



End-Dokumentation von H.A.M.  
Henrike Rähse, Amos Aulbach, Markus Pfliegensdörfer  
Semester 2 2019

# INHALT



Abstract  
S.4-5



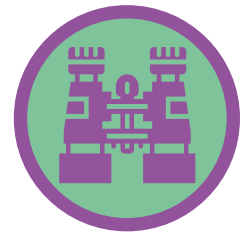
Idee  
S.6-7



Anwendung  
S. 8-11



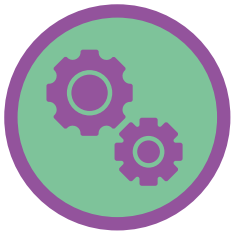
Styleguide  
S. 12- 17



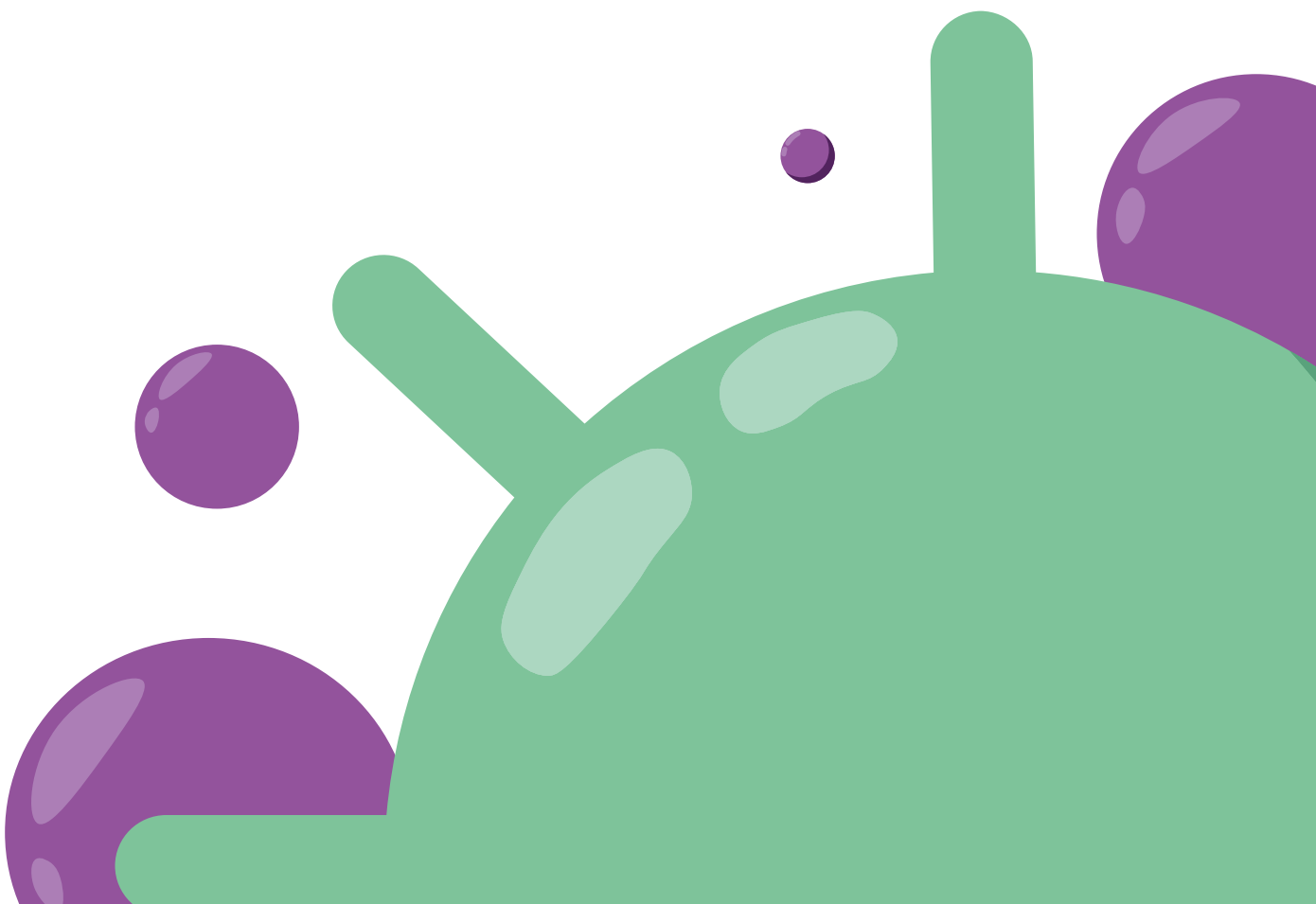
Ausblick  
S.36-37

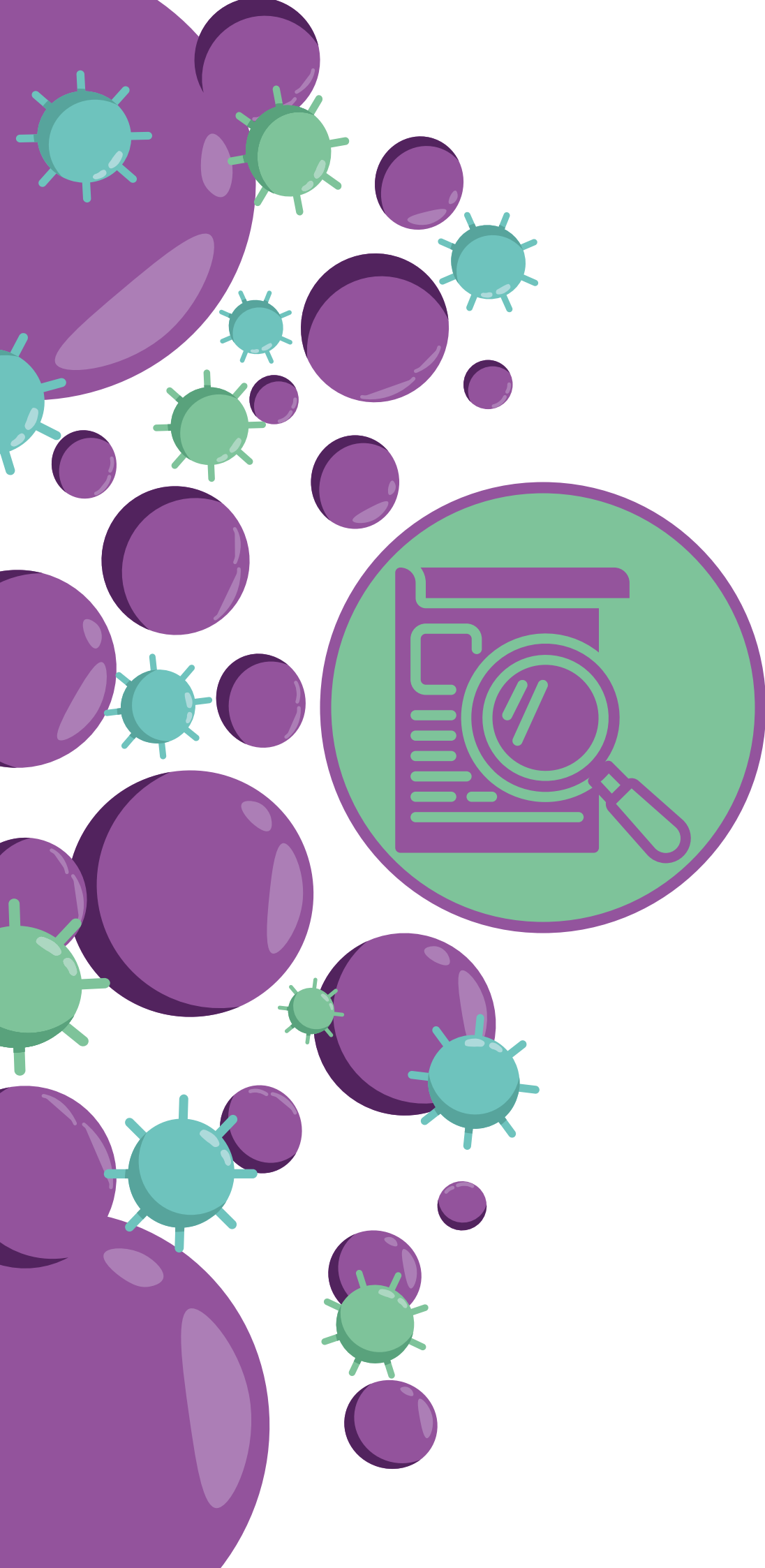


Management  
S.28-35



Technische  
Umsetzung  
S.18-27

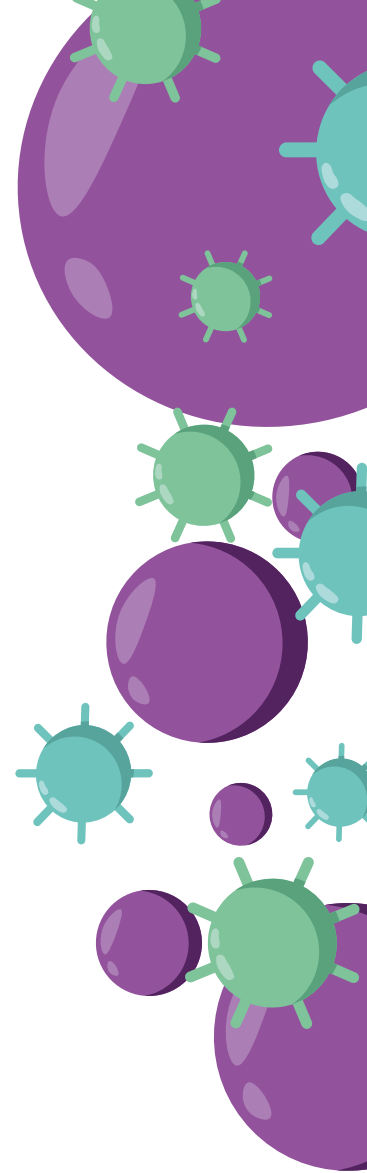


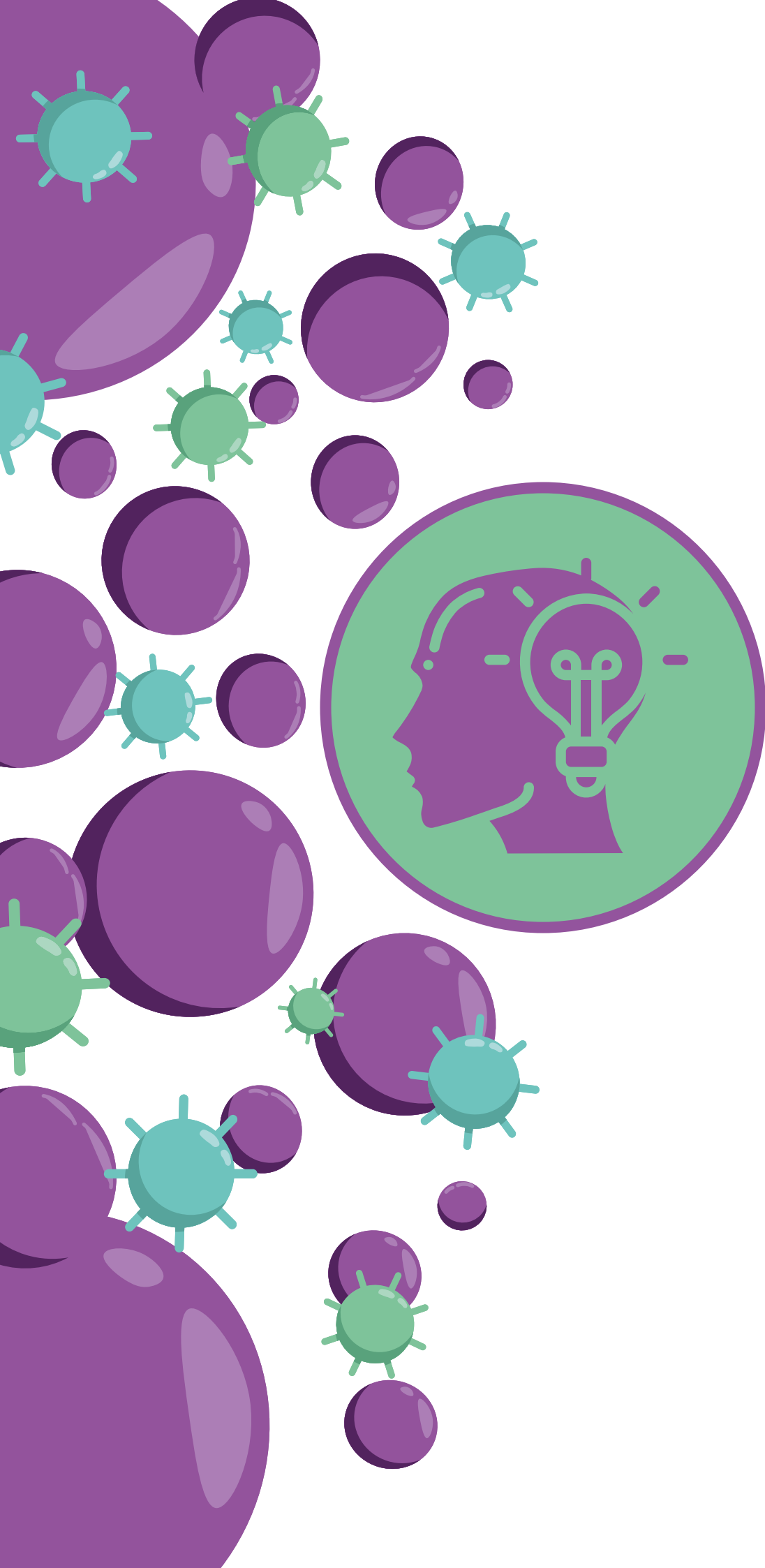


# Abstract

In today's society vaccines and vaccinations are very controversial topics. Anti-vax communities keep gaining a bigger audience and through that epidemics like the measles also gain more relevance. Epidemia is a simulation for pregnant women that are unsure whether vaccinating their kids is necessary or not. Our objective is to spread a wider conscience and understanding for the importance and necessity of vaccines especially the measles vaccine through an interactive simulation with visual feedback.

Henrike





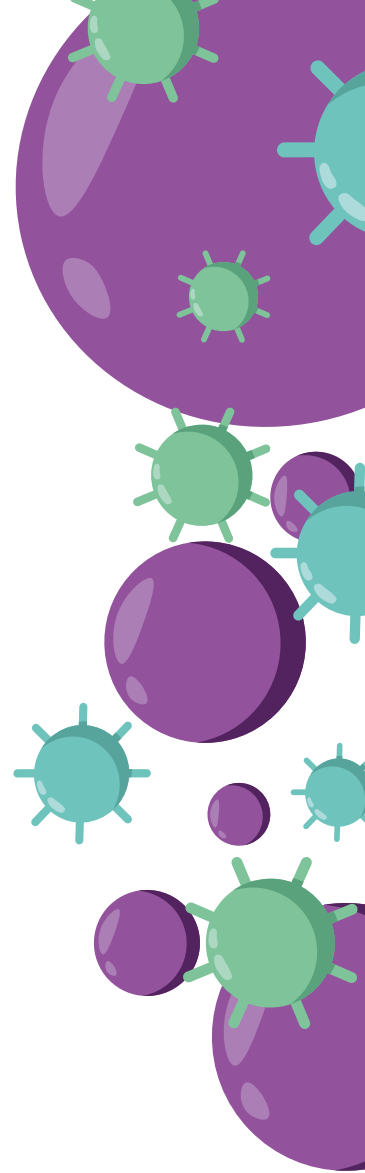
## Idee

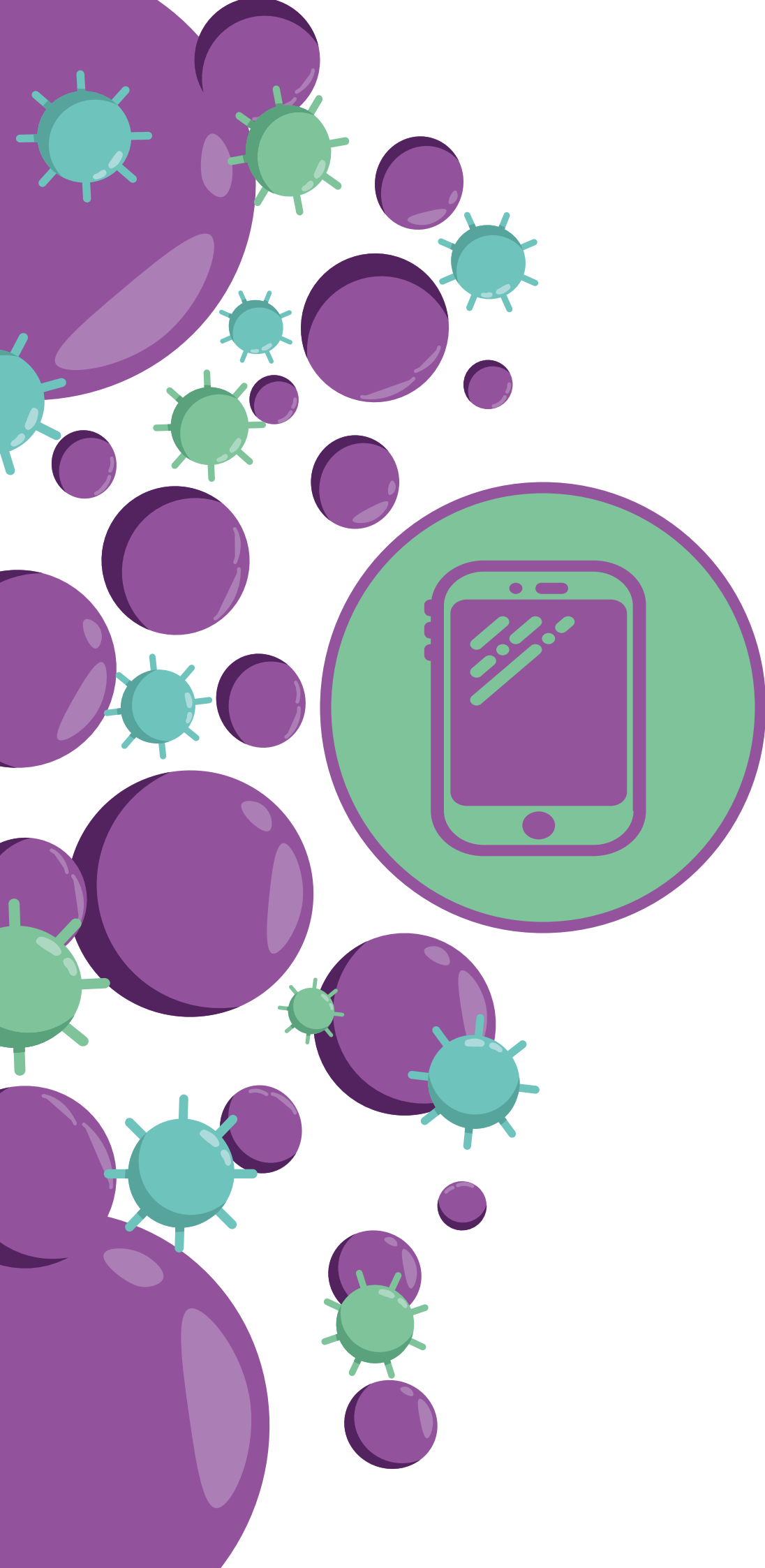
Im Gegensatz zu unserem vorherigen Konzept setzen wir auf ein Spiel, das den Nutzern die Bedrohung von Masern näher bringt, indem sie die visualisierten Viren von einer Gruppe Kinder fernhalten muss, während sie gleichzeitig versuchen müssen die Kinder zu impfen, um sie somit gegen die Bedrohung der Krankheit zu immunisieren.

Den Aufklärungspart übernimmt nun eine Website, auf welcher Informationen zum Thema Impfen und Masern bereitgestellt werden. Dadurch wollen wir unseren Nutzern eine weitere Informationsquelle bieten, der sie vertrauen können.

Die Nutzergruppe bleibt weiterhin schwangere Frauen, welche unsicher sind, ob sie ihr Kind impfen sollten.

Markus





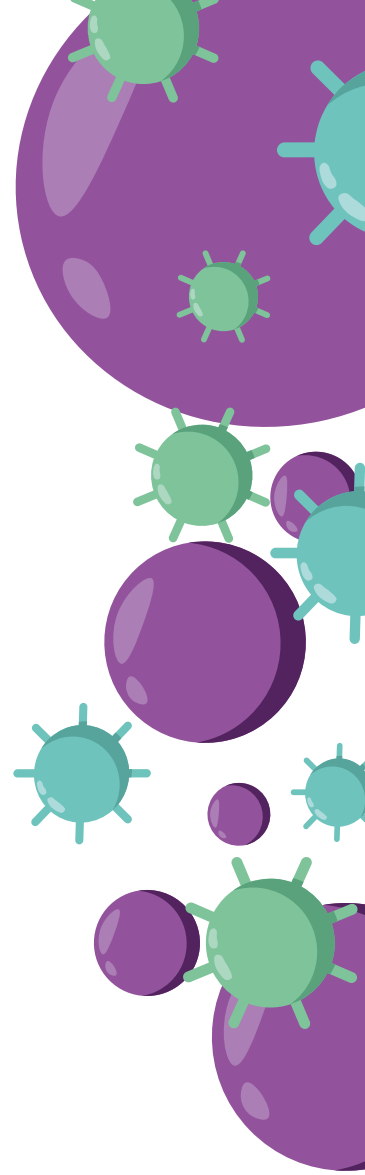


# Anwendung

## Nutzung und Kontext

Wir möchten unsere Simulation gerne in das Wartezimmer beim Frauenarzt bringen, sodass sich Schwangere oder andere Interessierte beim Warten auf ihren Arzttermin mit der Simulation auseinandersetzen können. Über eine Broschüre zum mitnehmen werden die Nutzer mithilfe eines QR-Codes zu unserer Simulation im App-Store geleitet. Die Broschüren sollen gut sichtbar im Wartezimmer platziert werden. Wenn der Nutzer unsere Simulation über den App-Store findet, so bekommt er am Ende der Simulation einen Hinweis auf unsere Website. Auf dieser kann er auch ohne Broschüre die wichtigen Infos zum Thema Impfen einsehen.

Markus



# Medienrecherche

## Warum die App?

Zu Beginn dachten wir eher an ein Touch User Interface im Wartezimmer des Arztes, jedoch hatten wir nicht beachtet, dass viele Leute von öffentlichen Bedienoberflächen angeekelt sein könnten. So kamen wir zu dem Entschluss die Simulation als App anzubieten und einen dazu führenden QR-Code auf den Flyer zu drucken. Damit ist es dem Nutzer möglich auf seinem eigenen Smartphone die Simulation zu spielen und somit kann er die Simulation auch außerhalb des Wartezimmers spielen und sie womöglich anderen zeigen.

## Warum die Website?

Die Website dient hauptsächlich dazu, potentiellen Nutzern die etwas von der Simulation gehört haben, eine Anlaufstelle zu bieten. Außerdem kann der Nutzer auf der Website die Informationen einsehen, die auf dem Flyer stehen, sollte er keinen besitzen.

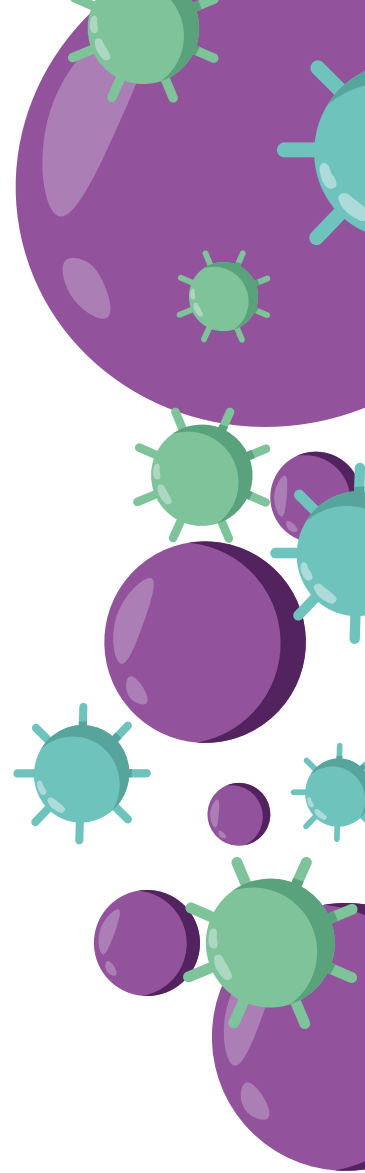
## Warum der Flyer?

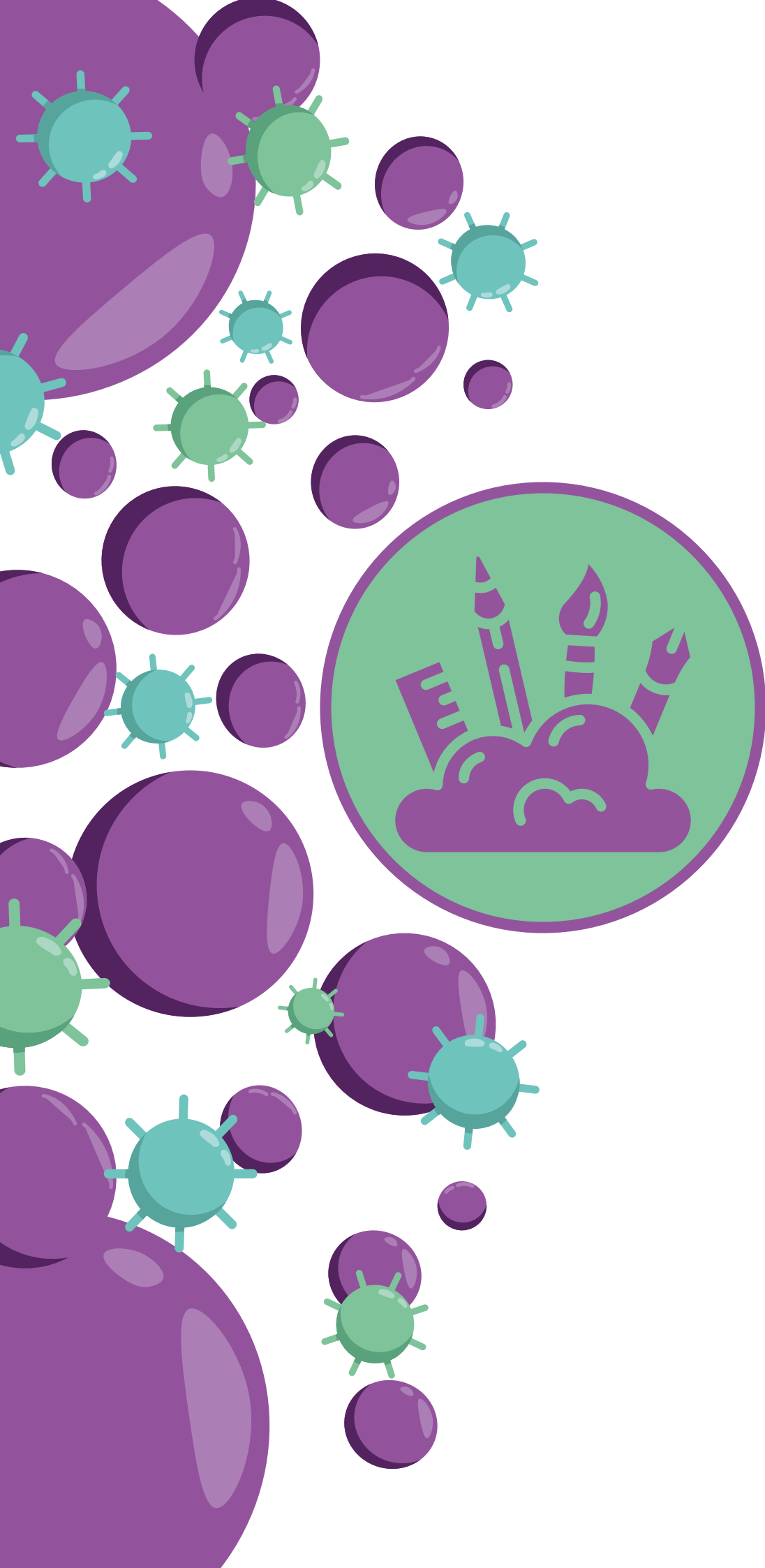
Über den QR-Code auf dem Flyer findet der Nutzer unsere Simulation im App-Store. Zudem kann der Nutzer auf einen Blick die wichtigsten Infos in Bezug auf Impfen einsehen. Abgesehen davon ist es kostengünstiger Flyer in Wartezimmern auszuhängen oder auslegen zu lassen, als in jedem Wartezimmer ein Touch User Interface zu installieren.

## Beschreibung der Elemente

Hygiene und Impfstatus des Kindes sind die wichtigsten Faktoren in unserer Simulation. Hygiene wird in Form von Schwämmen dargestellt. Die Schwämme können vor dem eigentlichen Spielbeginn vom Spieler frei platziert werden. Mithilfe der Schwämme kann der Spieler Viren in einem bestimmten Bereich verschwinden lassen umso die Kinder zu schützen. Am Anfang der Simulation sind alle Kinder ungeimpft und somit sehr anfällig für die Viren. Ziel der Simulation ist es so viele Kinder wie möglich zu impfen. Die geimpften Kinder sind gegen die Viren resistent. Um ein Kind zu impfen muss der Spieler eine auf dem Spielfeld versteckte Impfung in Form einer Spritze finden und diese zu einem Kind tragen. Der Impfstatus der Kinder lässt sich anhand des Aussehens des Kindes ermitteln.

Markus

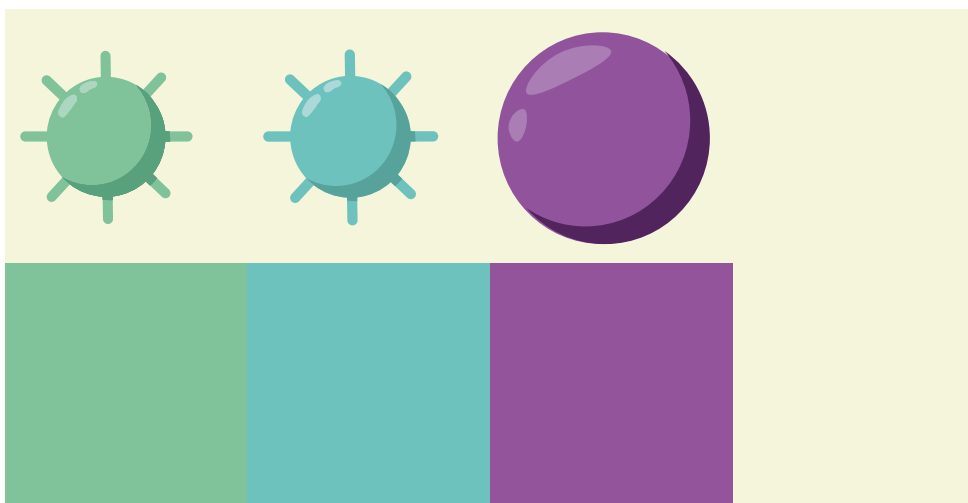




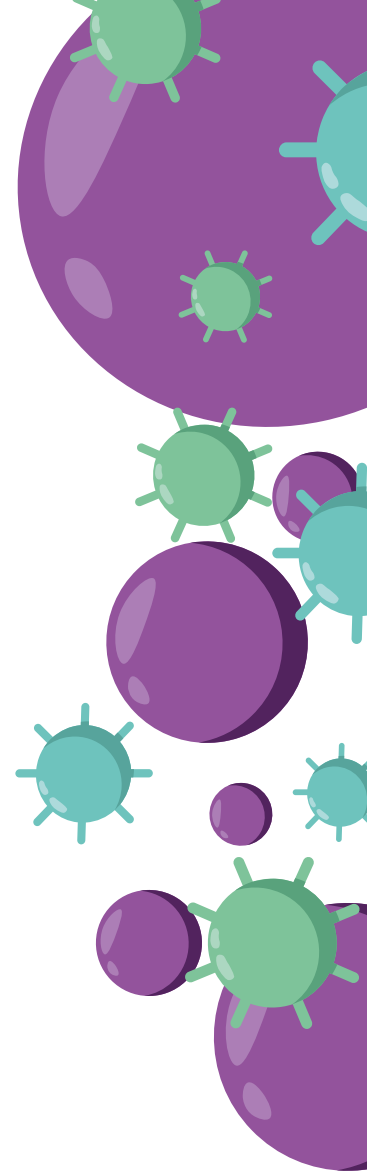
# Styleguide

## Farbpalette

Auch die grundlegende Farbpalette ist gleich geblieben. Um eine unnatürliche und bedrohliche Atmosphäre zu schaffen, haben wir die Farben grün, türkis und violett gewählt, da diese vermehrt mit Krankheit und Unnatürlichen Dingen in Verbindung gebracht werden. Demgegenüber stehen die Farben der Kinder und der Protagonistin welche in warmen und gedeckten Farben auftreten. Dies sorgt für einen hohen Kontrast zwischen dem 'Guten' und den 'bösen' Viren. Auf der Website wurde daher ein beiger Hintergrund gewählt um eine Kontinuität zwischen der Website und den positiven Farben der App zu gewährleisten.

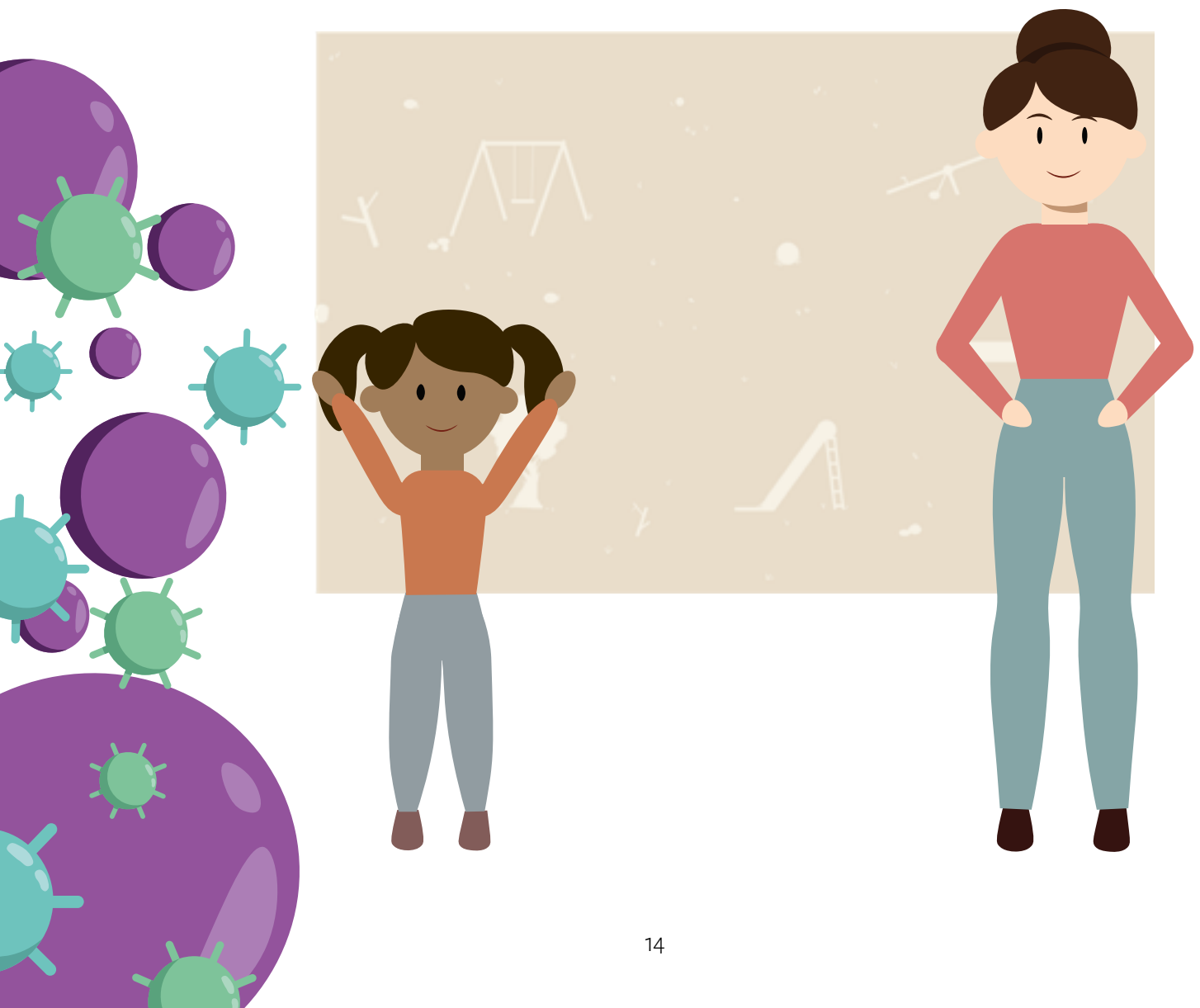


#7FC399 #6EC3BD #93549C #F5F5DC



## Stil

Da unsere App auf einem Handy gespielt wird, haben wir einen sehr stilisierten Stil mit wenig Details gewählt. Dadurch wollen wir uns von dem sterilen, leblosen Wesen das oft Ärzten zugesprochen wird abheben. Unser Ziel ist dadurch freundlicher, offener und vertrauenswürdiger zu wirken. Durch Verzicht auf Details wollen wir zugleich jedoch einen seriösen, nicht verspielten Charakter behalten, damit unsere Nutzergruppe nicht den falschen Eindruck von unserer Simulation erhält.



# Typographie

Unsere Schriftarten sind dynamisch und serifenlos, was zu einer guten Lesbarkeit beiträgt. Außerdem war uns wichtig das die Schrift vertrauenserweckend und nicht zu verspielt wirkt. Jedoch soll sie nicht wie eine "Zeitschrift Schrift" aussehen.

Deshalb haben wir uns bei der großen Überschrift für Exo in Semibold entschieden.

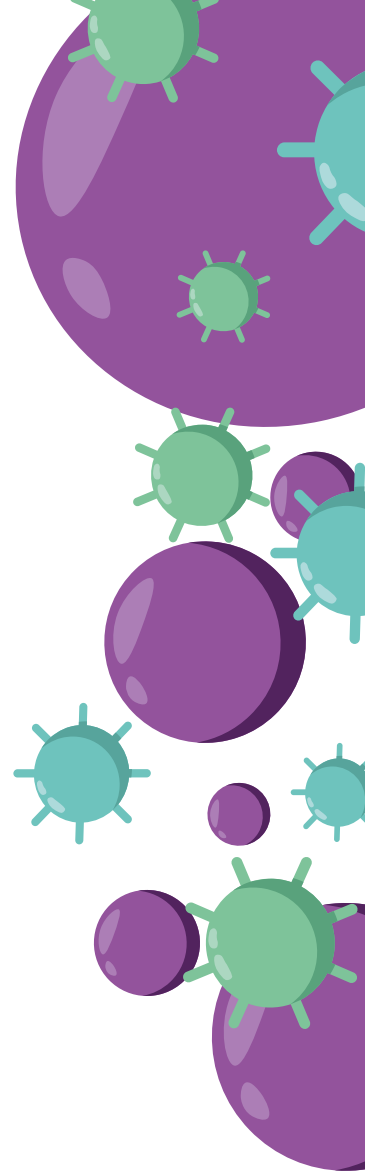
**EXO**  
**abcdefghijklmnopqrstu**  
**vwxyz0123456789**  
**Das ist die Exo in Semibold.!?**

Bei Zwischenüberschriften verwenden wir Exo in Medium

**EXO**  
**abcdefghijklmnopqrstu**  
**vwxyz0123456789**  
**Das ist die Exo in Medium.!?**

Für Fließtext verwenden wir Roboto Condensed in Regular

Roboto Condensed  
abcdefghijklmnopqrstu  
vwxyz0123456789  
Das ist die Roboto Condensed in Regular.!?

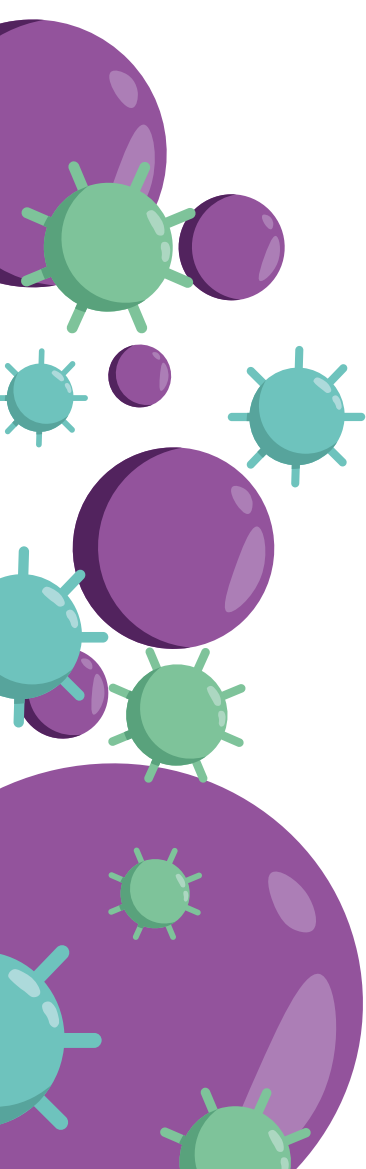


# Website

Das Design der Website ist an unsere allgemeinen Entscheidungen angepasst. So ist die Farbgebung des Hintergrunds die gleiche wie in der Simulation. Der violette Anteil überwiegt, da dies die Gefahr darstellen soll, die von den Viren ausgeht. Dem Nutzer soll bewusst sein, wie präsent das Problem bereits ist.

Um nicht zu sehr von dem Inhalt abzulenken, wurden nur ein paar Elemente der App in das Design der Website übernommen. Die Viren, die man auf dem Handybildschirm sonst nicht so genau erkennt, findet man auf der Website ebenfalls wieder. Auch der Stil der Charaktere findet sich indirekt in der Darstellung des Teams wieder, da diese in dem gleichen Stil erstellt wurden.

Die Icons wurden auf der Startseite als eine Art Eyecatcher genutzt, um den Nutzer nicht bereits auf der ersten Seite mit reinem Fließtext zu langweilen. Dazu dienen sie als eine Visualisierung um die Art unserer Simulation und ihren Nutzen auf einen Blick dazu stellen. Auf den Informationsseiten hingegen, wurde auf Icons verzichtet, da sie bereits einem Thema untergeordnet sind und Icons somit überflüssig wären.

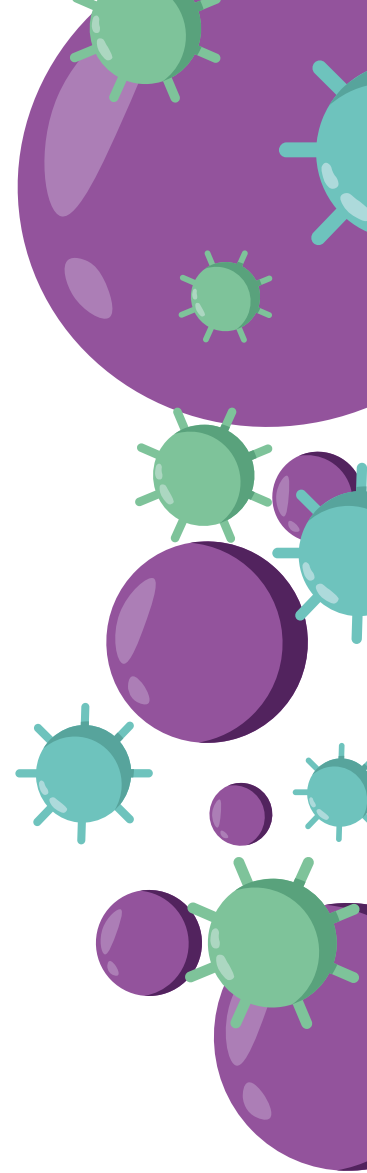


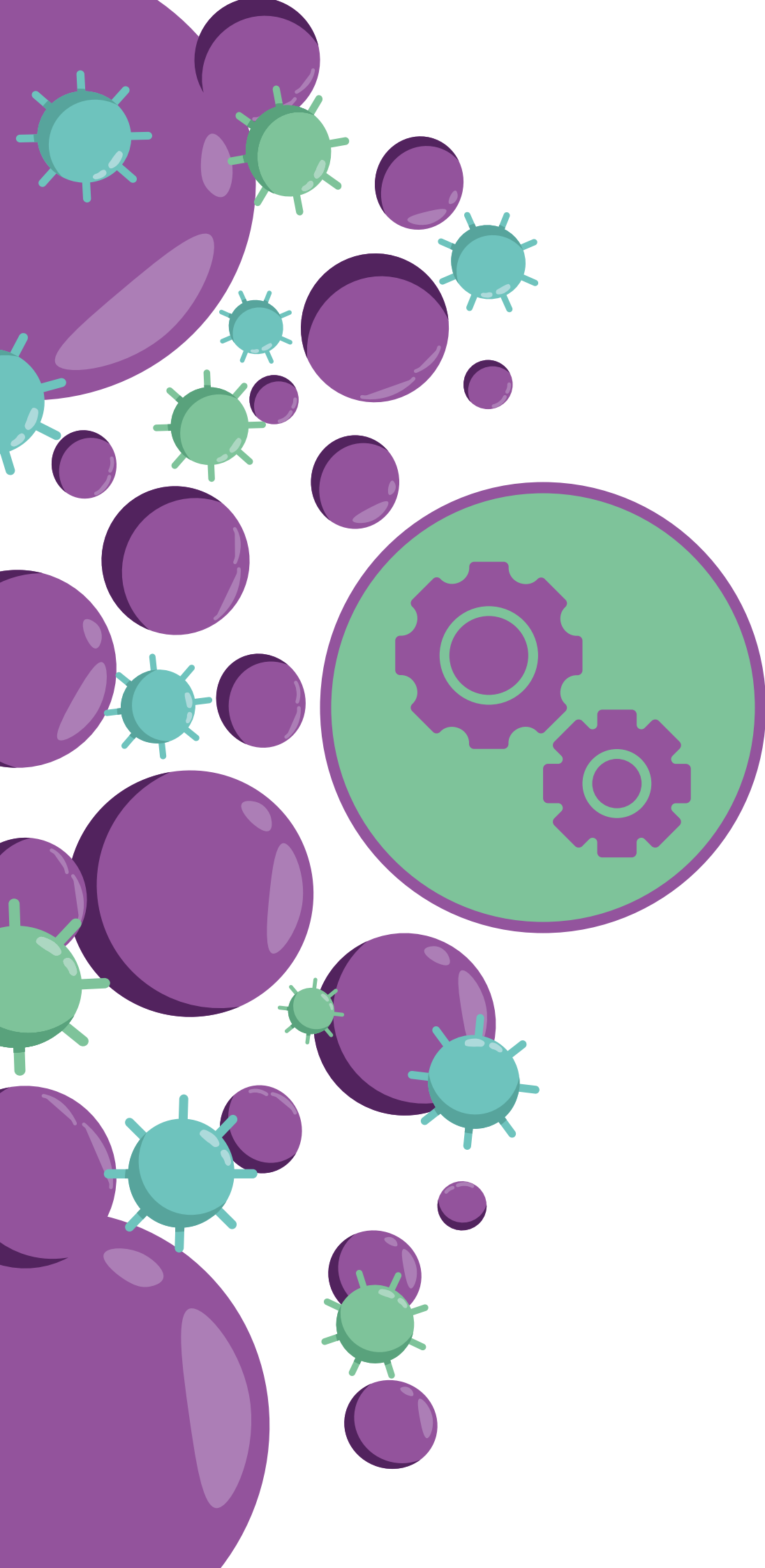


# Animation

Da unser Spiel von sich aus in Bewegung ist und jede zusätzliche Animation kontraproduktiv wäre, haben wir auf Animationen verzichtet. Auf der Website haben wir auf Micro Animationen gesetzt, um den Nutzer nicht von dem Inhalt abzulenken. Wie bereits erwähnt bewegen sich die Viren auf den Seiten minimal, sodass es merkbar ist, jedoch nicht ablenkend wirkt. Alle Buttons und Links vergrößern sich wenn der Nutzer über ihnen hovers, um zu vermitteln, dass sie klickbar sind. Sobald sie geklickt werden, verringert sich ihre Größe um den Eindruck zu vermitteln, dass sie in die Website wie ein Knopf reingedrückt wurden.

Henrike





# Technische Umsetzung

## App- Hauptdatei

Die Simulation ist in Javascript geschrieben und wird über HTML aufgerufen. Um hierbei den Überblick zu bewahren haben wir uns entschieden, objektorientiert zu arbeiten und die Klassen in externen Dateien, die in der Hauptdatei eingebunden werden, zu coden.

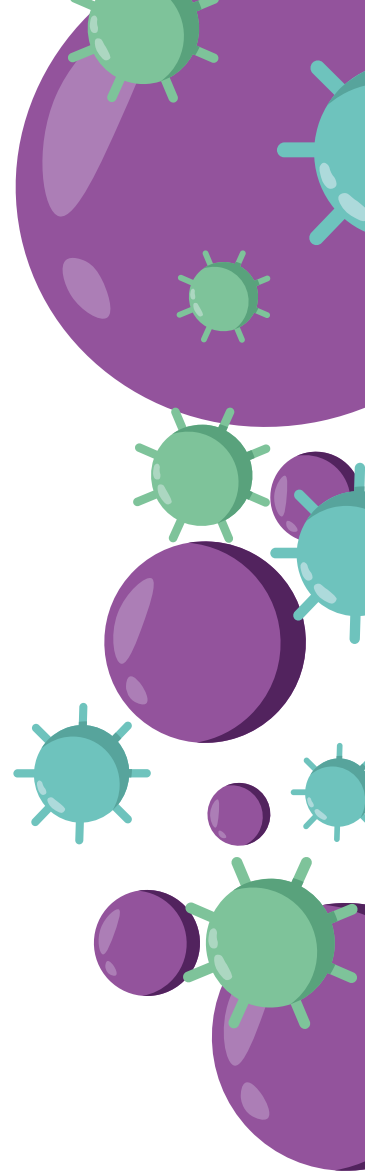
Die Bewegung der Kamera und der Protagonistin ist in der Hauptdatei geschrieben, da die Bewegungen der anderen Objekte auch davon abhängig sind und so am einfachsten darauf zugreifen können. Die obere Ecke des Hintergrundes ist als Variable festgelegt, damit sich der Hintergrund unter dem Sichtfeld bewegen kann.

Durch if-Abfragen wird ermittelt, welche Taste gedrückt ist und addiert oder subtrahiert so pro Durchlauf einen Wert zur aktuellen Position des Hintergrundes. Will man in der Simulation also beispielsweise nach rechts laufen, fährt der Hintergrund dementsprechend nach links und erzeugt so den Eindruck von Bewegung.

Erreicht die Position der Kamera allerdings eine gewisse Grenze, bewegt sich nicht mehr die Kamera, sondern die Protagonistin, die ansonsten in der Mitte des Sichtfeldes wäre. Dadurch kann die Spielfigur den Rand des Sichtfeldes erreichen.

Sobald sich der Charakter wieder in der Mitte befindet, bewegt sich die Kamera wieder. Dies wird umgesetzt, indem durch if-Abfragen die Position des Hintergrundes und des Spielers ermittelt wird. Die Variablen für die Hintergrundposition werden mit Variablen in einer externen Datei gleichgesetzt.

Auf diese Datei und die darin vorhandenen Variablen können auch die Dateien für die Klassen zugreifen, damit diese mit dem aktuellen Wert für den Hintergrund rechnen können.



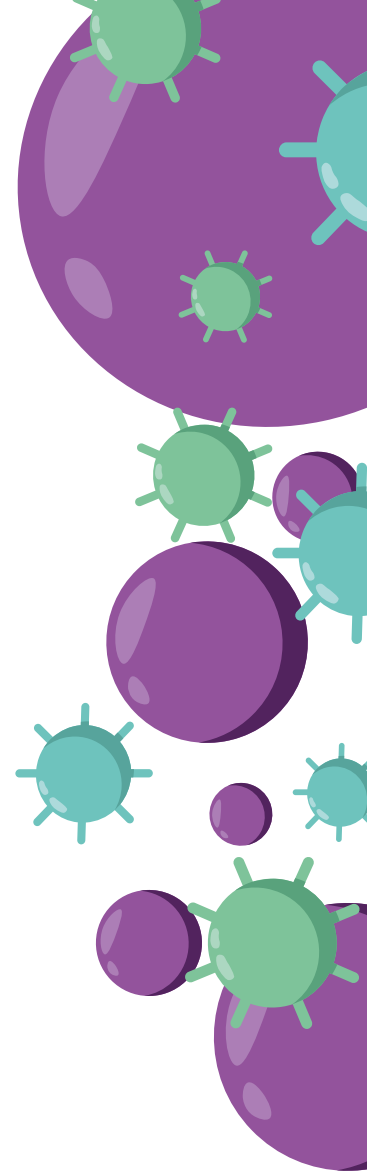
Zudem gibt es eine Variable screens, die bestimmt, in welcher Phase sich das Spiel befindet. Es gibt die Vorbereitungsphase, die Spielphase und das Ergebnis des Spiels. Wenn die Position der Maus innerhalb des Start Buttons ist und die Maus geklickt ist wird der Screen von 0 auf 1 gesetzt.

## Klasse Eltern

Um den Charakter glaubhaft im Raum darstellen zu können muss es möglich sein, verschiedene Blickrichtungen darstellen zu können. Dafür wird ermittelt, welche Tasten gedrückt und welche nicht gedrückt werden, um die Variable für die Richtung auf einen Wert von 0 bis 8 zu setzen. Zudem gibt es einen Zähler, der bis zu einem gewissen Wert hochzählt und sich anschließend wiederholt. Je nachdem, welcher Wert durch diesen Zähler dargestellt wird, wird ein anderer Frame der Laufanimation dargestellt. Durch eine If-abfrage wird geprüft, welche Richtung und welcher Zeitpunkt der Laufanimation gerade zutrifft, wodurch die richtige Grafik angezeigt wird. Dieses Bild wird über die Methode display, die in der Hauptdatei aufgerufen wird, an der Position des Spielers angezeigt. Außerdem muss in der Klasse der Protagonistin eine Variable, die prüft, ob der Hauptcharakter eine Impfung mit sich trägt. Die Variable für die Impfung kann entweder true oder false sein. Ist die Variable true, bedeutet das, dass der Spieler eine Impfung mit sich trägt.

# Klasse Virus

Der Hauptaspekt der Klasse für die Viren ist das Bewegungsmuster. Hierzu benötigt man vor allem die momentane Position auf der X- und Y-Achse, die Startposition auf der X- und Y-Achse sowie die Distanz, die pro Durchlauf auf der X- und Y-Achse zurückgelegt wird, als Variablen. Da sich die Viren vom Bildschirmrand zur Mitte des Spielfeldes bewegen sollen, muss hier mit dem Satz des Pythagoras gearbeitet werden. Dieser besagt, dass die Ankathete hoch zwei plus die Gegenkathete hoch zwei gleich der Hypotenuse hoch zwei ist. Dieser Satz gilt für rechtwinklige Dreiecke. In diesem Fall ist die Distanz auf der X-Achse die Ankathete, die Distanz auf der Y-Achse die Gegenkathete und die gesamte Distanz die Hypotenuse. Da die Ankathete und Gegenkathete ermittelt werden können, muss man diese nur in die Formel einfügen und die Wurzel des Ergebnisses ausrechnen, um die Gesamtstrecke zu ermitteln. Um zu ermitteln, welchen Anteil dieser Strecke pro Durchlauf des Computers zurückgelegt werden soll, benötigt man einen Teiler, durch den die Gesamtstrecke geteilt werden soll. Dieser ist die Gesamtstrecke geteilt durch die gewünschte Distanz pro Durchlauf. Anschließend teilt man die Distanz zwischen Start- und Mittelpunkt auf der X- beziehungsweise Y-Achse durch dieses Ergebnis und bekommt so die Distanz auf den jeweiligen Achsen, die zurückgelegt werden soll. Diese Ergebnisse kann man nachträglich noch durch beispielsweise Multiplikation oder Division verändern, um die Geschwindigkeit anzupassen. Die beiden Werte müssen aber gleichmäßig verändert werden, also die Proportionen zueinander beibehalten. Dieser Wert wird jede Runde zur aktuellen Position addiert, wodurch eine Bewegung der Viren entsteht. Zur Erleichterung bei der Fehlersuche wurden die Zwischenergebnisse als Variablen umgesetzt. Diese Bewegung wird durch die Methode `move`, die in der Hauptdatei aufgerufen wird, ausgeführt. Damit sich die Viren verhältnismäßig zum Hintergrund bewegen gibt es jeweils eine Methode pro Richtung, also oben, unten, links und rechts. Bewegt sich der Hintergrund in eine bestimmte Richtung, wird die jeweilige Methode ausgeführt, die die Startposition und die momentane Position des Virus parallel zum Hintergrund bewegt. Dies ist die einzige Methode, die unabhängig von der Variable `screens` ausgeführt wird. Alle anderen Funktionen finden nur statt, wenn die Variable `screens` den Wert 1 hat.



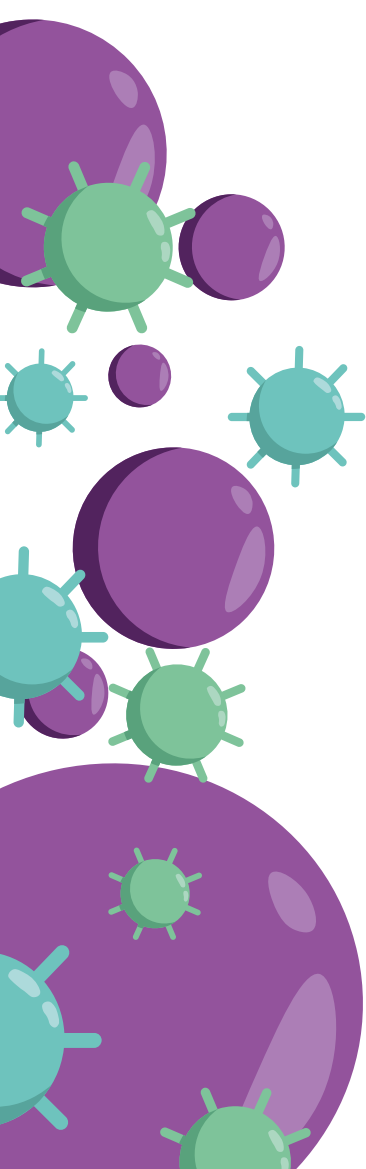
Die Funktion `respawn` besagt, dass die Position des Virus auf die Startposition gesetzt wird. Der Virus wird also an die ursprüngliche Position am Bildschirmrand zurückgesetzt.

Über die Methode `display` wird die Grafik für den Virus auf den jeweiligen Koordinaten angezeigt.

Da eine große Anzahl an Viren vorhanden ist, werden die Viren in einem Array erschaffen. Dadurch muss man die Methoden nicht für jeden Virus einzeln ausführen, sondern kann mit einer `for`-Schleife alle Viren auf einmal durchgehen und so die Methoden `move`, `display`, `respawn` sowie die Methoden zur Anpassung an den Bildschirmrand für alle Viren einfach und übersichtlich ausführen.

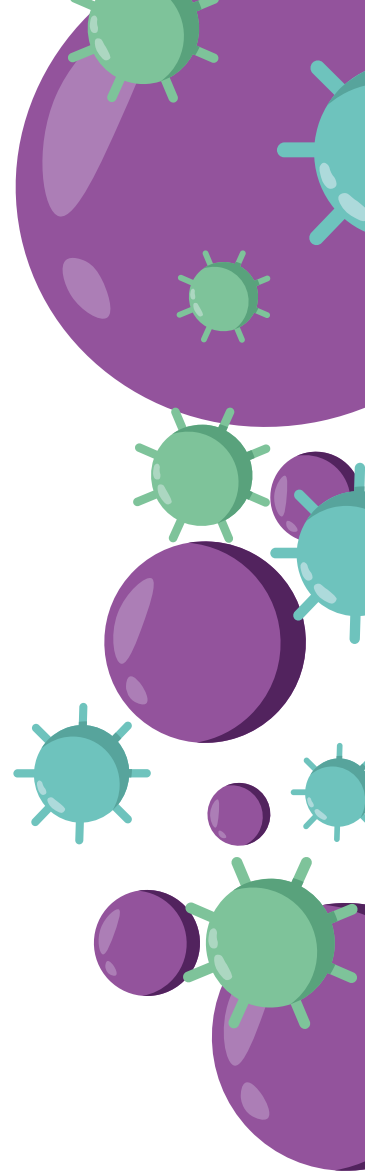
Innerhalb einer `for`-Schleife für die Viren befindet sich eine `for`-Schleife, die für jeden Virus jeden Schwamm durchläuft und durch eine `if`-Abfrage auf eine Kollision überprüft. Da es sich um einen Radius handelt, wird auch hier der Satz des Pythagoras angewendet, um den Abstand zwischen Virus und Schwamm zu ermitteln. Ist diese Distanz unter dem Radius des Schwamms, wird eine Kollision erkannt. Ist dies der Fall und ist der Schwamm aktiviert, wird die Methode `respawn` für den jeweiligen Virus ausgeführt.

Wenn die Protagonistin mit einem Virus kollidiert, soll sich die Geschwindigkeit des Spielers halbieren. Durch `if`-Abfragen werden die Positionen des Virus und des Protagonisten verglichen. Befindet sich die Position des Virus also innerhalb des Bereiches des Spielers, wird die Kollision erkannt und die Variable `stun` auf 0.5 gesetzt. Dabei beginnt ein Timer hoch zu zählen, der, sobald er einen gewissen Wert erreicht wird, wieder auf 0 gesetzt wird und `stun` wieder auf 1 setzt. In jeder Klasse, in der eine Position abhängig von der Hintergrundposition eine Rolle spielt, existiert ebenfalls eine Variable `stun`, die mit der Variable `stun` in der Hauptdatei gleichgesetzt wird. Die Bewegung des Hintergrunds, das Mitbewegen der Objekte sowie die Bewegung des Spielers werden halbiert, indem die jeweilige Distanz pro Durchlauf mit der Variable `stun` als Faktor multipliziert wird. Ist `stun` also auf 0,5, bewegt sich der Spieler halb so schnell.



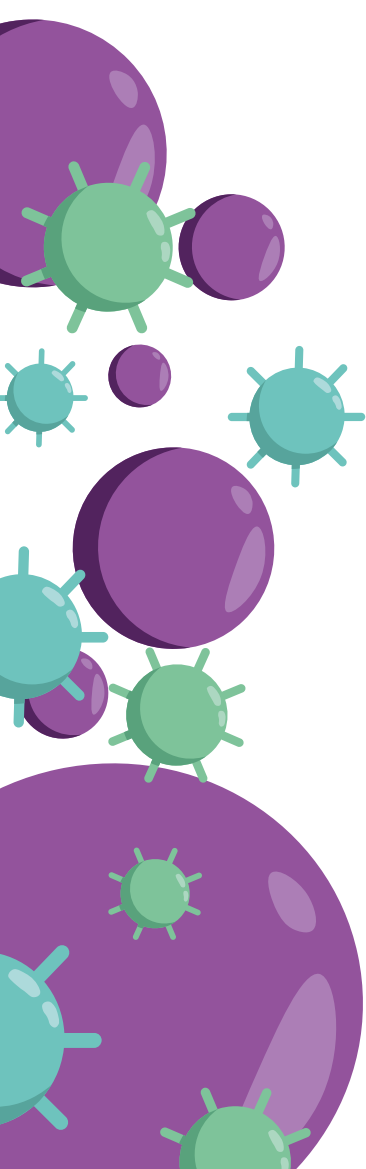
## Klasse Kinder

Auch die Kinder wurden als Klasse geschrieben. Hierfür werden Variablen geschaffen, die festlegen, auf welchen X- und Y-Koordinaten sich das Kind befindet, ob das Kind krank ist, ob es geimpft ist, und ein Timer, der festlegt, wann das Kind erkrankt. Die Methode `infect` wird nur ausgeführt, wenn die Variable für die Impfung des jeweiligen Kindes auf `false` ist, das Kind also ungeimpft ist. Wird die Methode `Infect` ausgeführt, beginnt der Timer, sich pro Runde um einen bestimmten Wert zu erhöhen. Erreicht dieser den Höchstwert, wird die Variable für die Krankheit auf `true` gesetzt. Über die Hauptdatei wird abgefragt, ob das jeweilige Kind krank ist, also die Variable für die Krankheit auf `true` gesetzt ist. Trifft dies zu, wird für das nächste Kind, sofern dieses nicht geimpft ist, die Methode `Infect` ausgeführt, wodurch der bereits beschriebene Ablauf für das nächste Kind ausgeführt wird. Die Grafik wird über die Methode `display` an den jeweiligen Koordinaten des Kindes angezeigt. Ebenso wie bei den Viren gibt es vier Methoden, die die Koordinaten parallel zur Bewegung des Hintergrunds verändern.



## Klasse Impfung

Für die Impfungen gibt es eine Variable `impftimer`, die bestimmt, ob und welche Impfung angezeigt wird beziehungsweise ob diese Impfung aktiv ist. Wenn der Spieler also eine Impfung aufsammelt, erhöht sich der Wert dieser Variable und die Impfung wird nicht mehr auf dem Spielfeld angezeigt. Zudem wird die Variable für das Impfen in der Protagonistenklasse auf `true` gesetzt, der Spieler trägt also eine Impfung. Eine `If`-Abfrage prüft, ob diese Variable auf `true` gesetzt ist und zeigt gegebenenfalls eine Spritze in einem Feld am Bildschirmrand an. Impft der Protagonist ein Kind durch eine `If`-Abfrage bezüglich einer Kollision, erhöht sich der `Impftimer` und die nächste Impfung ist aktiv und kann angezeigt werden. Zudem ändert sich die Variable des Kindes für die Impfungen auf `true`. Über die Methode `display` kann die Impfung auf der jeweiligen X- und Y-Koordinate angezeigt werden. Dies findet nur statt, wenn die Impfung aktiv ist. Zudem gibt es wieder Methoden, die die Impfung mit dem Hintergrund mit bewegen, genau wie auch bei den Kindern und den Viren.

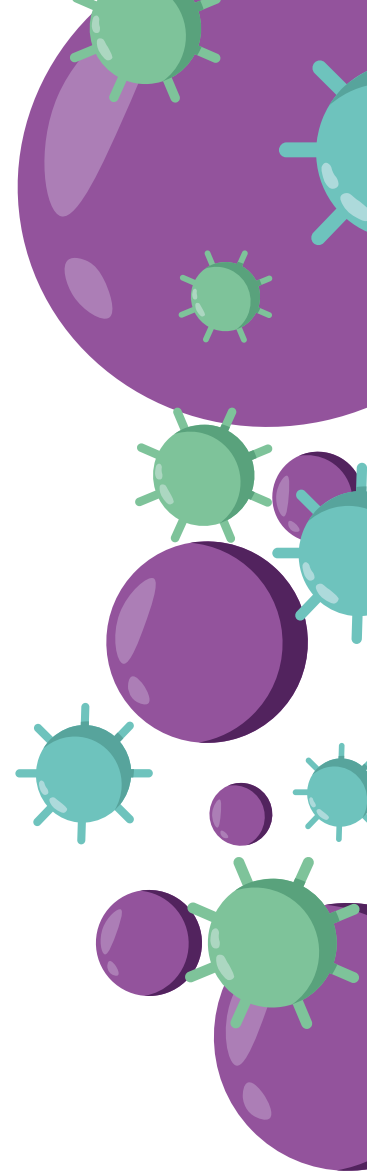




# Klasse Schwamm

Für einen Schwamm werden die X- und Y- Koordinate, den Index als z Wert, s für selected, also ausgewählt, a für active, also ob der Schwamm aktiviert ist, sowie den Wert prep, also ob man in der Vorbereitungsphase ist, als Variablen festgelegt

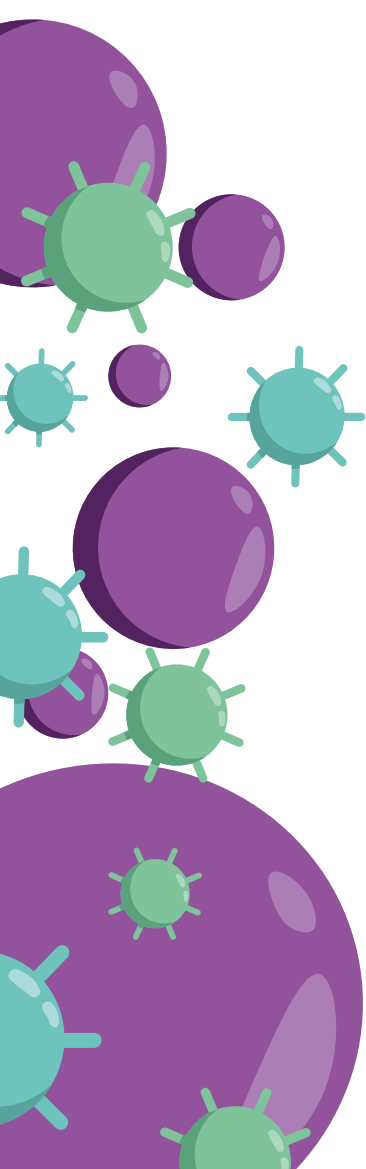
Schwämme können, solange die Variable screens gleich 0 ist, frei per drag and drop platziert werden. Zunächst wird über die Methode isSelected abgefragt, ob die Maus mit dem Schwamm kollidiert und ob diese gedrückt wurde. Ist dies der Fall, wird der Schwamm dem Array selectedschw auf der ersten Position hinzugefügt. Ist der Schwamm nicht mehr angeklickt, nimmt der z-Wert die Zahl der Stelle im Array an. Durch eine If-Abfrage wird daraufhin geprüft, ob dieser Array eine Länge über Null hat. Ist dies der Fall, wird eine zuvor erstellte Variable namens select mit dem ersten Element des Arrays, also dem angeklickten Schwamm, gleichgesetzt. Durch eine for Schleife wird nun für jeden Schwamm in diesem Array geprüft, ob der darauffolgende Schwamm einen größeren z-Wert hat. Ist dies der Fall, wird select mit dem neuen Schwamm gleichgesetzt. Daraufhin wird außerhalb der If-Abfrage der z-Wert von select auf 100 gesetzt, wodurch die Priorität des Schwamms erhöht wird. Würde man die Maus loslassen und einen neuen Schwamm anklicken, würde dieser neue Schwamm zu selected werden und den z-Wert 100 haben, während der Schwamm, der zuvor angeklickt war, den z-Wert der jeweiligen Arraystelle einnehmen würde und so die Priorität verlieren würde. Für den Schwamm, der gleich select ist, wird die Methode select ausgeführt, die, solange der Schwamm angeklickt wird, die Koordinaten des Schwamms mit der Maus gleichsetzt. Die Explosion des Schwammes, die die Viren respawnen lässt, wird durch Kollision des Schwamms mit dem Spieler verursacht. Durch eine If-Abfrage werden die Positionen miteinander verglichen und gegebenenfalls wird die Methode activate ausgeführt, die die Variable a, die bestimmt, ob der Schwamm aktiviert ist, auf true setzt. Ist keine Kollision vorhanden, wird die Methode deactivate ausgeführt und die Variable wird auf false gesetzt. Die Methode display zeigt den Schwamm auf dem Bildschirm an, wobei der Radius der Explosion nur angezeigt wird, wenn der Schwamm aktiviert ist, was durch eine If-Abfrage über die entsprechende Variable geprüft wird. zudem gibt es auch hier Methoden, die den Schwamm mit dem Hintergrund mitbewegen.

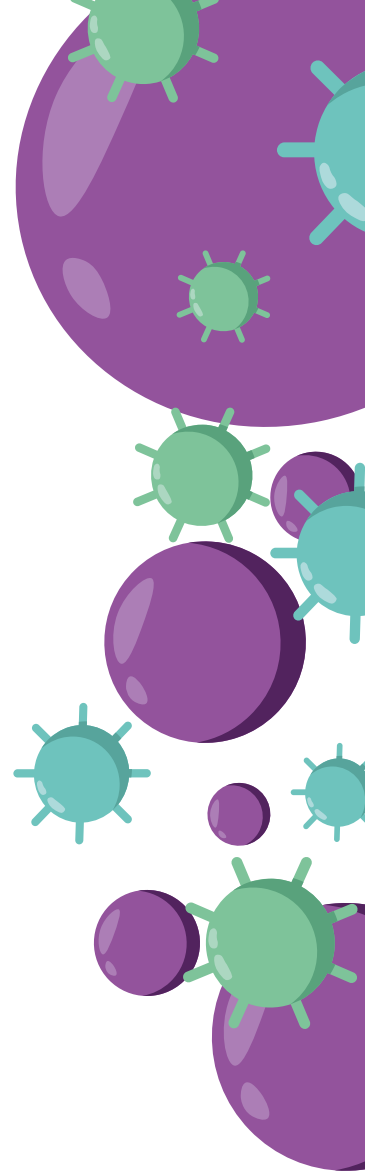


# Website

Die Website wurde mit CSS und HTML programmiert. Dabei wurde vor allem mit Flexboxen gearbeitet, um einen regelmäßigen Abstand zwischen den einzelnen Inhalten zu gewährleisten. Um verschiedene Inhalte übereinander zu lagern wurden meistens Margins oder Floats benutzt, wobei darauf geachtet wurde, dass die Abstände weiterhin eingehalten werden. Die Animation der Viren wurde durch zwei Bilder, einmal die grünen und einmal die blauen Viren, die übereinander liegen und sich unabhängig voneinander bewegen mit CSS programmiert. Da es stark ablenken würde, wenn sich die Viren über weite Strecken hinweg bewegen, haben wir uns dazu entschlossen lediglich eine minimale Bewegung einzubauen, welche jedoch durch die entgegen gesetzte Richtung der beiden Farben ausreicht. Die Links und der Appstore Button verändern auf hover und active ihre Größe um dem Nutzer zu verdeutlichen, dass sie sie anklicken können bzw. dass der Link oder Button gedrückt wurde. Responsive ist die Website zu diesem Zeitpunkt nicht, jedoch kann und sollte man dies noch hinzufügen, um die Inhalte auf allen Geräten zugänglich zu machen.

Henrike







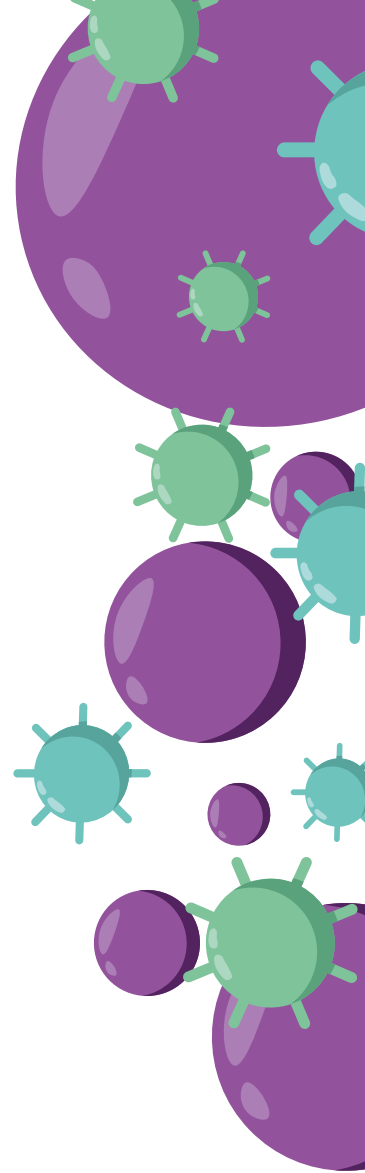
# Management

## Teamregeln

1. Wir gehen freundlich und respektvoll miteinander um.
2. Jedes Mitglied arbeitet in der dafür vorgesehenen Zeit aktiv mit.
3. Wir arbeiten konzentriert.
4. Erwartungen, Probleme und Kritik müssen offen und ehrlich mit den Team Mitgliedern besprochen werden.
5. Wir hören einander zu und lassen einander ausreden.
6. Deadlines müssen eingehalten werden.
7. Falls das nicht möglich ist muss das Team frühzeitig verständigt werden.
8. Die Arbeitszeit muss gut eingeteilt und jedes Teammitglied muss anwesend sein.

## Regelverstöße

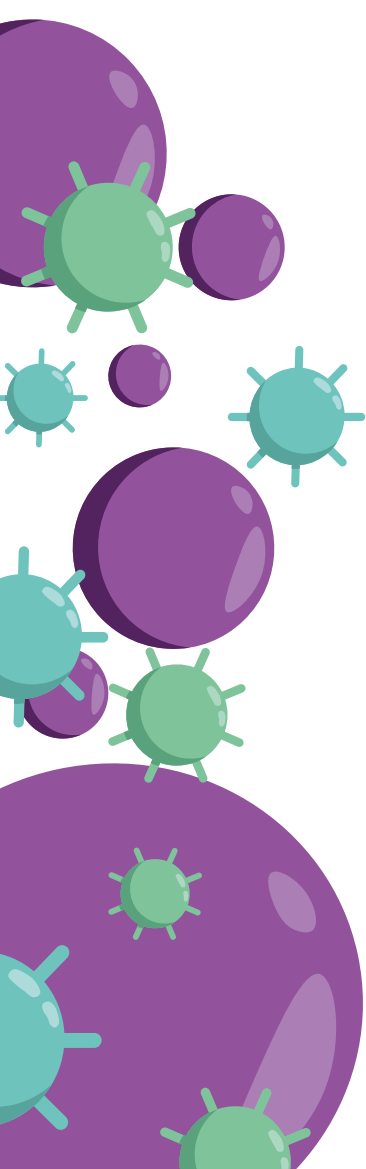
1. Darauf hinweisen (Besprechung)
2. Verschärfter Hinweis
3. Coaching mit Professoren
4. Bann



# Arbeitsklima

Allgemein wurde das Arbeitsklima als positiv eingeschätzt. Zwar gab es kleinere Meinungsverschiedenheiten und Diskussionen, doch sind diese in jedem Team abzusehen. Während der Konzeptphase im Besonderen hat die Teamarbeit sehr gut funktioniert. Das Team stand in regelmäßiger Kommunikation miteinander und hat sich ebenso regelmäßig getroffen. Entscheidungen konnten nach Überlegungen im Konsens gefunden werden, und der Zeitplan, wenn vorhanden, wurde eingehalten. Nach der Konzeptpräsentation mussten wir unseren Plan neu gestalten und auch dies hat reibungslos funktioniert.

Während der Umsetzungsphase nahm die Kommunikation jedoch nach und nach ab. Auch die Sprints die zu Anfang noch eingehalten wurden wurden nach Verteilung der Aufgaben nicht mehr eindeutig deklariert. Stattdessen hat sich jeder auf die Aufgaben konzentriert die er zu erledigen hatte, ohne das restliche Team über Fortschritte oder ähnliches zu informieren. Dies hat bis zum nächsten Treffen zur Besprechung des Trailers angehalten, wo das Problem angesprochen wurde. Daraufhin hat sich die Kommunikation deutlich verbessert bis zum Projektende.



# Probleme während des Projekts

## Probleme im Team

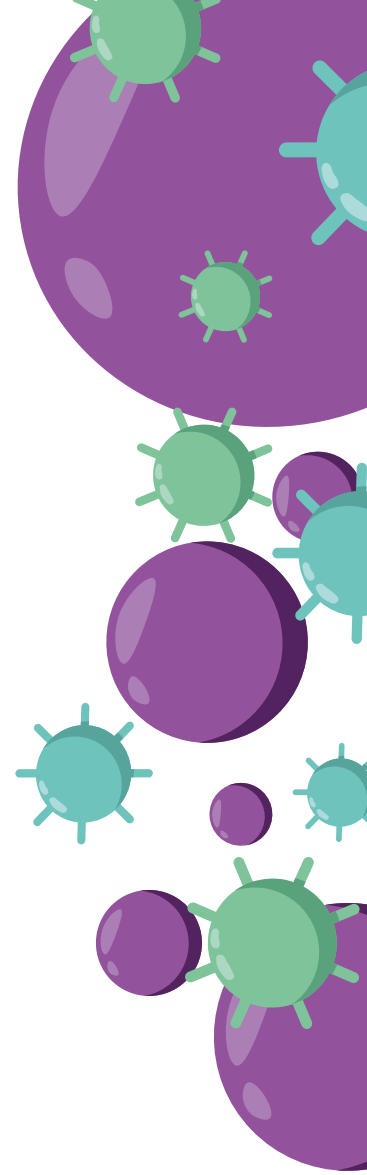
Zu den Hauptproblemen innerhalb des Teams zählte definitiv der fehlende oder mangelhaft ausgearbeitete Zeitplan, wodurch einzelne Teammitglieder keine Struktur in ihrer Arbeit hatten. Dadurch wurden ab Mitte Juni auch die Deadlines nicht mehr konsequent eingehalten, was zu vermehrtem Stress und Anspannung geführt hat. Jedoch haben wir alle Probleme lösen können und sind als Team an ihnen gewachsen.

## Probleme für das Team

Das größte Problem, dem wir uns als Team stellen mussten, war der Konzeptwechsel nach der Präsentation. Somit mussten wir schnell ein neues Konzept finden, wobei uns Anfangs nicht einmal bewusst war, was das Problem mit dem aktuellen Konzept war.

Nachdem wir uns dann auf etwas neues geeinigt hatten, haben wir sehr schnell mit der Umsetzung angefangen. Dadurch haben wir nicht von Anfang alles beachtet, jedoch hat sich im restlichen Verlauf alles wichtige zusammengefunden.

Während der Umsetzung traten vor allem zu Beginn Probleme mit dem Code auf, welche jedoch behoben werden konnten. Generell kamen aufgrund des fehlenden Zeitplans schubweise sehr stressige Phasen, abgewechselt von ruhigeren Phasen, die man wahrscheinlich hätte vermeiden können.



# Erkenntnisse für das Team

## Henrike

Während der Teamarbeit habe ich vor allem gelernt wie wichtig Kommunikation innerhalb eines Teams ist. Nicht nur damit jedes Teammitglied weiß auf welchem Stand das Projekt ist, sondern auch um Feedback zu seinen Aufgaben zu bekommen. Dadurch würde man auch eher bemerken, wenn man zu viele Aufgaben hat und könnte sich absprechen um eine Lösung zu finden. Zudem habe ich Erkenntnisse über mich selbst sammeln können, die mir in zukünftigen Gruppen weiterhelfen werden.

## Amos

Was mir in der Arbeit in der Gruppe besonders aufgefallen ist, ist dass ich besser an meiner Zeitplanung arbeiten muss. Zudem habe ich gemerkt, wie hilfreich eine strukturierte Arbeitsweise, wie zum Beispiel To-Do Listen, sein kann. Vor allem in der Konzeptphase habe ich gelernt, nicht mehr zwanghaft an Ideen fest zu halten und offener für neues zu sein. Diese Einstellung war auch beim Coden sehr hilfreich, da ich zuvor nie in diesem Umfang Objektorientiert programmiert habe und so vieles gelernt habe und viele Herausforderungen bewältigen konnte. Dadurch fällt es mir auch einfacher, die Hemmschwelle, die man hat, bevor man mit einer neuen Aufgabe beginnt, zu überwinden.

## Markus

Ich habe während der Teamarbeit gelernt mit Stress klar zu kommen, wie Zusammenarbeit im Team funktioniert und wie man seinen Platz in einem Team findet. Zudem habe ich mich selbst teilweise besser kennengelernt und das die erste Idee nicht immer die beste ist. In Zukunft möchte ich mit meinen Teamkameraden öfter und offener kommunizieren, um Missverständnisse gleich aus dem Weg zu räumen.



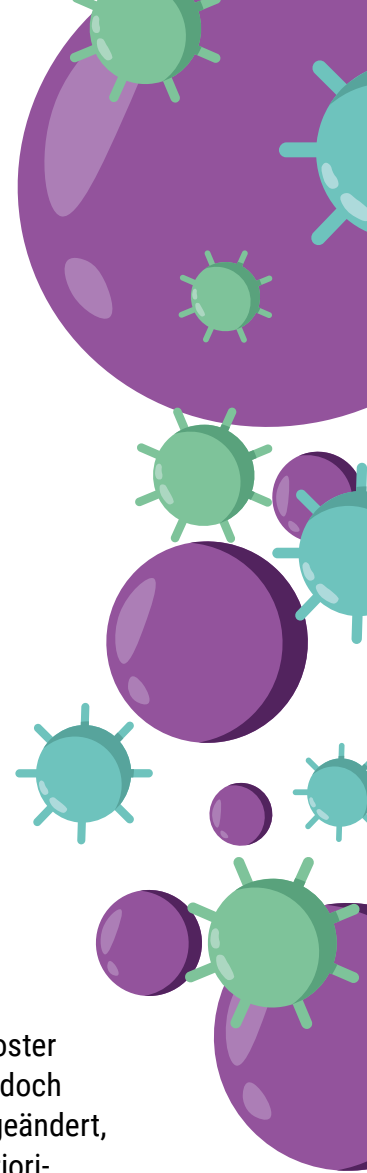
# Fortschrittsberichte

Die Arbeitsweise wurde nach dem Scrum Prinzip geplant. Dafür haben wir uns anfangs einmal die Woche getroffen und Sprints erstellt und besprochen.

Jedoch hat dies in der Hochphase der Umsetzung aufgrund von mangelnder Kommunikation nicht mehr so gut geklappt. Während dieser Wochen haben wir uns nicht bewusst getroffen, um die Sprints zu besprechen, sondern lediglich hin und wieder Updates zu den Einzelaufgaben mitgeteilt.

Der letzte Fortschrittsbericht, den wir geführt haben, ist vom 10.06 zu dem User Poster und der dazugehörigen Tabelle. Dort wurde der Sprint bis zum 19.06 festgelegt. Jedoch wurden diese Ziele nicht bis zu der Deadline vollendet. Die Prioritäten haben sich geändert, allerdings teilweise ohne das Team auf dem Laufenden zu halten. Die gesetzten Prioritäten wurden nicht in einem neuen Sprint festgehalten, sondern einfach durchgeführt.

Erst am 21.06 haben wir uns erneut zusammengesetzt um das Konzept für den Trailer zu entwerfen und das weitere Vorgehen zu besprechen.



# Businessplan

## Value Proposition

Für Schwangere Frauen, die sich unsicher sind, ob sich das Impfen ihres Kindes lohnt, ist Epidemia die Simulation, die spielerisch Inhalte vermittelt und als weitere Informationsquelle dient.

## Customer Relationships

Die Nutzer können uns über die Website, über Feedback an die Krankenkasse oder über den Appstore erreichen.

## Customer Segments

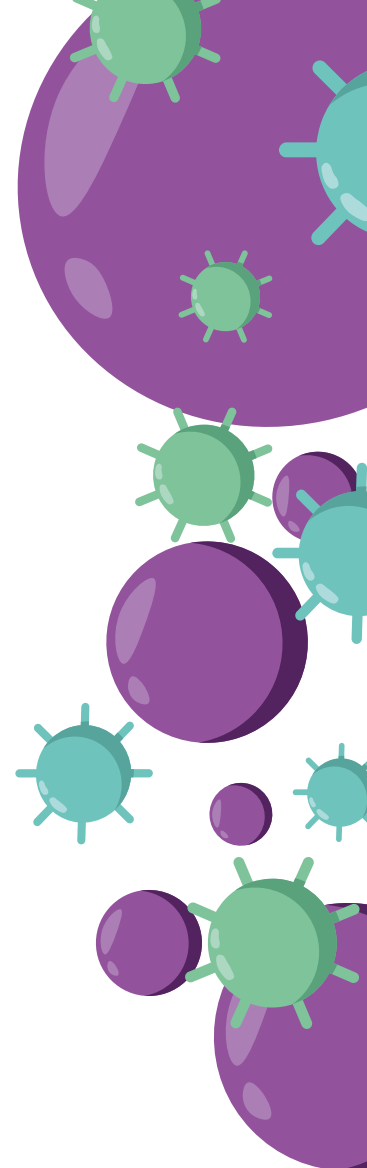
Unsere Kunden sind voraussichtlich Versicherungen und Krankenkassen, die unsere Simulation den Frauenarztpraxen und somit den schwangeren Frauen zu Verfügung stellen.

## Channels

Die Krankenkasse stellt den Pappaufsteller auf und platziert die Flyer. Über den Appstore können die Nutzer zu der Simulation gelangen sowie über die Webseite.

## Key Partners

Unsere Geschäftspartner wären zum einen Druckerei und Google/Apple Store zum Vertrieb, zum anderen Krankenkassen, Versicherungen, das Robert-Koch-Institut(RKI) und der Marburger Bund (Ärzteverband).



## Key Activities

Für die Realisierung des Projekts sind einige Schritte notwendig. Zunächst muss in der Konzeptphase geklärt werden, in welche Richtung sich das Projekt entwickeln soll. Anschließend werden sowohl die App als auch die Webseite designet. Parallel dazu werden sowohl Webseite als auch die Simulation programmiert. Daraufhin wird ein Trailer erstellt der zur Vermarktung des Produkts dienen soll.

## Key Resources

Als Ressourcen zur Realisierung des Projektes benötigen wir zum einen unsere Laptops und Software sowie Arbeitszeit, um diese zu nutzen. Außerdem benötigen wir neben dem Bildschirm für den Trailer auch Papier für die Flyer sowie die nötigen QR-Codes, die darauf gedruckt werden sollen.

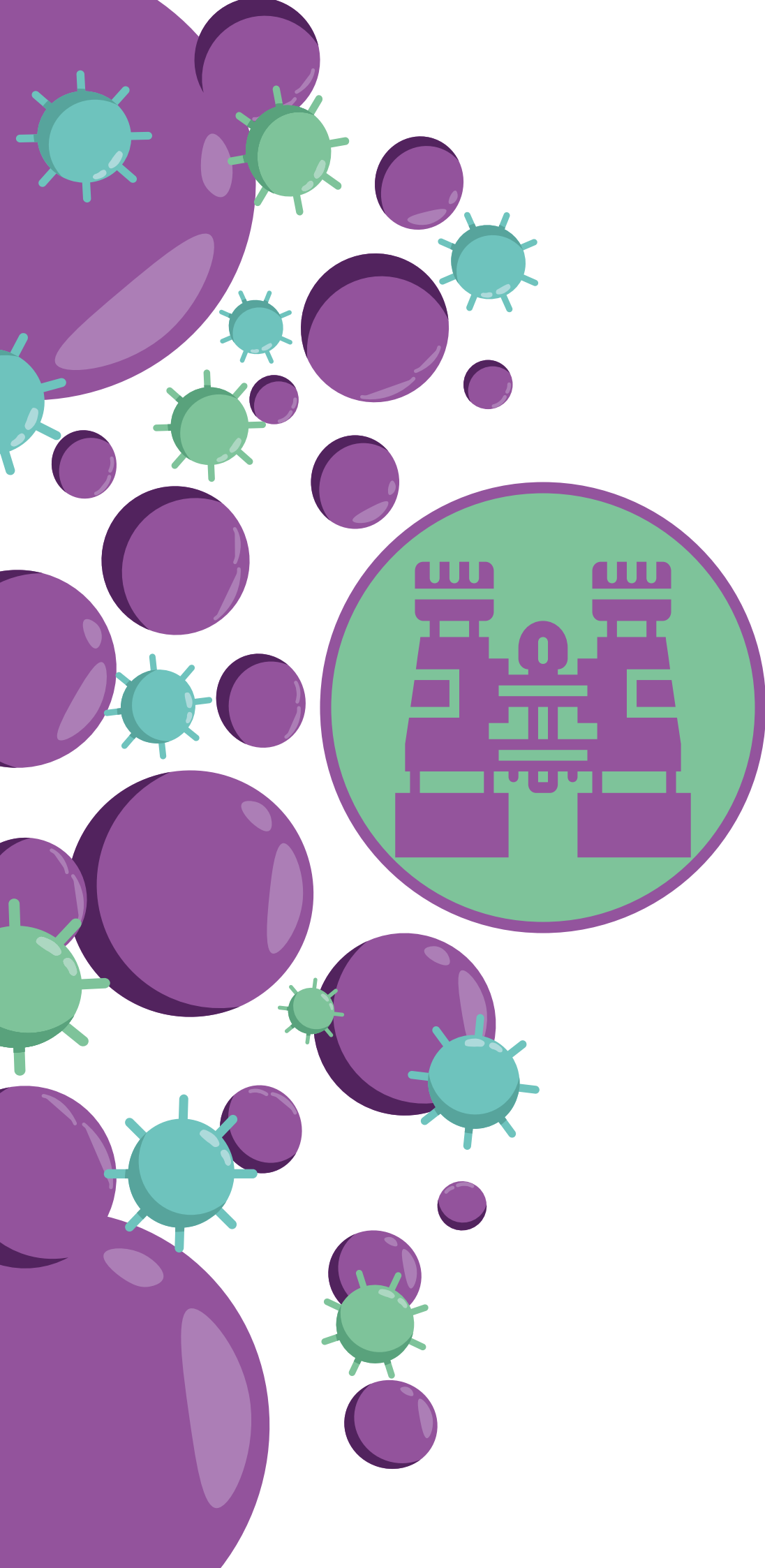
## Cost Structures

Um das Projekt umzusetzen, müssen die Kosten von Bildschirm, Druck, Personal, Server und die Gebühren für den Appstore gedeckt werden.

## Revenue Streams

Wir arbeiten als non-profit Organisation und unsere Projekt soll vom RKI, Versicherungen und den Krankenkassen finanziert werden.

HAM



# Ausblick

Bevor Epidemia tatsächlich vermarktet werden könnte, müssten noch weitere Schritte eingeleitet werden. Das wichtigste hierbei ist, dass das Spiel als App umfunktioniert und die Website responsive gemacht werden müsste.

Des weiteren fehlen in der App noch Funktionen, die eingefügt werden müssen. Dazu zählt eine Sprühflasche, die der Protagonist mit sich führen kann und zum Bekämpfen der Viren dient und eine Weiterleitung zur Website, falls man die App nicht über den Flyer gefunden hat.

Sound Design ist definitiv auch etwas was noch bedacht werden muss, allerdings haben wir keinen genauen Plan inwiefern wir Sound in der Simulation verwenden würden.

Dann muss sowohl die App als auch die Website öffentlich zugänglich werden und die Flyer und PR Gegenstände wie Pappaufsteller gedruckt werden. Um die Simulation zu verbreiten und zu finanzieren planen wir, Kontakt zu den Krankenkassen aufzunehmen und mit diesen zu kooperieren. Diese wiederum würden unsere Simulation an ihre Kunden und somit unsere Nutzer weiterleiten.

Amos, Henrike

