe ich nun auch Bücher über Ballistik oder wie man einen Krimi schreibt.	Leider muss man in der ZB im Internet die Signaturen schon heraussuchen und kann nicht einfach in der Bibliothek alle Bücher anschauen und die Besten heraussuchen, da sie nach Bestandsaufnahme sortiert sind. Somit konnte ich nicht so gut beurteilen, ob die Bücher die sind, die ich benötige oder nicht.
2. Mai	1 h	Mordstory	Ich habe erste Ideen für den Kriminalfall gesammelt und auch ein bisschen recherchiert.	Einige Ideen finde ich grossartig und ich kann mir gut vorstellen, dass man sie gut umsetzen kann. Jedoch kann man sich jetzt noch nicht zu genau festlegen, da die Hintergrundgeschichte zu den Posten entsprechend passen sollte.
7. Mai	2 h	Bücher	In der ZB habe ich die schon herausgesuchten Bücher ausgeliehen.	Auch hier war ich das erste Mal in der ZB und musste mich erstmal zurechtfinden.

5. Mai	½ h	Interview	Ich habe Mails an Fritz Gassmann (Physiker) und Roger Graf (Autor von den haarsträubenden Fällen des Philip Maloney) geschrieben und sie um ein Interview gebeten.	Fritz Gassmann hat mir recht schnell zurückgeschrieben und wir haben nun ein Treffen am 20. Juni abgemacht. Leider ging es ihm nicht früher.
9. Mai	½ h	App Design	Ich habe mir ein paar Gedanken zum Design meiner App gemacht. Denn ich möchte ein paar Buttons haben, die sozusagen auf jeder Seite sind und mit denen man dann z. B. zu den Einstellungen oder der Karte kommt.	Ich habe zwar noch nicht genau herausgefunden, wie man ein Grunddesign in einer App erstellt, aber ich glaube, das werde ich schon schaffen.
10. Mai	½ h	Posten	Ich habe mir grobe Gedanken zu den verschiedenen Posten gemacht.	Es sind immer noch solche Fragen wie: Wo ist der Einstieg ins Thema? Braucht man Vorwissen? Ist es nur auf dem Schulstoff aufbauend? Man muss die Stationen auch in einer Reihe aufbauen, dass das Vorwissen für gewisse Themen schon vorhanden ist.
12. Mai	2 h	Foxtrail	Wir haben heute einen Foxtrail in Zürich gemacht. Dort sind mir viele Methoden für einzelne Stationen in Erinnerung geblieben. Z. B. hat man einen Text der gespiegelt ist und dann muss man einen montierten Spiegel finden, um den Text richtig lesen zu können.	Ich habe einige Anregungen für meine Arbeit bekommen und auch einiges im Bereich Teamwork gelernt.
21. Mai	2 h	Programmieren	Ich habe mich in mein Swift 4 Handbuch hereingelesen und Grundlagen gelernt.	Das Handbuch, das ich mir gekauft habe, ist sehr dick und dies hat mich am Anfang ein bisschen erschreckt. Doch habe ich dann gesehen, dass ich

				nicht alles, was dort drin geschrieben steht, wissen muss.
27. Mai	2 ½ h	Programmieren	Ich habe ein bisschen mit einem Quizlayout experimentiert und habe herausgefunden, wie man Bilder einfügt und sie wechseln kann. Auch habe ich eine Tonspur eingefügt.	Den Code für die Tonspur habe ich aus einem Buch kopiert und leider habe ich mit meinen jetzigen Kenntnissen nicht genau verstanden, wie das geht. Aber ich denke optimistisch und hoffe, dass ich dies bald auch verstehen werde.
30. Mai	1 h	Schilder	Ich habe heute Nachmittag herumtelefoniert, ob ich gewisse Dinge, wie Schilder usw. im Sihlwald deponieren könnte. Schliesslich habe ich die richtige Ansprechperson gefunden, welche mir aufgetragen hat ein Exposé zu schreiben.	Ich habe etwas gebraucht um die richtige Ansprechperson zu finden, jedoch waren alle Menschen am Telefon sehr nett zu mir und haben mir gut weitergeholfen.
2. Juni	1 ½ h	Exposé	Ich habe angefangen, das Exposé für den Wildnispark Zürich zu schreiben, um das Aufstellen von Schildern oder anderen Dingen zu genehmigen.	Ich weiss nicht ganz genau, was alles in so ein Exposé hineingeschrieben werden muss. Zudem muss ich genau schildern, welche Infrastrukturveränderungen ich vornehmen würde. Jedoch weiss ich noch nicht genau welche, da meine einzelnen Stationen noch nicht festgelegt wurden. Doch ich habe einige Ideen, die ich einbauen werde. Ausserdem nehme ich lieber zu viel erstmals hinein und lasse später noch etwas weg, als nachträglich noch etwas beantragen zu müssen.

2. Juni	¼ h	Digitalisierung Arbeitsjournal	Ich habe das handschriftliche Arbeitsjournal digitalisiert.	Jetzt habe ich einen besseren Überblick als auf dem handgeschriebenen Papier. Zudem geht es in Zukunft dann schneller.
3. Juni	1 ½ h	Exposé	Ich habe das Exposé, das ich gestern angefangen habe, fertiggeschrieben und verschickt.	Ich hoffe sehr, dass mein Antrag bewilligt wird.
9. Juni	1 h	Interview	Ich habe das Interview per Videokonferenz mit Heike Riel geführt.	Heike war sehr nett und hat mir sehr geholfen. Vor allem bei der Hintergrundstory hat sie mir viele hilfreiche Anregungen gegeben.
9. Juni	¾ h	Interview	Ich habe angefangen, das Interview mit Heike Riel, welches ich aufgezeichnet habe, abzutippen.	Ich finde, dass das Abtippen sehr mühsam ist. Es dauert viel, viel länger als das Interview selbst.
10. Juni	1 h	Interview	Ich habe das Interview mit Jan-Peter Meyn per Telefon geführt.	Herr Meyn hat viel über Didaktik erzählt und dass ich die Physik in meiner Krimi-App sehr einfach halten sollte, damit es immer noch Spass macht. Frage: Sollte ich die Physik sehr simpel halten, sodass jeder Otto-Normalbürger die Rallye mit viel Spass lösen kann oder möchte ich etwas anspruchsvollere Physik auswählen, mit schwereren Aufgaben und daher auch schon ein wenig Vorwissen?
10. Juni	1 ½ h	Interview	Ich habe das Interview von gestern, also mit Heike, weiter abgetippt.	

14. Juni	½ h	Besprechung	Herr Torriani und ich hatten eine Besprechung.	
20. Juni	2 ½ h	Interview	Ich habe ein Interview mit Dr. Friz Gassmann im Schulhaus durchgeführt.	Ich habe viele Ideen und Anregungen für die Physikposten bekommen.
13. Juli	1 ½ h	Physik	Ich habe in den Büchern aus der ZB nach möglichen Experimenten gesucht und diese gesammelt.	Leider gibt es viele spannende Experimente, die viel Material benötigen, welches man nicht so gut im Wald deponieren kann.
14. Juli	½ h	Interview	Ich habe das Interview mit Jan-Peter Meyn abgetippt.	
15. Juli	½ h	Koordinationsblatt	Ich habe das Koordinationsblatt vom Wildnispark Zürich ausgefüllt.	Frau Aebli, eine Rangerin im Sihlwald, hat mir eine Mail geschrieben, mir mitgeteilt, dass ich mein Projekt sicher im Sihlwald durchführen kann und ich bitte das Koordinationsblatt ausfüllen soll, damit mein Projekt dort offiziell gemeldet ist.
18. Juli	1 h	Gespräch	Ich hatte ein Gespräch mit Frau Aebli über das Ausführen meiner Arbeit und über die Umsetzung im Gelände.	Ich darf alles so positionieren, wie ich es mir vorgestellt habe. Es sollte bloss nicht zu gross sein und leicht entfernbar sein. Bevor ich die Sachen deponieren werde, soll ich Frau Aebli die genauen Standorte mitteilen, damit sie nochmals überprüfen kann, ob keine Verletzungsgefahr für Tiere im Wald besteht.
20. Juli	1 h	Physik	Ich habe alle Ideen für die Physikposten zusammengetragen und mir grob	Ich finde, dass es sehr schwer ist, für alles einen Zusammenhang zur

			überlegt, inwiefern ich sie mit der Mordstory verbinden könnte.	Hintergrundstory zu finden. Doch ich finde, das muss auch nicht bei jedem Posten der Fall sein.
21. Juli	2 h	Programmieren	Ich habe weiter das Programmieren ausprobiert und erarbeitet.	Ich habe endlich genau verstanden, wie das mit dem Layout auf den verschiedenen Geräten mit verschiedenen Bildschirmgrößen funktioniert.
23. Juli	2 h	Mordstory	Ich habe angefangen, mir die Schlüsselemente der Mordstory zu überlegen und festzusetzen.	Ich finde, es ist schwer so zu gestalten, dass es nicht zu logisch und einfach für den Aufklärer ist. Deswegen habe ich mir mehrere Personen überlegt, die ein Motiv haben und der Mörder sein könnten. Dann kann ich darauf aufbauend entscheiden, wer sich am besten dazu eignen würde, der Mörder zu sein.
25. Juli	2 h	Programmieren	Ich habe weiter mit Xcode gearbeitet.	
26. Juli	2 h	Physik	Ich habe angefangen, die Physikposten auszusortieren und mich zu entscheiden, welche ich nehme und welche nicht.	Ich musste ein paar lustige Experimente streichen, da ich auch möglichst einige Themen der Physik anschneiden möchte und ich sonst zu einem Thema zu viele Posten hätte.
26. Juli	¼ h	Interview	Ich habe das Interview von Heike Riel fertig abgetippt.	
28. Juli	1 h	Physik	Ich habe die Posten fertig aussortiert und noch richtig ausformuliert und an Herr Torriani geschickt.	Die Reihenfolge der Posten ist noch unklar, da dies abhängig von Ort und Mordstory ist.

31. Juli	2 h	Programmieren	Heute habe ich mit Navigation-Bars beschäftigt.	Wahrscheinlich werde ich die Pfeilfunktion nicht verwenden, da dies recht unübersichtlich werden kann. Doch ich finde die Titelfunktion sehr praktisch.
3. August	2 h	Story	Ich habe die Antworten auf die fünf Leitfragen für die Mordstory bestimmt und Motiv, Täter, etc. festgesetzt.	Ich habe zwei Varianten, wer der Täter sein könnte. Aber ich tendiere zur Ehefrau.
5. August	4 h	Virtual Machine	Eigentlich wollte ich heute weiterprogrammieren. Doch meine Virtual Machine hat nach einem Update der Firmware in der endlosen Bootschleife festgesteckt verharrt und ich musste sie reparieren bzw. neu installieren.	Das Installieren ist immer sehr mühsam. Schliesslich musste ich die Software wechseln, da die alte Software VMware macOS nicht mehr unterstützt. Bei VirtualBox gibt es leider keine Guest Additions, also funktioniert der USB-Anschluss nicht. Aber ich hoffe, dass ich dieses Problem noch beheben kann.
8. August	2 h	Programmieren	Ich habe die Knöpfe nextScreen und lastScreen eingestellt und die Segues eingerichtet.	
13. August	2 h	Programmieren	Ich habe den Hilfebutton erstellt und in der PhysPosten-Klasse definiert.	Ich habe leider noch nicht herausgefunden, wie man ein kleines Fenster erstellt, das aufgeht, wenn man auf den Knopf drückt.
14. August	2 h	Story	Ich habe das Buch Businessplan Mord gelesen, um den Aufbau beim Erstellen und der Planung eines Mordes genauer zu realisieren.	

16. August	1 h	Programmieren	Ich habe die Info-Seite eingerichtet. Es fehlt zwar noch der endgültige Text, doch dieser kann am Schluss einfach eingefügt werden.	
17. August	2 h	Programmieren	Heute habe ich auf der Navigationsseite eine Karte mit Standortbestimmung eingerichtet.	Leider habe ich erst als ich fertig war herausgefunden, dass die Karte ohne Internetverbindung nicht funktioniert. Vielleicht könnte dies ein Problem werden, wenn es in gewissen Stellen im Wald kein Netz gibt. Es könnte mit einem Kompass vielleicht besser funktionieren.
27. August	5 h	Programmieren	Heute Nachmittag habe ich angefangen die Multiple-Choice-Seite zu programmieren.	Leider hat es ganz am Schluss einen Bug gegeben und ich konnte den Fehler nicht finden und mir auch nicht erklären, wie es dazu gekommen ist. Also musste ich eigentlich alles nochmal neu machen.
2. September	7 h	Programmieren	Heute habe ich probiert, eine Art Kompassfunktion nach meinem Swiftbuch zu programmieren. Hierbei habe ich die Grundfunktionen übernommen, musste jedoch vieles auf meinen Fall abändern.	Am Anfang habe ich nicht wirklich verstanden, wie die Autoren es in meinem Buch machen. Deshalb musste ich mich eine Zeit lang erst einmal einarbeiten, bis ich verstanden habe, wie die Funktion funktioniert.
8. September	3 h	Mordstory	Ich habe weiter an meiner Mordstory gearbeitet und mir verschiedene Details überlegt. So bin ich auf die Idee gekommen, Tatortfotos zu schiessen. Dann habe ich mich darum gekümmert,	

			mit wem ich diese machen könnte. Dazu habe ich verschiedene Leute angefragt.	
18. September	1 h	Polizei	Heute war ich bei der Polizei in Zürich und konnte, wie in der Mail abgemacht, Absperrband und Nummerntafeln abholen, damit die Tatortfotos authentischer wirken.	
20. September	1 h	Fotos	Ich habe mir nochmals genau aufgeschrieben, welche Dinge wo auf die Fotos müssen und geschaut, dass alles vorhanden ist.	
22. September	4 h	Fotos	Heute habe ich die Tatortfotos mit Herrn Emery aufgenommen. Am Morgen bin ich extra vorher in den Sihlwald gefahren und habe zuerst noch nach der idealen Position gesucht.	Das "Foto-Shooting" selber ging eigentlich sehr schnell, doch das Vorbereiten war sehr aufwändig. Denn man musste den perfekten Ort finden und alles aufstellen und auch schauen, ob man nichts vergessen hat.
3. Oktober	4 h	Verstecke	Ich habe mir überlegt, für welche Posten ich Verstecke brauche. So habe ich diese auch mit der Mordstory grob verknüpft und eine Reihenfolge festgelegt.	
8. Oktober	5 h	Programmieren	Ich habe die Notizbuchfunktion angefangen und mir überlegt, wie man diese koordinieren kann.	
9. Oktober	6 h	Programmieren	Heute habe ich weiter an meiner Notizbuchfunktion gearbeitet.	
10. Oktober	6 h	Programmieren	Ich habe mich mit ScrollViews beschäftigt, welche es erlauben, mehr auf einen View	Hierbei ist das Problem aufgetreten, dass obwohl ich eigentlich alles

			zu tun und dann hinunter zu scrollen. Dafür habe ich mehrere Videos geschaut und schliesslich in meiner Postenübersicht probiert, die Scroll-Funktion einzubauen.	richtiggemacht hatte, es schlussendlich in der Testversion auf meinem Handy nicht funktioniert hat. Irgendwo war ein winziges Detail falsch, ich wusste leider nicht welches. So habe ich alle verschiedenen Varianten, die es gibt, ausprobiert und oft auch mehrere Varianten viele Male neu gemacht, bis es endlich funktioniert hat.
12. Oktober	5 h	Programmieren	Ich habe weiter an meiner Quizfunktion gearbeitet und auch daran, wie diese eingespeichert werden können.	
14. Oktober	3 h	Verstecke	Ich bin heute in den Sihlwald gefahren und habe auf dem geplanten Weg geeignete Verstecke für die geplanten Posten, welche Material benötigen, gesucht. Wenn ich ein geeignetes gefunden habe, habe ich mir die entsprechenden Koordinaten notiert und Fotos geschossen.	Hier war ein Problem, dass die Koordinaten auf der Kompass-App meines Handys sehr ungenau sind und ich für die abschliessende Position eine bessere Möglichkeit finden muss, um die Position in meine App einzubauen.
17. Oktober	1 ½ h	Verstecke	Ich habe die am Wochenende gesuchten Verstecke in einem Dokument zusammen mit den Bildern gespeichert und Frau Nicole Aebli gesendet, damit sie diese überprüfen kann, ob sie in Ordnung sind.	
18. Oktober	5 h	Programmieren	Ich habe für meine App die Speicherfunktion programmiert. So ist die App, wenn man sie verlässt, nicht wieder auf den Anfang zurückgesetzt, sondern die	

			Position, an der man stehengeblieben ist, wurde gespeichert.	
20. Oktober	6 h	Programmieren	Ich habe den Posten 1 in meine App eingefügt. Das heisst ich habe die Grundfunktionen der App für den Nutzer erklärt und sie in den Mordfall eingeführt.	Es geht immer sehr lang, die Constrains zu setzten, denn wenn sie nicht richtig sind, gibt es immer eine Fehlermeldung.
22. Oktober	5 h	Programmieren	Ich habe vom Posten 1 den ScrollView mit den Tatortfotos eingefügt.	Auch hier hatte ich wiederum viele Probleme mit dem ScrollView. Er wollte einfach nicht das machen, was man ihm gesagt hat. Schlussendlich habe ich es dann aber nach mehrfachem Neuprogrammieren geschafft.
23. Oktober	4 h	Programmieren	Ich habe nun auch eine Speicherfunktion für mein Notizbuch eingefügt, damit man dieses updaten kann und man auch immer an den selben Ort zurückkommt.	
27. Oktober	5 h	Programmieren, Fotos	Ich habe mir überlegt, wie ich die Fotos, die ich am Dienstag anfertigen werde, genau haben möchte und was ich dafür für Material noch zusätzlich brauche. Danach habe ich noch programmiert.	
30. Oktober	3 ½ h	Fotos	Ich habe die Fotos mit Cinja und Kerrin für meine Verdächtigen gefertigt. Insgesamt waren es fünf Fotos, für die ich jetzt mehr als genug zur Auswahl habe.	Ich fand dieses „Foto-Shooting“ sehr spassig und die beiden haben ihre Sache sehr gut gemacht, wie man auf den eingebundenen Fotos auch sehr gut erkennen kann.
3. November	5 h	Programmieren, Mordstory	Ich habe an der Navigationsseite gearbeitet und die Methode mit dem Pfeil	Hier ist es schwer zu wissen, wie viele Informationen zu viel sind und wie

			<p>noch optimiert und schöner dargestellt. Dabei habe ich verschiedene Tests bezüglich der Koordinatengenauigkeit durchgeführt. Dann habe ich mir genau überlegt, wo welche Hinweise zur Mordstory dargelegt werden. Dafür habe ich alle Infos auf kleine Zettel geschrieben und diese dann sortiert, so dass am Schluss alles Sinn ergeben sollte.</p>	<p>viele zu wenig sind, damit es für den Nutzer nicht zu einfach aber auch nicht zu schwer ist.</p>
4. November	3 h	Papierflieger	<p>Ich habe fünf verschiedene Papierflieger mehrmals gefaltet und draussen im Garten ausprobiert, welche wie weit fliegen oder besonders auffällig sind. Dazu habe ich mehrere Versuche durchgeführt und Fotos gefertigt.</p>	<p>Es kommt sehr darauf an, wie man den Papierflieger wirft, so sind viele immer verschieden weit gekommen. Doch der Kunstflieger war sehr auffällig, da er in Kreisen geflogen ist. Am Schluss waren zwei Papierflieger auf dem Dach.</p>
4. November	2 h	Fotos	<p>Ich habe die Fotos vom Bruder des Opfers geschossen und anschliessend alle Fotos, die ich habe, aussortiert und jeweils immer ein bestes herausgesucht.</p>	
5. November	4 h	Fotos	<p>Ich habe nun die gestern Abend ausgewählten Fotos bearbeitet. Bei vielen musste ich nur den Rand etwas zuschneiden oder die Helligkeit ändern. Doch bei den Verhaftungsfotos habe ich eine Vignette über den Fotos positioniert und schliesslich mit Photoshop die Namen auf die schwarze Tafel geschrieben.</p>	<p>Beim Fotografieren der Tatortfotos mit Herrn Emery habe ich leider eine falsche Überlegung angestellt. Ich habe gedacht, wenn das Opfer erschossen wurde, gibt es ein Loch im T-Shirt durch die Kugel, also habe ich mit der Schere ein Loch in das Shirt geschnitten. Als ich später das Blut auf das Shirt geschmiert habe, ist mir dann aufgefallen, dass man dort, wo das Loch</p>

				ist, die Haut erkennen kann und diese nicht rot durch das Kunstblut gefärbt ist, weil das Wasser der Sihl das Kunstblut verwässert hat. So gibt es mitten im Blutfleck einen weissen Fleck. Dieses Problem habe ich jetzt behoben, indem ich den weissen Fleck mit Photoshop bearbeitet habe.
7. November	3 h	Programmieren	Heute habe ich wieder einen Posten in meine App eingefügt. Dieser hat wiederum ein Quiz, für welches ich ein ScrollView eingebaut habe. Dies hat wieder sehr lange gedauert, bis es endlich funktioniert hat.	
8. November	3 h	Mordstory, Programmieren	Ich habe noch einmal über die Mordstory nachgedacht und habe mir überlegt, ob alles aufgeht. Danach habe ich noch etwas programmiert.	
9. November	1 h	Besprechung	Heute hatte ich wieder eine Besprechung mit Herrn Torriani. Ich habe ihn auf den neusten Stand gebracht.	
10. November	3 h	Mordstory	Ich habe die Mordstory noch etwas optimiert und gewisse Teile verschoben, gelöscht oder neu hinzugefügt.	
11. November	4 h	Programmieren	Ich habe heute wieder etwas programmiert und an einem Posten wieder ein paar Screens hinzugefügt.	

12. November	3 h	Schriftliche Arbeit, Programmieren	Zuerst habe ich ein kleines Kapitel in meiner schriftlichen Arbeit geschrieben und dann habe ich noch programmiert.	
13. November	4 h	Bücher	Heute habe ich viele Bücher, die ich mir aus der ZB oder der PH ausgeliehen habe, gelesen und mir verschiedene Zitate herausgesucht, welche ich in meine schriftliche Arbeit einbinden könnte.	Ich habe manche Quellen nicht so schnell wiedergefunden. Ich hätte sie gleich in meine Arbeit einbinden sollen.
16. November	4 h	Programmieren	Ich habe den letztens angefangenen Posten fertig abgeschlossen.	
19. November	7 h	Schriftliche Arbeit	Zuerst einmal habe ich mir überlegt, wie ich meine Arbeit genau strukturieren möchte. Ich habe zwar schon ein paar Kapitel geschrieben, doch die Frage ist, in welcher Reihenfolge diese vorkommen sollen. Also habe ich ein genaues Inhaltsverzeichnis erstellt, sodass ich nur noch den Text zu den einzelnen Kapiteln schreiben muss und nicht mehr gross überlegen muss, was alles wo hineinkommt. Deshalb habe ich dann auch zu jedem Kapitel kurz in Stichworten hingeschrieben, was es etwa enthalten soll. Dann habe ich auch schon einige Kapitel angefangen zu schreiben.	Jetzt fühle ich mich wohl, denn meine schriftliche Arbeit fängt auch an zu gelingen.
20. November	6 ½ h	Schriftliche Arbeit	Ich habe einige Kapitel zum Endprodukt und auch zum Vorgehen geschrieben.	

21. November	4 h	Schriftliche Arbeit	Ich habe vor allem die einzelnen Posten zusammengefasst und angefangen, das Kapitel zur Mordstory zu schreiben.	
23. November	7 h	Schriftliche Arbeit	Heute Abend wollte ich eine grobe erste Version meiner schriftlichen Arbeit fertigmachen, damit meine Mutter diese am Wochenende Korrekturlesen kann. Dafür habe ich noch ein paar Kapitel geschrieben und auch andere Teile überarbeitet.	Schlussendlich bin ich der Meinung, es fehlen noch ein paar wichtige Sachen, doch jetzt habe ich schon einmal den grössten Teil fertig.
24. November	3 h	Navigation	Ich habe nach einer Lösung gesucht, wie man die Koordinaten genauer machen könnte. In der Schule haben sie leider nirgendwo ein Koordinatenmessgerät. Also habe ich mir überlegt, dass man es einfach mit dem gleichen Standortmessgerät, das einen schliesslich zum richtigen Ort führen soll, auch einmessen kann. So habe ich eine Anzeige auf dem Screen erstellt, dass man die aktuelle Position auf mehrere Nachkommastellen genau weiss. Schliesslich habe ich damit ein paar Tests in meiner Strasse gemacht um herauszufinden, wie genau diese Möglichkeit ist.	Diese Methode der Koordinatenmessung klappt relativ gut. Man findet die eingespeicherte Position auf etwa 5m genau wieder. Die Frage ist, ob das dann mit grösseren Distanzen auch den Fall ist. Schliesslich musste ich jedoch feststellen, dass jetzt zwar die Entfernung zum nächsten Punkt stimmt, doch die Richtung in die mein Pfeil zeigt überhaupt nicht mehr. Dieser zeigte nämlich immer nur nach Norden und nicht in die Richtung in die man gehen muss.
25. November	5 h	Verstecke, Fussabdruck	Ich bin nochmals in den Sihlwald gefahren und habe die Weisungen von Frau Aebli bezüglich den von mir	Ich hoffe, dass die Koordinaten jetzt genauer sind und man später dann die Positionen findet.

			vorgeschlagenen Verstecken durchgeführt. Deshalb habe ich einige neue Verstecke finden müssen. Auch habe ich bei den einzelnen Positionen die Koordinaten mit meinem programmierten Navigationsgerät gespeichert und Fotos von den Orten geschossen. Dann habe ich am Nachmittag den Fussabdruck aus Beton hergestellt.	
26. November	6 h	Schriftliche Arbeit	Am Wochenende hat meine Mutter meine erste halbfertige Version meiner schriftlichen Arbeit durchgelesen und korrigiert. Diese Korrekturvorschläge habe ich dann umgesetzt und auch noch weitergeschrieben.	Mir ist aufgefallen, dass gewisse Elemente in meiner Arbeit noch fehlen, obwohl ich schon so viel geschrieben habe. Ich glaube, ich werde zu viele Wörter am Schluss haben.
27. November	8 h	Programmieren und schriftliche Arbeit	Ich habe nach einer Lösung gesucht, wie mein Richtungspfeil in die richtige Richtung zeigen kann. Dabei hat mir Gian geholfen. Nachdem wir festgestellt haben, dass der Code aus meinem Swiftbuch einfach irgendetwas macht. Haben wir eine eigene Möglichkeit gesucht. Durch die Annahme, dass die Erde flach ist, funktioniert es nun. Danach habe ich an meiner schriftlichen Arbeit weitergeschrieben.	Natürlich kann man nur Koordinaten im Kanton Zürich nehmen. Denn die Approximation stimmt ja schliesslich nicht und wenn man zum Beispiel New York nehmen würde, würde der Pfeil nicht mehr in die richtige Richtung zeigen.
28. November	2 h	Schriftliche Arbeit	Ein weiteres Kapitel zu verschiedenen Outdooraktivitäten habe ich angefangen zu schreiben. Dabei habe ich mich auf	

			einige Bücher bezogen. Es fehlen jedoch noch ein paar Elemente in diesem Kapitel, wie zum Beispiel das Geocaching.	
29. November	4 h	Programmieren	Ich habe den Posten 5 eingefügt.	
1. Dezember	9 h	Programmieren	Heute habe ich weitere Elemente von verschiedenen Posten eingefügt und auch die jeweiligen Koordinaten hinzugefügt.	
2. Dezember	10 h	Programmieren	Wieder habe ich Teile von Posten eingefügt und überarbeitet.	
3. Dezember	9 h	Programmieren, Notizbuch	Ich habe eine erste, funktionierende Version der App fertiggestellt, so dass man sie ausprobieren kann. Dazu musste ich den Posten mit dem Druck einfügen und noch viel Text bei anderen Posten überarbeiten. Auch habe ich noch einige Grafiken eingefügt und weitere Elemente erstellt, die zum Lösen des Kriminalfalles gebraucht werden. Zum Beispiel habe ich in winziger Schrift den Zettel vom Posten Optik auf Papier geschrieben und noch „Fälschungen“ vom Kaufvertrag hergestellt.	
4. Dezember	4 h	Test der App	Heute war ich mit Noémie, Robin und Jonas im Sihlwald und wir haben gemeinsam meine App ausprobiert. Der Inhalt der Notizbuchfunktion war noch nicht eingefügt, da dort noch gewisse Teile	Hier konnte ich feststellen, welche Sachen gut waren und bei welchen Sachen noch etwas fehlte oder überarbeitet werden müssen. Am Schluss haben sie nicht den Mörder

			fehlten, doch ich habe die entsprechenden Sachen für sie ausgedruckt.	gefunden, doch es hat ihnen viel insgesamt Spass gemacht.
6. Dezember	2 h	Programmieren	Ich habe die Fehler in den Texten meiner App, die am Dienstag gefunden wurden, verbessert und an manchen Stellen den Text optimiert.	
7. Dezember	3 h	Schriftliche Arbeit	In meiner schriftlichen Arbeit habe ich etwas zum Polizeimuesum in Hamburg geschrieben und ein paar Quellen eingefügt. Auch habe ich ein paar Absätze überarbeitet.	
8. Dezember	9 h	Notizbuch, Schriftliche Arbeit	Ich habe die Seiten im Notizbuch angefangen einzuscannen und in die App und die schriftliche Arbeit einzubinden. Dann habe ich auch noch an meiner schriftlichen Arbeit gearbeitet.	
9. Dezember	8 h	Notizbuch	Ich habe nun alle Seiten des Notizbuches fertig geschrieben und auch die verschiedenen Versionen der Personalakten erstellt. Die Schriften habe ich alle eingescannt und die Scans dann zugeschnitten. Schliesslich habe ich alle in der richtigen Reihenfolge in meine App eingebunden.	Ich musste mich beim Einbinden sehr stark konzentrieren, da man genau wissen musste, welches Bild wann auf welcher Seite freigeschaltet wird oder überschrieben wird.
10. Dezember	7 h	Schriftliche Arbeit	Ich habe heute die Korrekturvorschläge von meiner Schwester miteingebunden und danach selber ein paar Abschnitte korrigiert und umgeschrieben. Auch habe	Das Aufschreiben der gesamten Interviews hat sich als sehr aufwändig herausgestellt. Da diese nicht einen Hauptteil in meiner Arbeit sind, habe

			ich mich um den Anhang und die darin enthaltenen Interviews gekümmert.	ich mich entschlossen, bei den noch nicht abgetippten Interviews nur die wichtigsten Stellen zusammenzufassen.
11. Dezember	9 h	App Icon, schriftliche Arbeit	Ich habe festgestellt, dass ich als App-Icon immer noch das provisorische hatte, welches ich am Anfang einmal festgelegt habe, damit ich überhaupt eines habe. Dieses war jedoch nicht ganz passend. Also habe ich ein neues designt. Danach habe ich noch etwas in meiner schriftlichen Arbeit geschrieben und korrigiert.	
12. Dezember	8 h	Schriftliche Arbeit	Ich habe die letzten Kapitel der schriftlichen Arbeit fertiggeschrieben und die Arbeit einmal komplett durchgelesen und korrigiert.	
13. Dezember	7 h	App, schriftliche Arbeit	Heute Abend habe ich die Texte innerhalb meiner App korrekturgelesen und danach noch Korrekturvorschläge von Robin in meiner schriftliche Arbeit integriert.	
14. Dezember	6 h	Schriftliche Arbeit	Ich habe Korrekturen in meine schriftliche Arbeit eingearbeitet.	
15. Dezember	13 h	Schriftliche Arbeit	Ich habe den Programmcode und die Screenshots in den Anhang gebunden und weiter meine Arbeit überarbeitet und verbessert.	

16. Dezember		Drucken	Ich habe meine Arbeit nochmals abschliessend durchgelesen und dann mehrere Exemplare ausgedruckt.	
17. Dezember		Binden	Die Arbeit wird beim Buchbinder abgegeben.	Ich hoffe, alles geht gut.
19. Dezember		Abgabe	Mein Vater holt die Exemplare von der Buchbinderei ab, da ich Schule habe. Schliesslich bringt er mir diese und ich werde sie dann im Prorektorat abgeben.	