

Be[lat] the Virus



Einleitung

Teamvorstellung

Wir sind Team zwei aus dem Sommersemester 2020. Unser Team besteht aus Diandra Hermann, Florian Grünwald, Marie Sasse und Leander Schmidt. In dieser Dokumentation beschreiben wir unsere Simulation „Be[at] the Virus“.

Problem

Jedes Jahr erkranken und sterben viele Menschen an Infektionskrankheiten, wie Grippe, Malaria, Tuberkulose, Diphtherie, Ebola und vielen mehr. Einfache Maßnahmen, wie das richtige Händewaschen oder Tragen eines Mund-Nasen-Schutzes, können viel bewirken. Doch Wissenschaftler fanden heraus, dass die deutsche Bevölkerung eine unzureichende Hygienepraxis durchführt.

Unsere Simulation

Diesem Problem wollen wir uns annehmen und klären Kinder im Alter von acht bis dreizehn Jahren durch unsere Simulation über die Wichtigkeit und Durchführung einer wirkungsvollen Hygienepraxis auf.

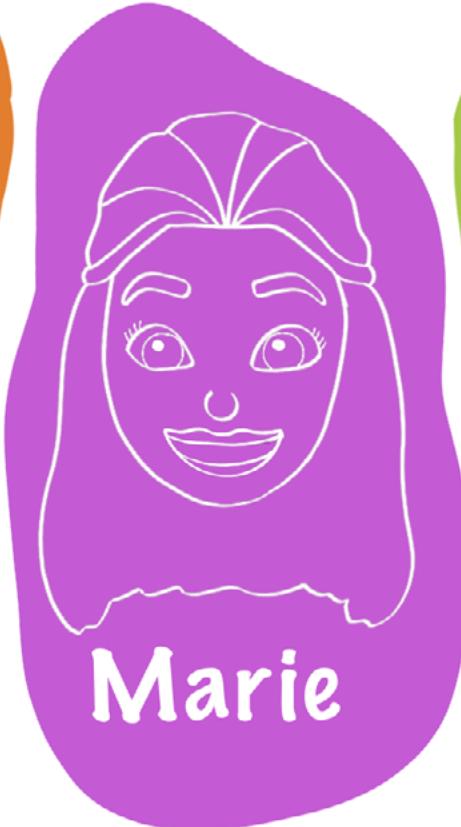




Leander



Diandra



Marie



Flo

Inhaltsverzeichnis

Konzeptänderungen

Händewaschen	7-8
People Bouncy	9-11

Technik

View Hierarchy	13-14
Action Hierarchy	15-16
Technische Umsetzung	17
Probleme	18
Globale Variablen	19
Probleme	20
Event Dispatcher	21
Parent Aufruf	22
Szenenverwaltung	23-24
Untersimulationen	25
Klassendiagramm	26

Design

Designprozess	28
Elemente	29-32
Animationsguide	33-41
Sound	42-43

Management

Organisation	45
Projektbericht	46
Ablauf	47-48
Herausforderung	49

Schluss

Ausblick	51
Plagiatserklärung	52
DOR & NVS	53
Quellen	54

Konzept

Konzeptänderungen
Händewaschen
People Bouncy

Händewaschen

Änderungen

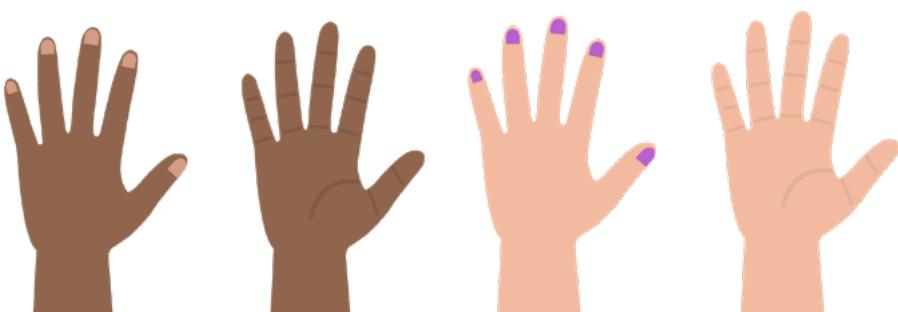
Während der Umsetzung unserer Simulation sind wir bei dem Spiel Händewaschen von dem Konzept abgewichen. Wir haben gemerkt, dass der geplante Durchgang zu starr und langgezogen wirkt. Um dem entgegenzuwirken, haben wir zwei Änderungen vorgenommen.

Wechselnde Hände

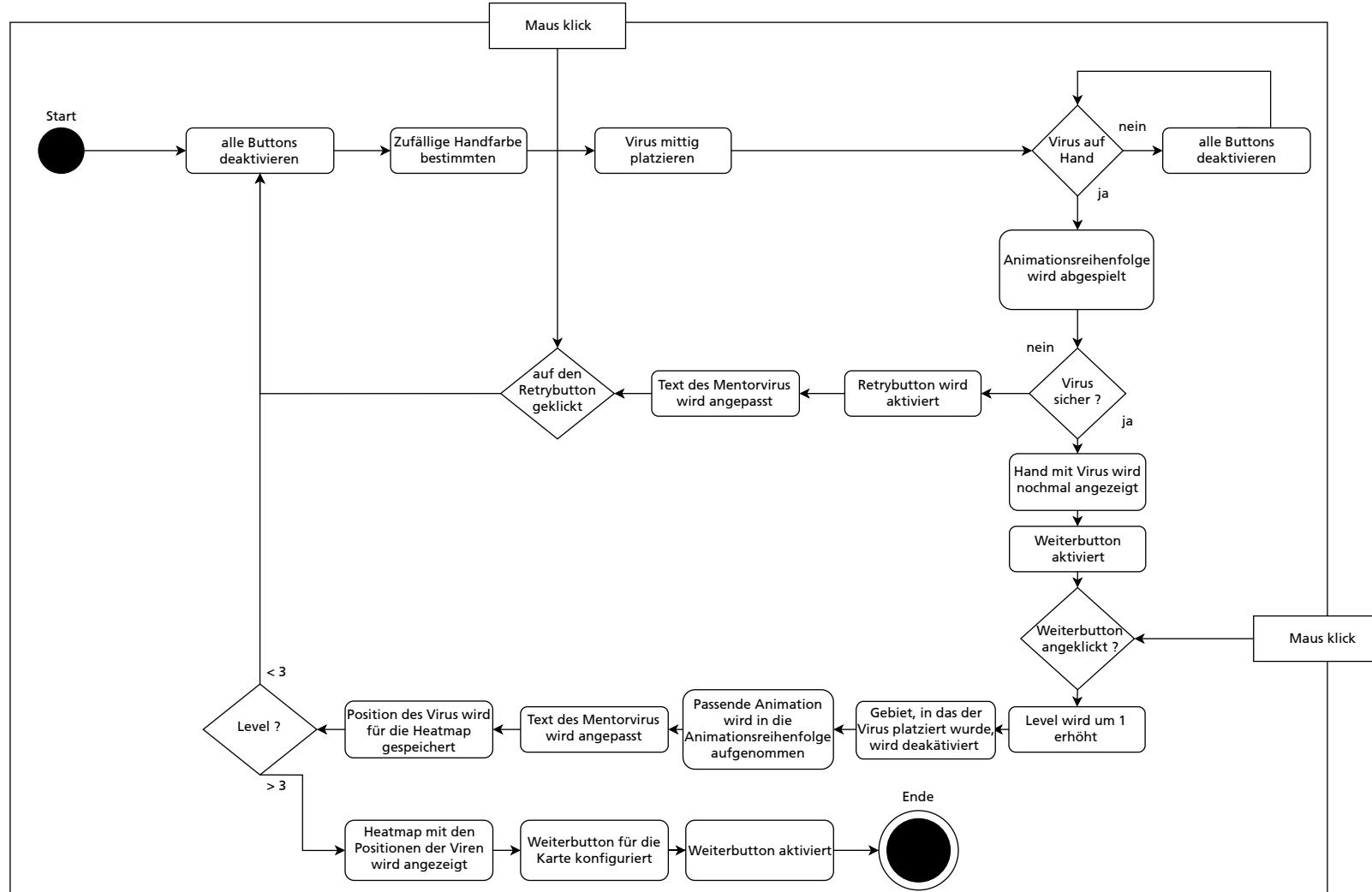
Sobald die nutzende Person in das nächste Level kommt, oder das Level wiederholen muss, ändert sich das Bild der dargestellten Hände. So entsteht der Eindruck, dass man nun eine neue Person infizieren muss. Durch die Abwechslung bleibt die Aufmerksamkeit der nutzenden Person erhalten.

Dynamischer Ablauf

Den wohl größten Effekt hatte die Änderung des Spielablaufs. Im Konzept beschrieben wir diesen als linear und gleichbleibend. Nun ist er jedoch dynamisch gestaltet. Das heißt, nachdem die nutzende Person den Virus platziert hat, wird immer die dazu passende Animation abgespielt. So ist die nutzende Person gezwungen, aufmerksam die Schwachstellen der Händehygiene herauszufinden. Platziert man den Virus wiederholt auf der gleichen Stelle, so wird dieser abgewaschen. Dadurch ist das Spielerlebnis interaktiv und abwechslungsreich gestaltet.



Händewaschen



People Bouncy

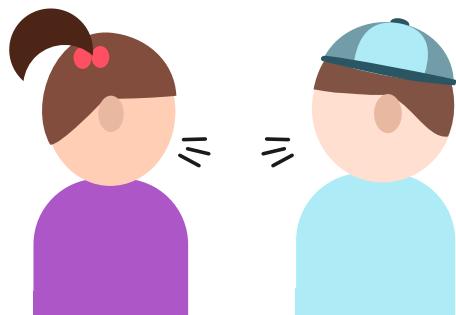
Konzept

Während wir People Bouncy programmierten, haben wir einige Konzepte zum Spielprinzip ergänzt und andere verändert oder verworfen.

Ergänzungen

Als erstes haben wir uns dazu entschieden, dass der Wecker Countdown beim Start des ersten Levels nicht sofort beginnt, um allen Kindern die Möglichkeit zu geben, sich zu orientieren.

Weiterhin haben wir kleine Striche an den Münden der Figuren beim Husten ergänzt, da die Aktion nur mit dem Soundeffekt nicht richtig beim Spielenden angekommen ist.



Veränderungen

Nach einigen Testläufen entschieden wir uns dazu, die Reihenfolge und den Sinn der Level zu ändern.

Im ersten Level werden die People nun vergrößert dargestellt, damit der Raum um sie kleiner wirkt. Auch haben die People so weniger Platz sich zu bewegen und es ist einfacher sich als Erreger zu verbreiten. Durch die Vergrößerung der People sind auch der Virus über deren Kopf und die Striche beim Husten größer und somit besser zu erkennen.

Im zweiten Level haben die People nun zur Hälfte Masken an und befinden sich auf dem Pausenhof. Hier ist die Verbreitung schwieriger, da die Masken die Flugweite reduzieren und die kleineren Personen deutlich mehr Platz haben,

People Bouncy



sich zu bewegen. Hier soll nun deutlich werden, dass Masken nicht besonders viel bewirken, wenn kein Abstand eingehalten wird und zu viele die Maske nicht tragen.

Im dritten Level haben alle People Masken auf und halten, so gut es geht, einen Abstand ein. Jetzt sollte es fast unmöglich sein, sich als Erreger zu verbreiten. So soll letztendlich deutlich werden, dass nur durch Einhalten von Abständen und das Tragen von Masken eine Ausbreitung wirklich eingedämmt werden kann.

Verbreitung trotz Maske

Wir haben bei der Entwicklung natürlich auch andere Ideen ausprobiert, unsere Botschaften zu vermitteln. So haben wir getestet, wie der Spielverlauf erlebt wird, wenn die Masken die Wahrscheinlichkeit ändern, mit der ein Virus

beim Drücken der Leertaste erzeugt wird. Dies fanden wir aber wenig befriedigend, da es sich komisch anfühlt, wenn ein Drücken der Taste keine sichtbare Auswirkung hat. Selbiges gilt für den Fall, wenn Masken den People eine gewisse Wahrscheinlichkeit geben, bei einem Treffer nicht angesteckt zu werden. Die Spielenden werden sich eher fragen, ob das Spiel fehlerhaft sei, als die eigentliche Botschaft zu verstehen.

Letztendlich war eine Flugweiten-Minimierung die Option, welche sich am besten angefühlt hat und auch ohne Erklärung am verständlichsten ist.

Im letzten Level reduzierten wir die Personenanzahl auf Hannah und einen People für unsere Hauptperson. Dies nahmen wir vor, um die Dramaturgie der Szene zu erhöhen und, um die Animation des Zerreißens einheitlich darstellen zu können. Diese war am Ende ein Problem, da

People Bouncy

die People sonst immer zufällig verteilt wurden und wir so nicht einfach ein Video einbauen konnten, in welchem der Screen zerreißt. Daher entschieden wir uns für die oben genannte Szene, um eine immer gleiche Ausgangslage für das Video zu schaffen.

Verworfen

Einige Konzepte haben wir bei unserem Prototyp erstmal ausgelassen oder verworfen, da diese entweder technisch für uns in der Zeit nicht umsetzbar waren, oder doch nicht in die Simulation sollten.

Grafisch entschieden wir uns dazu die Richtungspfeile an den Figuren wegzulassen, da die Figuren bei uns nur in vier Hauptrichtungen schauen können und diese klar zu erkennen sind. Beim Anstecken entfernten wir die Bedingung,

dass die People sich gegenseitig anschauen müssen, um den Erreger zu übertragen, da die Figuren sich zu zufällig zueinander bewegen und drehen. Auch sind wir bei dieser Einschränkung auf das Problem gestoßen, was überhaupt passieren soll, wenn der Virus jemanden von hinten trifft. Wenn der Erreger einfach verschwindet, sieht das zum einen komisch aus und fühlt sich zum anderen auch falsch an. Daher haben wir uns entschieden, diese Beschränkung zu entfernen.



Technik

[View Hierarchy](#)

[Action Hierarchy](#)

[Technische Umsetzung](#)

[Klassendiagramm](#)

View Hierarchy

Wir haben uns am Anfang der Umsetzung dazu entschieden, eine View Hierarchy umzusetzen. Nach dieser strukturierten wir unsere Klassen und Screens. Wir setzten diese so um, dass wir jedem Objekt beliebig viele Unterobjekte zuweisen können.

So haben wir ein „großes“ World Objekt erstellt, welches dann unsere Szenen als Unterobjekte beinhaltet. Die Szenen wiederum beinhalten alle Unterobjekte, welche zu ihnen gehören und so weiter.

Dies erleichterte uns die Arbeit enorm, da die Position der Unterobjekte immer relativ zu ihren Überobjekten definiert ist. So konnten wir zum Beispiel den Virus über den Kopf der jeweils aktiven Figur in People Bouncy sehr einfach programmieren.

Weiterhin wird immer erst das Überobjekt gezeichnet, darüber dann die Unterobjekte. So konnten wir bestimmen, welches Bild über welches, gezeichnet wird.

Auch werden nur die Objekte aktualisiert, deren Überobjekte aktiviert sind. Dies ermöglichte uns einen einfachen Szenenwechsel und sorgte allgemein für einen sehr strukturierten Code mit erleichterter Fehlersuche.

Wir haben uns entschieden, dass alle Klassen ab InteractiveObject Unterobjekte beinhalten können, da in unserer Anwendung nur InteractiveObjects Unterobjekte haben müssen.

Alles in allem ist unsere View Hierarchy eine der wichtigsten Grundlagen, auf welcher unser Prototyp beruht.

View Hierarchy

world

Unterobjekt

Szene Händewaschen

Unter-Unterobjekt

Hände

Unter-Unterobjekt

Button Weiter

Action Hierarchy

Weiterentwicklung

Als wir die View Hierarchy programmiert haben, ist uns aufgefallen, dass wir die Struktur dieser auch nutzen könnten, um Inputs und Updates zu verwalten.

Inputs

So haben wir beschlossen, dass alle Inputs (zum Beispiel ein Klicken mit der Maus), die ein Überobjekt treffen, an dessen Unterobjekte weitergegeben werden. Dies verbesserte die Performance, da die Seite bei einem Mausklick nicht mehr alle Objekte durchgehen und prüfen muss. Lediglich die Unterobjekte, deren Überobjekte auch aktiv sind und angeklickt wurden, müssen kontrolliert werden. Weiterhin ist so eine klare Reihenfolge vorgegeben, in der die Objekte ihre Aktionen durchführen.

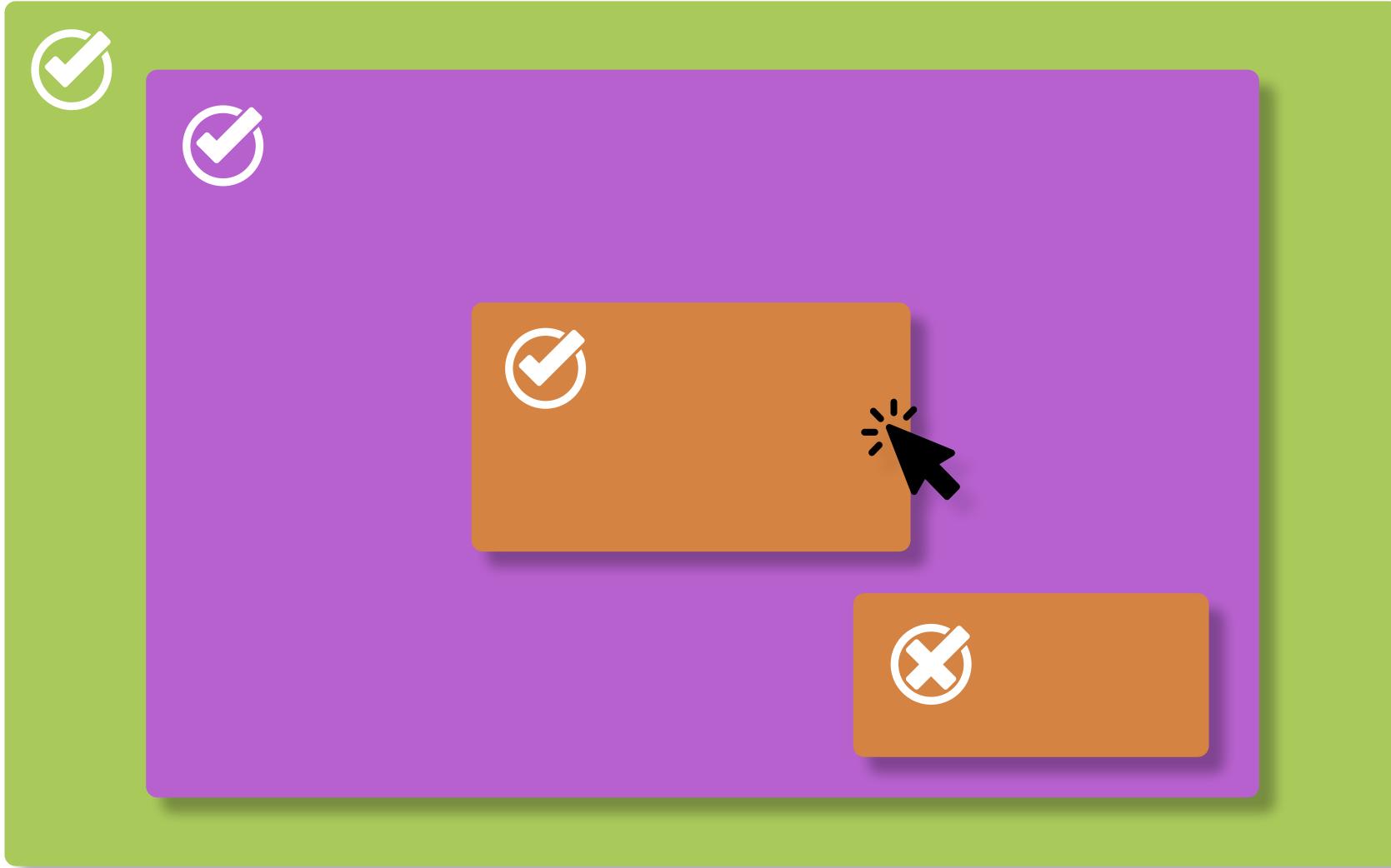
Update() und init()

Ähnlich ist es mit den Funktionen update() und init(), welche auch in einer festen Reihenfolge von den Überobjekten bei den Unterobjekten aufgerufen werden.

Für die init() Funktion haben wir uns entschieden, da der Konstruktor eventuell aufgerufen werden könnte, bevor die p5 draw() ausgeführt wird. Dies könnte dazu führen, dass die Bilder noch nicht geladen sein könnten, da nicht sicher gestellt ist, dass das preload() zu Ende bearbeitet wurde. Deshalb führten wir die init() Methode ein, welche erst mit dem Start der draw() Funktion aufgerufen wird.

Allerdings stießen wir auf das Problem, dass die Kopplung von Über- und Unterobjekt init() zu Problemen in der Definitionsreihenfolge führen kann. Für dieses Problem haben wir leider noch keine gute Lösung gefunden.

Action Hierarchy



Technische Umsetzung

Verlauf der technischen Umsetzung

Gestartet haben wir unsere Umsetzung in der Technik damit, dass wir uns nochmal unser Klassendiagramm angeschauten und daraus die Oberklassen gebaut haben.

So begannen wir damit, erstmal DisplayObject und InteractiveObject zu erstellen und zu testen. Bei den Tests bemerkten wir schnell, dass wir einige wichtige Variablen und Methoden in unserem ersten Klassendiagramm vergessen hatten und ergänzten diese nachträglich.

So arbeiteten wir uns von oben nach unten durch unser Klassendiagramm, bis wir Szenen mit bewegten Objekten und Buttons erstellen konnten. Bevor wir allerdings diesen Teil fertig stellen konnten, kamen noch zwei größere Probleme auf.

Probleme

Globale Variablen

Das erste war, die Bilder, Schriften oder Sounds, welche in der p5setup.js geladen wurden, den anderen Dateien zur Verfügung zu stellen. Zunächst lösten wir dies über ein ENUM Objekt, welches wir an window anhängten, um es überall aufrufen zu können.

Später kam dann der Tipp, dass wir dies auch über Import und Export hätten lösen können, was bei uns aber zu zahlreichen Bugs geführt hat. Im Nachhinein könnten diese Bugs durch einen anderen verzweigteren Bug mit der library p5sound zu tun gehabt haben. Auf Grund dieses Bugs, welchen wir erst kurz vor Ende mit Hilfe lösen konnten, verwendeten wir aber bis zum Schluss unser ENUM Objekt.

Dieses erwies sich generell als sehr praktisch, da wir dort auch andere Konstanten, wie die ID der verschiedenen Hitboxformen speichern und unter einem Namen zugängig machen. So wurde unser Code deutlich lesbarer, da nirgendwo eine Zahl übergeben werden musste, welche keine Koordinate oder Zeit war.

Globale Variablen

Button.js

window.ENUM.SHAPEROCT

People.js

window.ENUM.SHAPEROUND

window.ENUM.IMAGEPEOPLE

window

ENUM.

SHAPE.

RECT: 1

ROUND: 0

IMAGE.

PEOPLE: img,
BUTTON: img,

...

Probleme

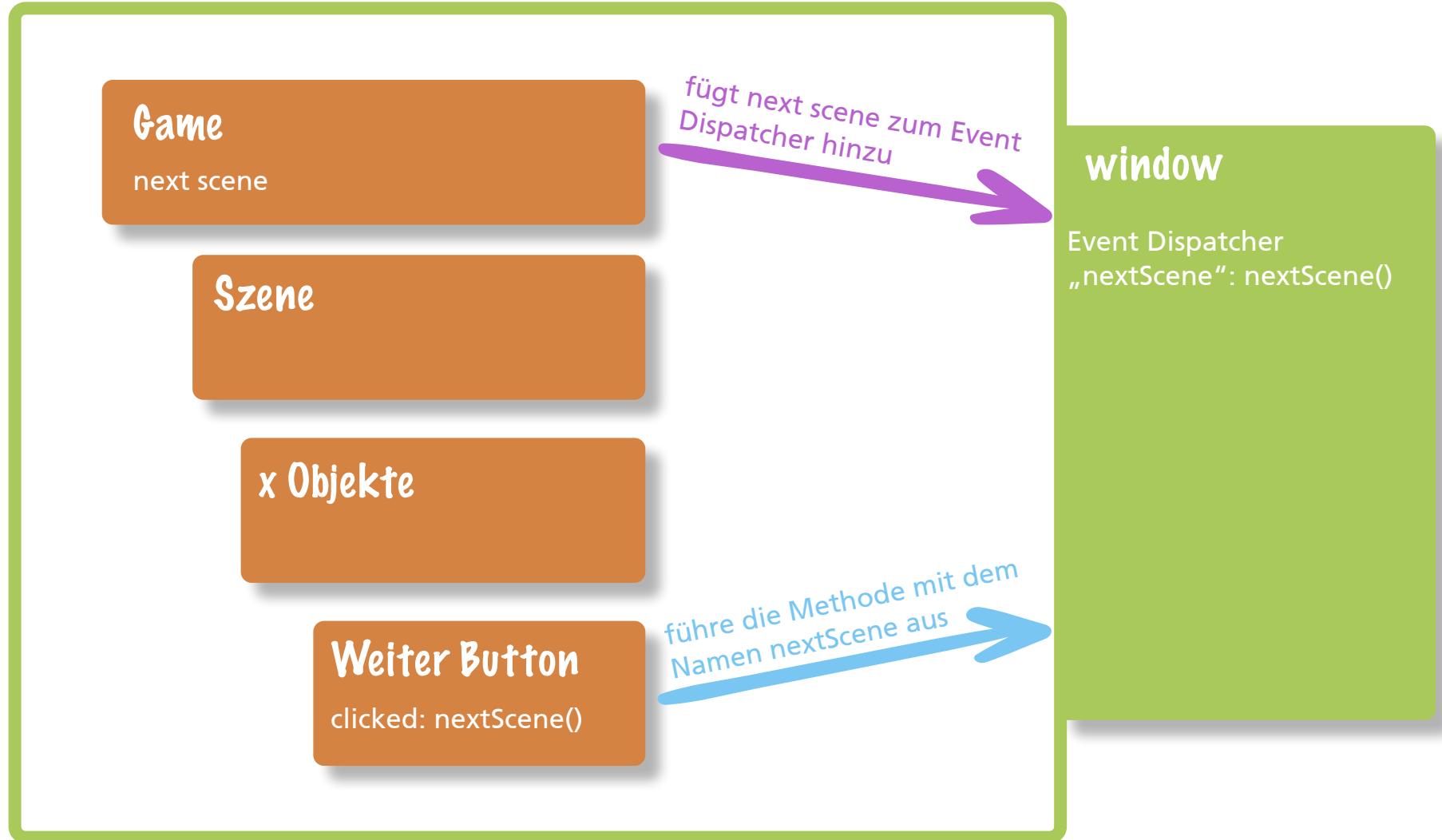
Methoden aus anderen Objekten aufrufen

Das zweite große Problem auf welches wir stießen war, wie wir eine Funktion aus einem übergeordneten Objekt von einem beliebig weit untergeordneten Objekt aufrufen können.

1. Wir stießen dabei auf einen simplen EventDispatcher, welchen wir an unser window anhängen konnten. Dieser sammelt praktisch Funktionen aus verschiedenen Objekten und speichert diese unter einem bestimmten Namen ab. Nun können wir egal aus welchem Objekt dem EventDispatcher Objekt mitteilen, dass dieser eine gewisse Funktion in einem anderen Objekt ausführen soll. Diese Lösung gefiel uns am besten.
2. Die Alternative wäre gewesen, auf das X. Überobjekt zuzugreifen, was allerdings die Wiederverwendbarkeit der Klassen beeinträchtigt hätte, Diese müssten dann immer als X. Unterobjekt an das Überobjekt angehängt werden. Theoretisch hätten wir auch die jeweilige Funktion durch die Unterobjekte übergeben können, was aber wiederum zu noch mehr Verwirrung geführt hätte.

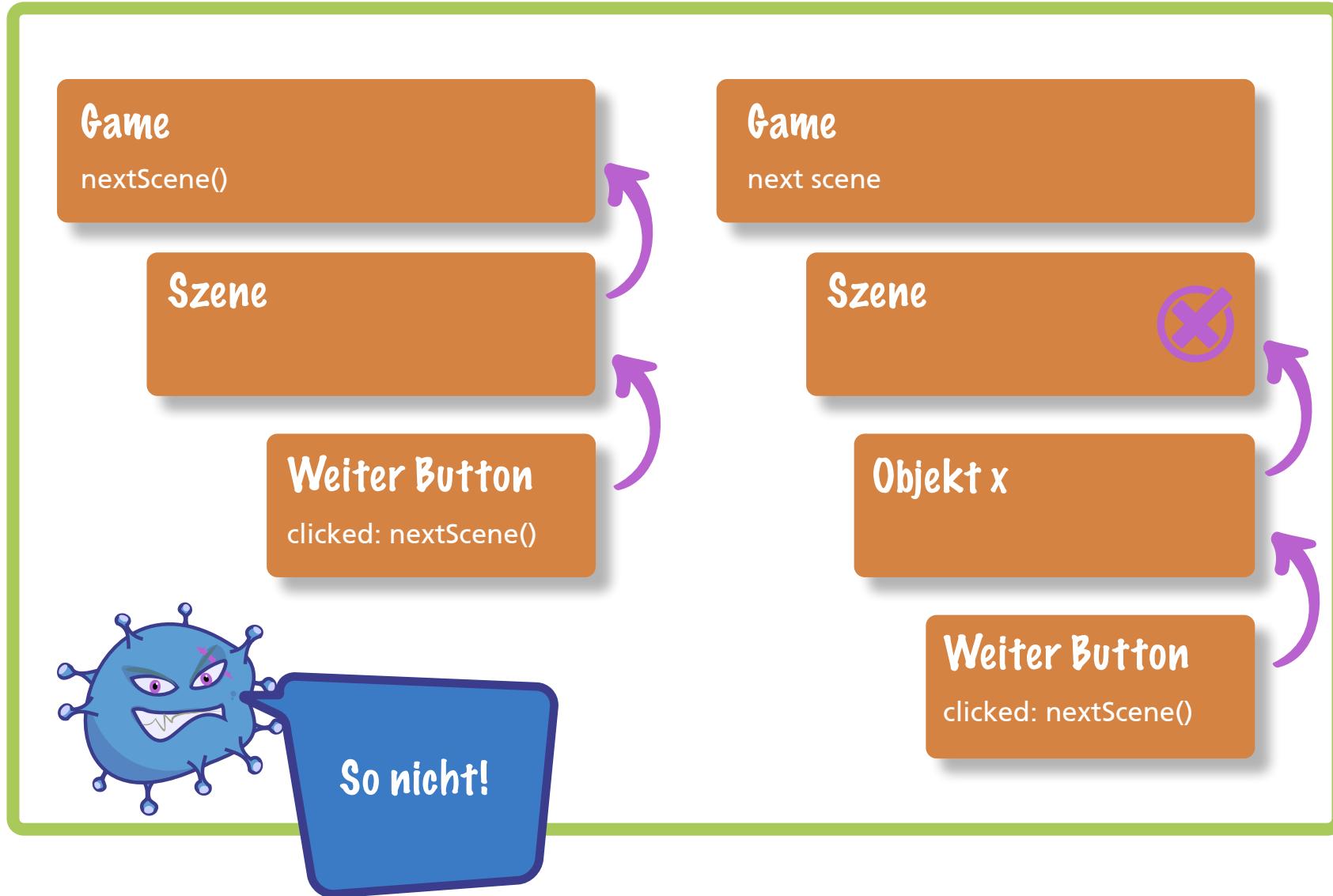
Event Dispatcher

1.



Parent Aufruf

2.



Szenenverwaltung

Nach dem Lösen dieser zwei großen Probleme, begannen wir endlich mit der eigentlichen Umsetzung.

Erste Idee

1. Als erstes hatten wir den Gedanken, dass jedes Spiel eine eigene Unterklasse von Game wird und die einzelnen Level dann als Szenen an diese angehängt werden. Dies erwies sich aber als unnötig, da sich unsere Level nicht so stark von einander unterschieden, um für diese wirklich einzelne Klassen zu erstellen. Der Code in den Szenen hätte sich wahrscheinlich relativ schnell gedoppelt, was wir unbedingt mit unserer

View Hierarchy und mit der Objektorientierung vermeiden wollten.

Bessere Lösung

2. Nach dieser Erkenntnis, kam uns aber die Idee, dass wir diese Dopplungen vermeiden können, wenn wir nur ein Game Objekt als oberstes Objekt erstellen und an dieses dann die einzelnen Simulationen und Screens als Szenen anhängen. So entstand letztendlich unsere Szenenverwaltung und jede Simulation/jeder Screen wurde als eine Unterklasse von Szene geschrieben.

2.

Game

Händewaschen
Level 1/2/3

Flappy Mask
Level 1/2/3

People Bouncy
Level 1/2/3

Szenenverwaltung

1.

Händewaschen

Level 1

Level 2

Level 3

Flappy Mask

Level 1

Level 2

Level 3

People Bouncy

Level 1

Level 2

Level 3



Untersimulationen

Als die Szenen Struktur nun klar war, konnte es wirklich mit unseren Untersimulationen losgehen.

Erste Tests

Zuerst programmierten wir die Untersimulation Händewaschen, da uns diese am einfachsten erschien und wir die obengenannten Konzepte an einem echten Teil unserer Simulation testen wollten.

Grundaufbau

Nachdem unsere Struktur den Test gut überstand, setzen wir eine erste Version der Karte und des Startscreens um.

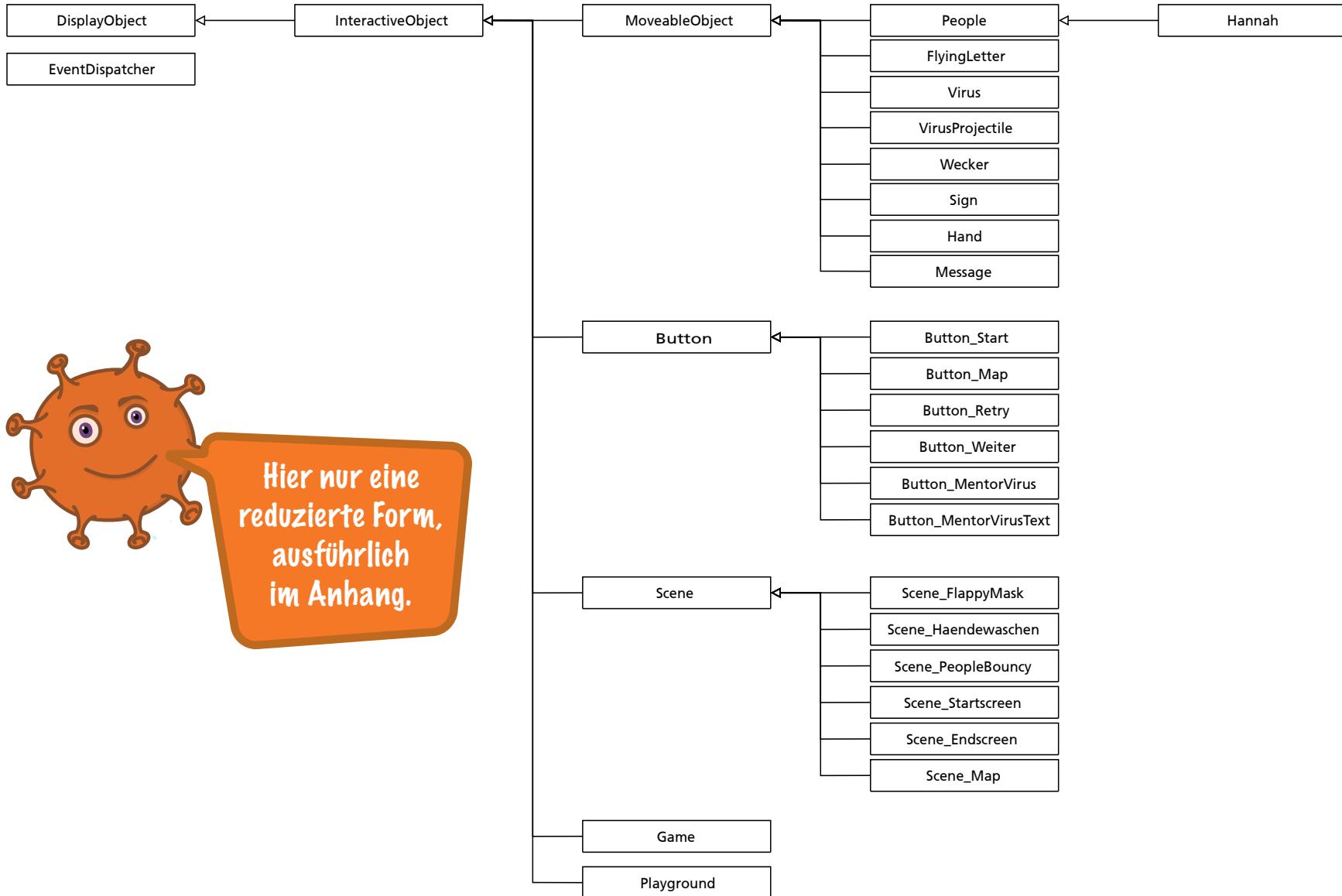
Schließlich begannen wir als letztes mit People Bouncy und überarbeiteten währenddessen auch nochmal die anderen Szenen.

Durch unser modulares Arbeiten konnten wir für People Bouncy viele Objekte aus Händewaschen übernehmen, wie z.B. den Mentorvirus oder den Weiterbutton.

Webseite

Als letztes erstellten wir dann die Webseite und alles, was um den Canvas zu sehen sein sollte. Ein kleines Problem blieb ungelöst, leider entsteht beim Überblenden zu oder nach den Videos ein kurzes Flackern.

Klassendiagramm



Design

Designprozess
Elemente
Animationguide
Sound

Designprozess

Vorgehensweise

Die Assets für unseren Prototyp gestalteten wir stets nach Coding Anspruch. So war jede Woche klar definiert, welcher Teil als nächstes kommt. Dadurch war auch festgelegt, welches Bild als nächstes gestaltet werden musste. Daher kümmerte sich in den Zweierteams, eine Person um Design und die Andere um Coding.

Beispielsweise haben wir beim Start der Programmierung von Händewaschen parallel den Hintergrund geschaffen. Als die Animationen der Hände im Code eingebunden wurden, erstellte parallel jemand anderes die Animationen dafür.

So konnten wir gründlich von Sprint zu Sprint alles Wichtige gestalten und es direkt im Team besprechen. Außerdem haben wir dadurch einen gestalterischen "Flickenteppich" vermieden.

Herausforderungen

Bei der Gestaltung war es eine Herausforderung veränderte Strukturen in der Programmierung zu berücksichtigen. So mussten wir häufig Elemente überarbeiten. Zum Beispiel mussten die Händewasch-Animationen noch einmal neu skaliert und platziert werden. Durch die Vorgaben in der Programmierung wurde die Erstellung der Bilder teilweise immer aufwendiger.

Da wir die verschiedenen Richtungen der People in People Bouncy durch einzelne Bilder umgesetzt haben, mussten wir bei einer Änderung pro Figur vier Bilder anpassen. Dies weitete sich dann auch durch den Status „infiziert“ aus, sodass wir am Ende bei der Erstellung der People mit Mund-Nasen-Schutz 16 Bilder zeichnen mussten.

Elemente

Eigene Elemente

Bei Gestaltung der Screens nutzten wir lizenfreie Vektorgrafiken von freepik.com und änderten diese so ab, dass sie einheitlich aussehen und zum Stil der Simulation passen.

Einige Elemente haben wir auch neu erstellt, um ein noch einheitlicheres Bild zu schaffen. Dabei nutzten wir unsere Hausfarben, sowie einen flachen 2D Stil mit sanften Outlines.



Elemente

Konzept

Technik

Design

Management

Hände

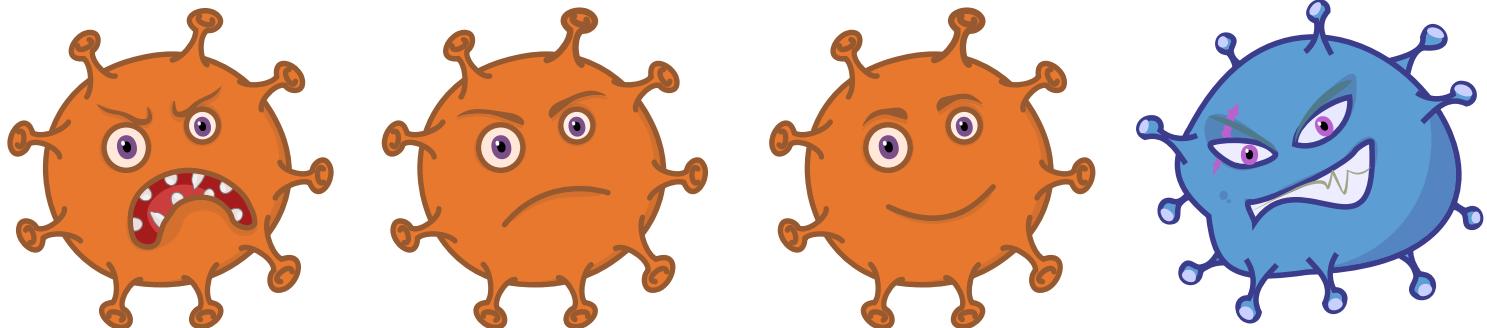
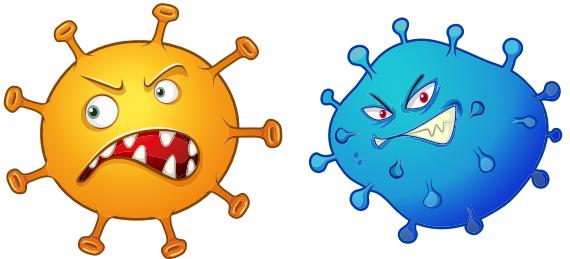
Marie gestaltete die Hände neu, auf die man den Virus platzieren muss, da sie realistischer aussehen sollten und so besser zu den Händewasch-Animationen passen.



Elemente

Viren

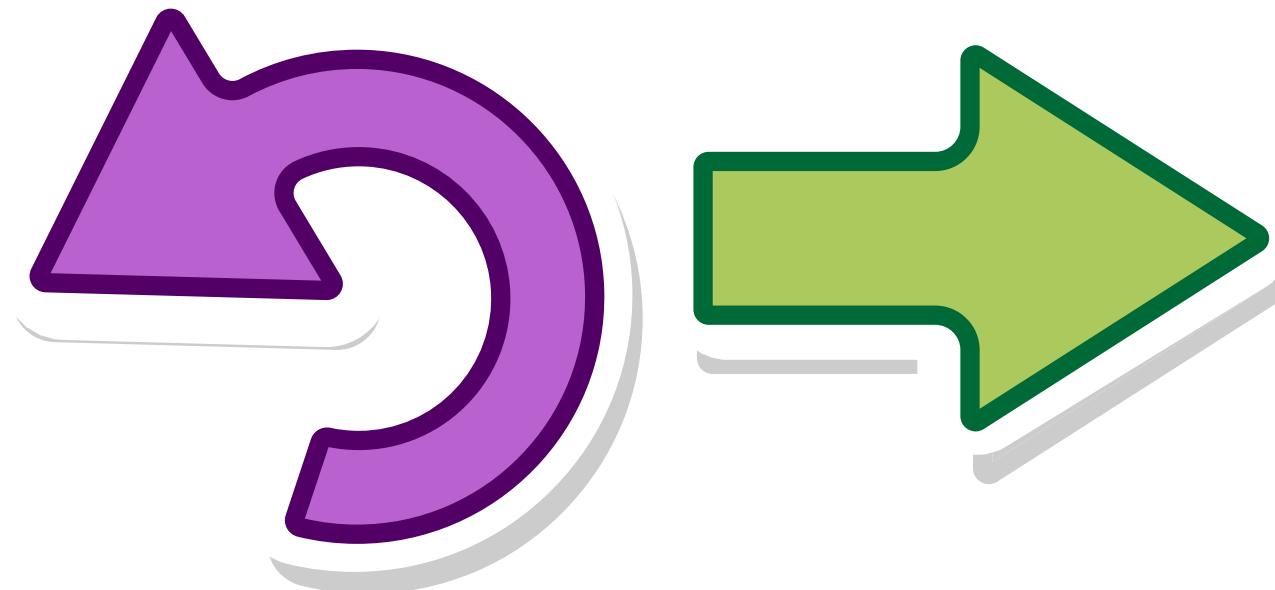
Flo passte unsere beiden Viren an, da sie nicht zu unserem Stil passten. Sie haben nun weniger Glanz und neue Details, wie eine Narbe am Auge und verschiedene Gesichtsausdrücke.



Elemente

Button

Leander bearbeitete den Weiter und Retry Button. Diese waren zuvor zu rundlich und nicht einheitlich gestaltet. Sie haben nun eine deutlichere Kontur, heben sich somit stärker vom Spielfeld ab und sind besser erkennbar.



Elemente

Mädchen

Diandra erstellte ein Mädchen, welches in unserer Anwendung den Traum hat. Die erste Version war zu jung und passte nicht zur Simulation. Außerdem kreierte sie auch auf der Karte das Mädchen, um konsistent zu bleiben. Zuvor wechselte der Charakter zwischen männlich und weiblich, was zu Verwirrung führte.



Animationguide

Easing

Bei der Erstellung der Animationen haben wir besonderen Fokus auf ein passendes Easing gelegt und versucht möglichst keine linearen Bewegungen in die Simulation einzubauen. Die runden Bewegungsabläufe unterstützen unseren freundlichen Charakter. Dazu zählen neben allen Buttons auch die Bewegungen der People in People Bouncy.

EaseOut

Bei den Animationen der interaktiven Objekte (z.B. Button) wird ein EaseOut Easing genutzt, damit die Nutzenden ein möglichst schnelles Feedback erfahren. Dies bedachten wir vor allem in Hinblick auf unsere Zielgruppe. Weiterhin konzentrierten wir uns auf Bewegungs-Animationen, da diese deutlicher für die Zielgruppe sind und die spielerische Atmosphäre unterstreichen.

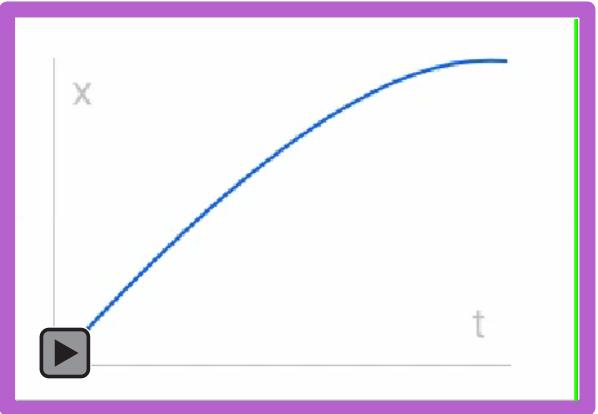
Programmierung

Die meisten Animationen wurden auf eigene Weise in Java Script umgesetzt. Die Animationsgeschwindigkeit entspricht dem linearen Zeitfortschritt pro Frame, dabei ist 1 das Ende der Animation. Die Animationsgeschwindigkeit wird durch die angegebene Easing Funktion in einen nicht-linearen Fortschritt umgewandelt. Diese mathematischen Umwandlungs-Funktionen haben wir von der opensource Plattform easings.net bezogen.

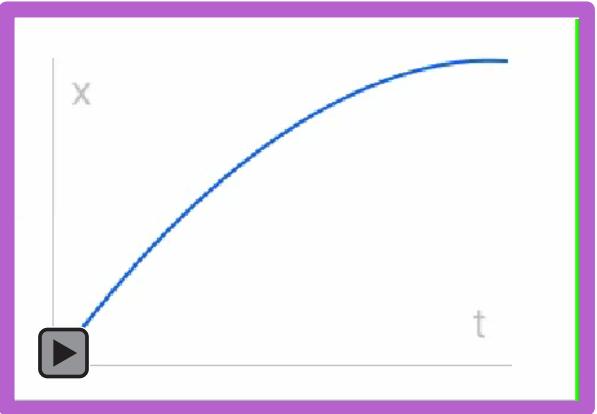
Die von uns verwendeten Easing Funktionen sind nachfolgend dargestellt.

Animationguide

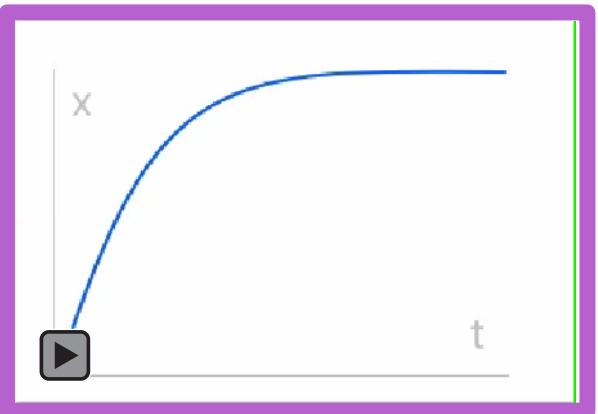
EaseOutSine



EaseOutQuad



EaseOutQuint



EaseOutBounce



Animationguide

Nachrichten Startscreen

Diese Animation soll die Aufmerksamkeit der Nutzenden auf die Mitteilungen im Handy richten, die den erste Einstieg in die Rahmenhandlung geben.

- Animationstyp: EaseOutSine
- Animationsgeschwindigkeit: 0,06



Startbutton

Das Vergrößern des Buttons verdeutlicht, dass er klickbar ist.

- Animationstyp: EaseOutQuint
- Animationsgeschwindigkeit: 0,06



Animationguide

Übergang Traum

Diese Animation wurde mit Hilfe von After Effects erstellt. Der „Swirl“ Effekt verdeutlicht, dass der Charakter sich nun schlafen legt und der Nutzende sich in einem Traum befindet.



Untersimulation Auswahl

Die Anpassung der Transparenz verdeutlicht, dass das Bild der Untersimulation klickbar ist.

- Animationstyp: EaseOutQuint

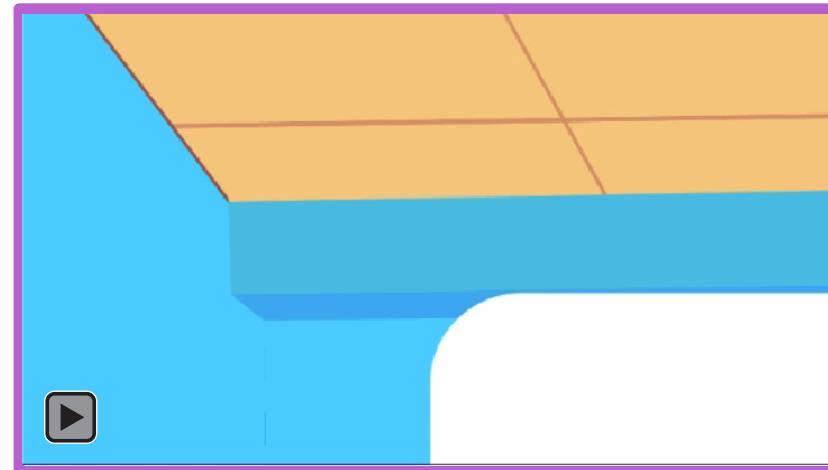


Animationsguide

Schilder und Wecker

Die Schilder und der Wecker „fallen“ in die Spielumgebung hinein, um kurz Aufmerksamkeit auf sich zu ziehen und den Nutzenden eine Orientierung zu bieten. Der Bounce Effekt unterstreicht die optisch „an Fäden aufgehängte“ Darstellung.

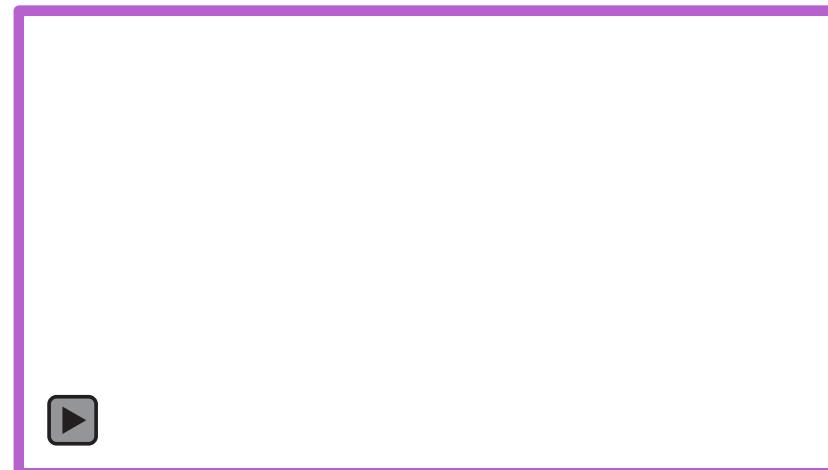
- Animationstyp: EaseOutBounce
- Animationsgeschwindigkeit: 0,04



Hände

Diese Animation lässt die zu infizierenden Hände lebendiger wirken und verdeutlicht, nach Ablauf der Animation, dass die Simulation nun gespielt werden kann.

- Animationstyp: beschleunigte Bewegung

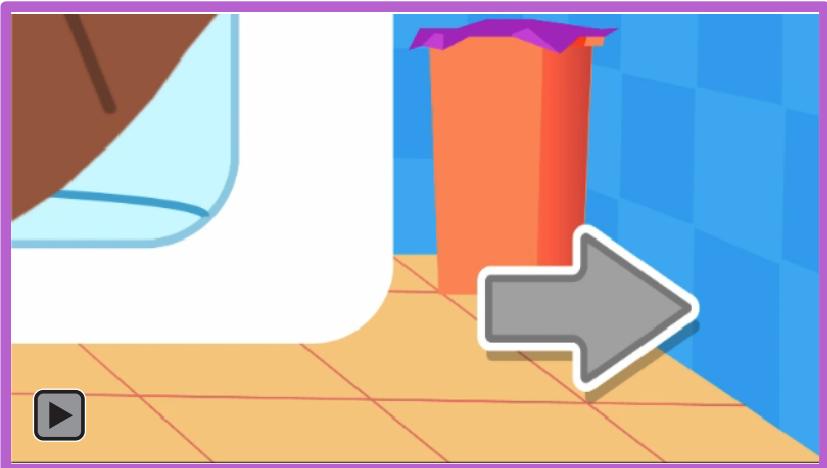


Animationguide

Weiter und Retry Button aktiv

Sowohl der Weiter als auch der Retry Button ändern ihre Farbe und werden größer, wenn sie aktiviert sind. Diese Animation lenkt den Fokus der Nutzenden auf die Button, damit verständlich ist, dass man sie nun klicken muss.

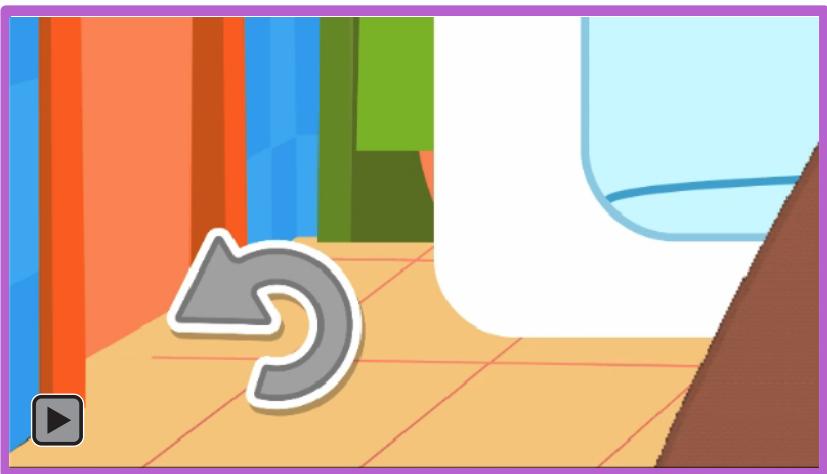
- Animationstyp: EaseOutSine
- Animationsgeschwindigkeit: 0,12



Weiter und Retry Button hover

Diese Animationen zeigen einerseits an, dass die Elemente anklickbar sind und verdeutlichen auch, was beim Anklicken passieren wird. Der Retry Button dreht sich nach links, um die Wiederholung des Levels zu verdeutlichen. Der Weiter Button verlängert sich nach rechts, um anzusehen, dass die Nutzenden so in das nächste Level gelangen.

- Animationstyp: EaseOutQuint
- Animationsgeschwindigkeit: 0,06

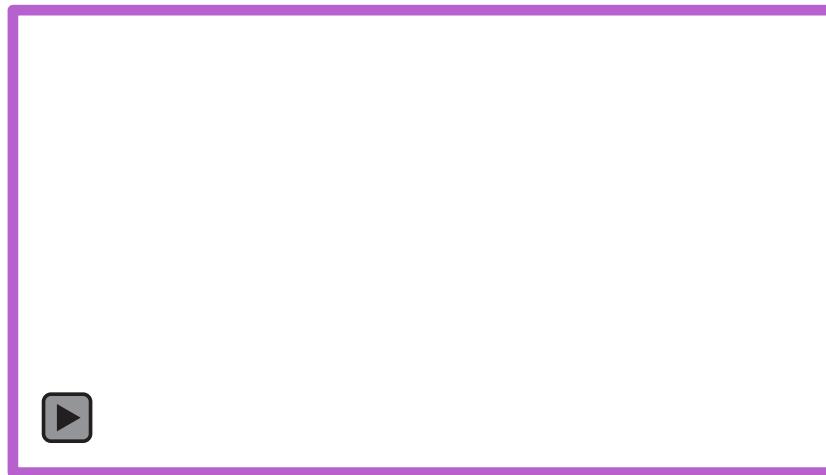


Animationsguide

Mentorvirus Wiggle

Der Mentorvirus rotiert, um zu verdeutlichen, dass er ein klickbares Objekt ist.

- Animationstyp: EaseOutQuad
- Animationsgeschwindigkeit: 0,16



Mentorvirus Sprechblase

Die Sprechblase des Mentorvirus fährt aus und ein, um Aufmerksamkeit auf sich zu lenken, wenn es eine neue Mitteilung gibt.

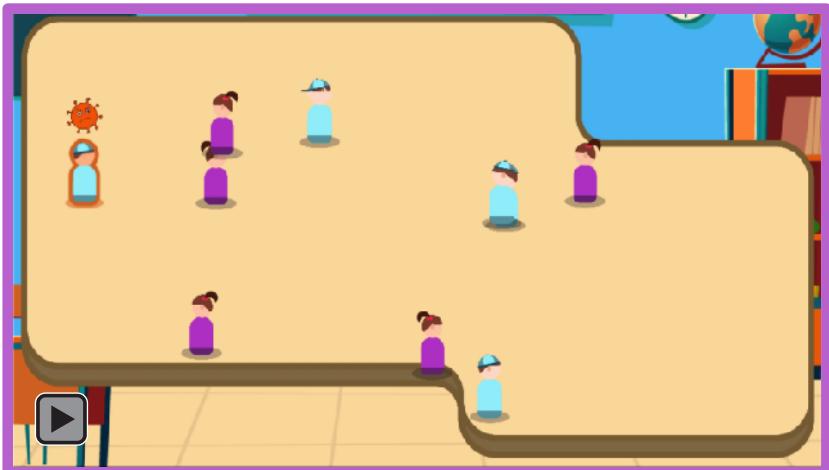
- Animationstyp: EaseOutSine
- Animationsgeschwindigkeit: 0,1

Animationguide

People Bewegung und fliegende Viren

Die People in People Bouncy, sowie die Viren, die sie verbreiten, beschleunigen sich in der Bewegung, da dies natürlicher wirkt. Dennoch konnten wir kein Easing einbauen, da sie eine unbekannte Strecke zurücklegen und die Dauer der Bewegungen unterschiedlich lang ist.

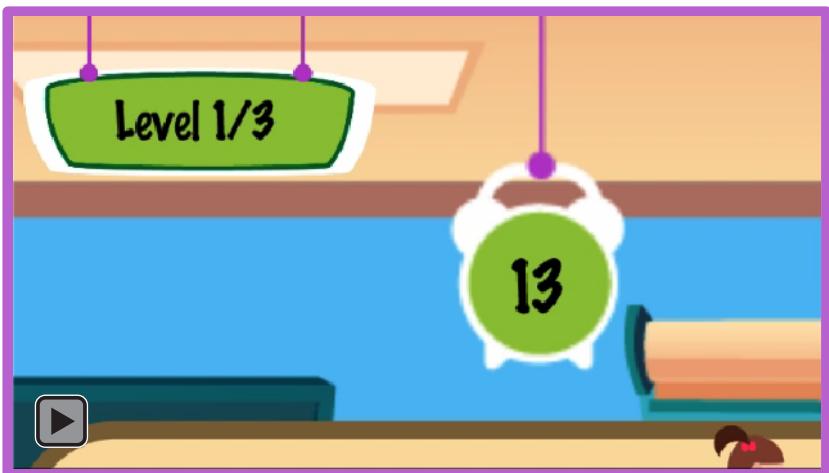
- Animationstyp: beschleunigte Bewegung



Wecker klingeln

Diese Animation ist besonders auffällig gestaltet, da sie die Aufmerksamkeit der Nutzenden auf die Zeit lenken soll, um sie zu warnen, dass das Level nur noch 10 Sekunden dauert.

- Animationstyp: EaseOutSine
- Animationsgeschwindigkeit: 0,2



Animationguide

Zerreissen Ende

Diese Animation erstellten wir in Adobe XD erstellt. Sie soll die Nutzenden überraschen und schockieren, da sie anschließend ihr Handeln und ihre Hygienepraxis überdenken sollen.

- Animationstyp: Easeln
- Animationsgeschwindigkeit: 1 Sekunde



Sound

Soundeffekte

Wir haben unsere Soundeffekte selbst aufgenommen, da wir besondere Ansprüche an das Timing hatten. Weiterhin entschieden wir uns allgemein für unverzerrte Soundeffekte, da sie realer wirken und so unseren Aufklärungscharakter unterstützen.

Händewaschen

Gerade bei den Händewaschanimationen sollte der Sound so gut es geht zu den gezeigten Bildern passen. Dies war uns wichtig, da der Sound so zum einen realistischer wirkt, zum anderen dadurch auch die Animation echter erscheint. Beim Händewaschen liegt darauf ein besonderer Fokus, da die Animation die Anleitung zum richtigen Händewaschen für die Nutzenden sein soll.

People Bouncy

In People Bouncy variieren wir die Soundeffekte „husten“ und „niesen“. Auch hat jeder von uns ein Set an Sounds aufgenommen, um noch mehr Varianz reinzubringen.

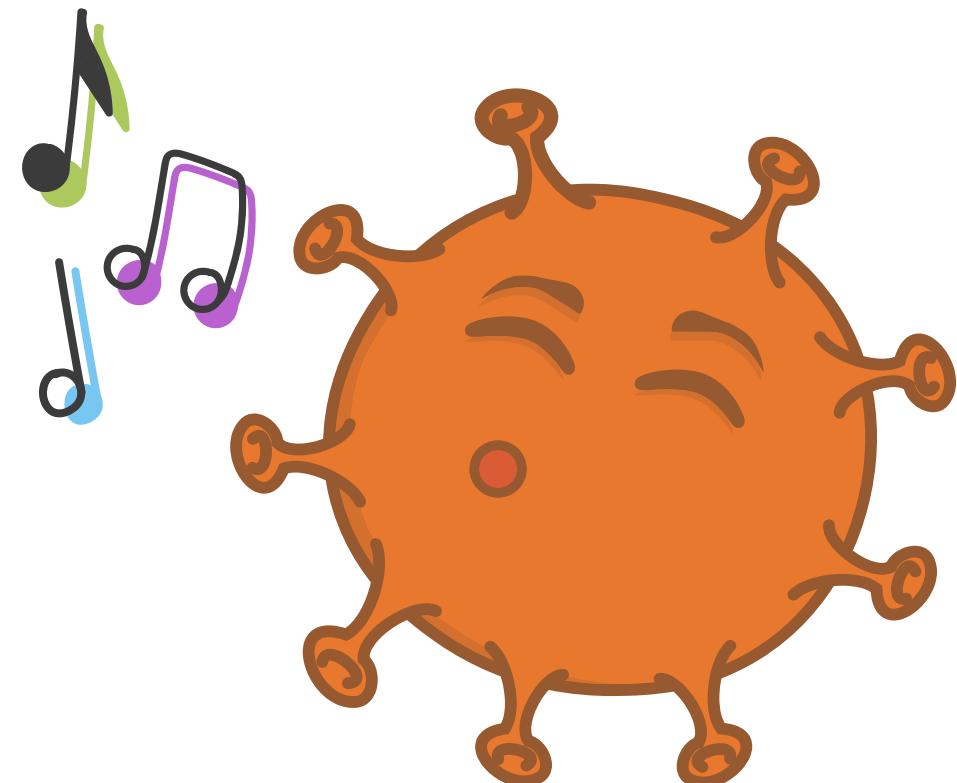
Hintergrundgeräusche

Bei den Hintergrundgeräuschen begannen wir in People Bouncy. Wir suchten Umgebungsgeräusche von einem Spielplatz und einem Klassenzimmer heraus, in der Hoffnung eine Schulatmosphäre zu erzeugen. Auch bauten wir in dem 3. Level Windgeräusche ein, um die düsterere Stimmung zu unterstützen. In unserer finalen Szene steigerten wir diese Geräuschkulisse, indem wir sie um den Krach eines richtiges Gewitter ergänzten. Wir bemerkten schnell, dass uns diese Geräusche ausreichten und

Sound

entschieden uns damit gegen eine Hintergrundmusik. Weiterhin hatten wir in der kurzen Zeit, die uns blieb, keine Möglichkeit diese Geräusche selbst aufzunehmen und mussten deshalb auf Royalty-Free Aufnahmen zurückgreifen.

Leider fanden wir keine passenden Hintergrundgeräusche für die Untersimulation Händewaschen. Wir haben es sowohl mit Geräuschen von Wasser in allen möglichen Formen, als auch mit Umgebungsgeräuschen aus echten Toiletten versucht. Wir haben allerdings keine Tonaufnahme gefunden, welche auch nur ansatzweise zu unserer Darstellung passte. Zum Selbsterstellen fehlte uns am Ende die Zeit, darum ließen wir diese Hintergrundgeräusche in unserem Prototyp aus.



Management

Teamorganisation

Projektbericht

Ablauf

Herausforderungen

Organisation

Team

Für die Phase der Umsetzung behielten wir unsere Teamregeln bei. Wir legten weiterhin viel Wert auf eine offene Kommunikation, einen respektvollen Umgang miteinander und einen offenen Informationsaustausch. Wie in der Konzeptphase bereits erlebt, spürten wir nun auch in der Umsetzung, die Vorteile von zuvor vereinbarten Teamregeln.

Tools

Auch die Tools zur Strukturierung und effektiven Zusammenarbeit blieben die gleichen.

Conceptboard nutzten wir weiterhin, um Mindmaps zu erstellen, Ideen zu sammeln, oder Zwischenergebnisse festzuhalten.

Wir nutzten Trello, um unsere einzelnen Sprints zu organisieren. Den Backlog befüllten wir dabei zu Beginn mit allen Themen bis zur Abgabe, die zu erledigen waren. Wöchentlich entschieden wir uns dann, welche Aufgaben wir in der kommenden Woche bearbeiten wollten. So behielten wir über die gesamte Zeit einen guten Überblick über unsere Aufgaben.

Um unsere gesamte Arbeitsphase protokollarisch festhalten zu können, schrieben wir für jedes Meeting einen Projektbericht. In diesem hielten wir fest, was wir im jeweiligen Meeting bearbeitet und besprochen haben, welche Aufgaben oder Maßnahmen daraus entstanden sind und natürlich auch, wie lange eine Aufgabe dauerte und bis wann eine Maßnahme zu erledigen war. Auf der nächsten Seite ist beispielhaft ein Auszug aus dem Projektbericht vom 15.06.2020.

Projektbericht

Projektmeeting Team 2 15. Juni 2020

1. Ziel des Meetings

Abgleich der Aufgabe, die wir über das Wochenende erledigt haben

2. Teilnehmer

Diandra Hermann, Florian Grünewald, Marie Sasse, Leander Schmidt

3. Tagesordnung, Notizen, Beschlüsse, Probleme

THEMA	VERANTWORTLICH	DAUER
<p>Neue Elemente</p> <ul style="list-style-type: none"> - Leander: Buttons Anmerkungen aus dem Team: Breiten/Stärken anpassen, dass die beiden Button sich ähnlicher sehen - Marie: neue Hände Anmerkungen aus dem Team: linke Hand Daumen auf der falschen Seite - Diandra: Mädchen Endscren, Karte, Startscreen Anmerkungen aus dem Team: Gesicht anpassen, passt nicht zum Stil des Körpers <p>Gliederung: wir sind uns unsicher, ob die Abschlussdoku eine Zusammenfassung unseres gesamten Projekts darstellt oder, ob die 3 Dateien (Recherche, Konzeptdoku, Endoku) „eine Datei“ am Ende ergeben. Sprich, wir würden dann keine Sachen doppeln, auf die wir bereits in der Konzept Doku eingegangen sind, und diese sich nicht geändert haben. Das werden wir morgen die Tutoren fragen.</p>	alle	30 Minuten
Skalierung der Screens, damit es auf allen Geräten gleich ist und wir nicht jedes Objekt skalierbar machen müssen: Wir einigen uns auf die Skalierung 1366x768 px	Beschluss	15 Minuten

4. Aktionspunkte

MASSNAHME	VERANTWORTLICH	DEADLINE
Sebastian Haase wegen einem Layout-Coaching schreiben	Marie	15.06.2020
Jeder passt eigene Elemente nochmal an	alle	16.06.2020

5. Nächstes Meeting

Datum	Dienstag, 16.06.2020	Uhrzeit	10 Uhr	
Ziel	Neue Elemente in Simulation einbauen			

Ablauf

Testphase

Wir sind nach der Konzeptphase mit einer linearen Planung in die Umsetzungsphase gestartet. Als erstes programmierten wir die Grundstruktur unserer Klassen und Objekte für alle Simulationen, sodass wir etwas haben, worauf wir aufbauen konnten. Dies probierten wir anhand vieler einzelner Tests mit Test-Videos/Bilder und klickbaren Objekten aus. Als wir uns sicher waren, dass alles wie geplant eingebunden werden kann, sind wir in die Realisierung unseres Prototyps gestartet.

Untersimulation Händewaschen

Hier haben wir zunächst mit der Untersimulation Händewaschen begonnen. Als das Grundgerüst stand, begannen wir parallel mit dem Startscreen und der Karte.

Flappy Mask oder People Bouncy?

Nachdem wir Händewaschen abgeschlossen haben, wollten wir Flappy Mask beginnen. Jedoch waren wir uns unsicher, ob wir sowohl Flappy Mask als auch People Bouncy in der restlichen Zeit schaffen. Daher berieten wir uns mit unserem Coding Dozenten Garrit Schaap und sind zu dem Schluss gekommen, dass wir mit People Bouncy mehr lernen können als mit Flappy Mask – selbst wenn es nicht fertig wird. Außerdem fanden wir, dass People Bouncy viel mehr den Kern unserer gesamten Simulation vermittelt, als es Flappy Mask könnte.

Hier erarbeiteten wir zuerst wieder die Grundstruktur, damit wir alle Level nur noch zusammenbauen mussten.

Ablauf

Produktvideo

Als sich People Bouncy langsam dem Ende neigte, begannen wir mit unserem Produktvideo. Dazu erstellten wir zunächst ein Storyboard, aus dem ein Animatic entstand. Dieser war eine Blaupause für unsere anschließende Animation.

Wir haben es geschafft!

Nun sind wir am Ende der Umsetzungsphase und haben sowohl den Startscreen, Händewaschen, die Karte, People Bouncy und das Ende technisch in einem Prototyp umgesetzt.

Da wir tägliche Team Meetings geplant haben, konnten wir uns gegenseitig gut helfen und unsere Stärken gleichmäßig nutzen. Wir sind Level für Level vorgegangen, um eine gute Übersicht zu behalten. So sind wir nie im Chaos versunken, da wir wussten, was im nächsten



Sprint ansteht. Zwar lief nicht alles immer nach Zeitplan, aber die Struktur hat uns geholfen den Überblick zu wahren.

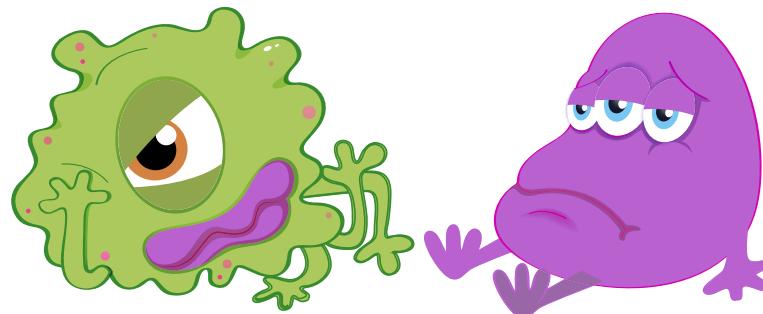
In diesem ganzen Prozess rotierten wir stets zwischen den Aufgaben, damit jeder in allen Bereichen sein Wissen erweitern konnte. Auch wenn nicht alles immer glatt lief, haben wir doch unglaublich viel voneinander gelernt und können stolz auf unseren Prototyp sein.

Herausforderung

Blockade

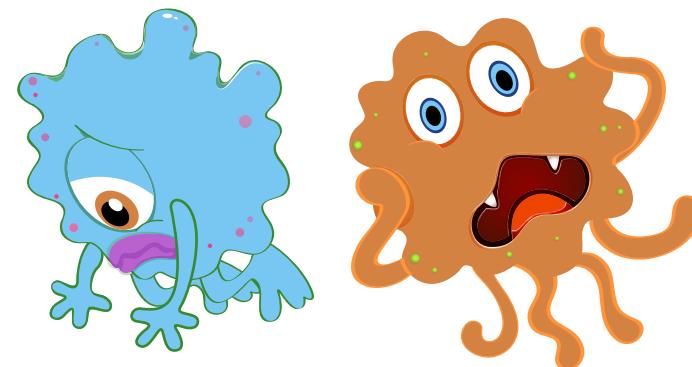
Zu Beginn der Umsetzungsphase hatten wir circa eine Woche lang Probleme uns zu organisieren. Wir fanden keinen Anfang und das gemeinsame Arbeiten fiel uns schwer.

Eine besondere Herausforderung war das interdisziplinäre Arbeiten: aus der eigenen Komfortzone herauszugehen und etwas zu erarbeiten, in dem wir uns noch nicht so sicher fühlten.



Lösung

Wir haben schließlich realisiert, dass wir umdenken und uns neu strukturieren müssen. Wir entschieden uns dazu in Zweierteams zu arbeiten. Beide Teams arbeiteten parallel in Discord, so waren alle immer erreichbar. Dieses Vorgehen half uns interdisziplinär zu arbeiten. Wir trafen uns jeden Tag und legten morgens die Aufgaben der Teams fest. Die täglichen Sprints gewährleisteten einen regen Austausch untereinander.



Schluss

Ausblick

Danksagung

Plagiatserklärung

DOR & NVS

Quellen

Ausblick

Auch wenn unser Prototyp ziemlich alle Konzepte enthält, fehlen dennoch einige Aspekte.

So haben wir Flappy Mask nicht umgesetzt. Dies könnte in der Zukunft noch erstellt werden.

Weiterhin hätten wir bei Händewaschen die Hitboxen der Hände in genauere Bereiche unterteilt, wie bei der HeatMap, oder es mit Musik hinterlegt. Dort könnten auch mehr Hintergrundbilder sinnvoll sein, um mehr Abwechslung in die Simulation zu bringen.

Schließlich wäre Dreck auf der Hand denkbar, welcher als Hilfestellung dient, wenn jemand ein Level zu oft hintereinander nicht schafft.

People Bouncy hingegen wäre durch Objekte, die auf dem Spielfeld platziert sind, spannender geworden. Wir hätten z.B. Klettergerüste auf

dem Pausenhof eingefügt, um die sich die Kinder versammelt hätten. Außerdem können weitere Level, wie „Maske aber kein Abstand“, „Herdenimmunität“ oder eine „zeitliche Immunisierung des Wirts“ eingebaut werden.

Durch eine besser Bewegungssteuerung und mehr Drehrichtungen könnte die Untersimulation auch noch deutlich natürlicher wirken. So würden ihre Botschaften noch besser ankommen. Auch hätten wir gerne mehr Assets und Sounds selbst kreiert.

Abschließend wäre ein Test in Schulen oder generell bei unserer Zielgruppe noch sehr interessant geworden, da wir genauer auf die Bedürfnisse hätten eingehen können. Weiterhin wäre es wichtig zu wissen, ob der Aufklärungsfaktor, um das sich unser ganzes Projekt dreht, wirklich ankommt.

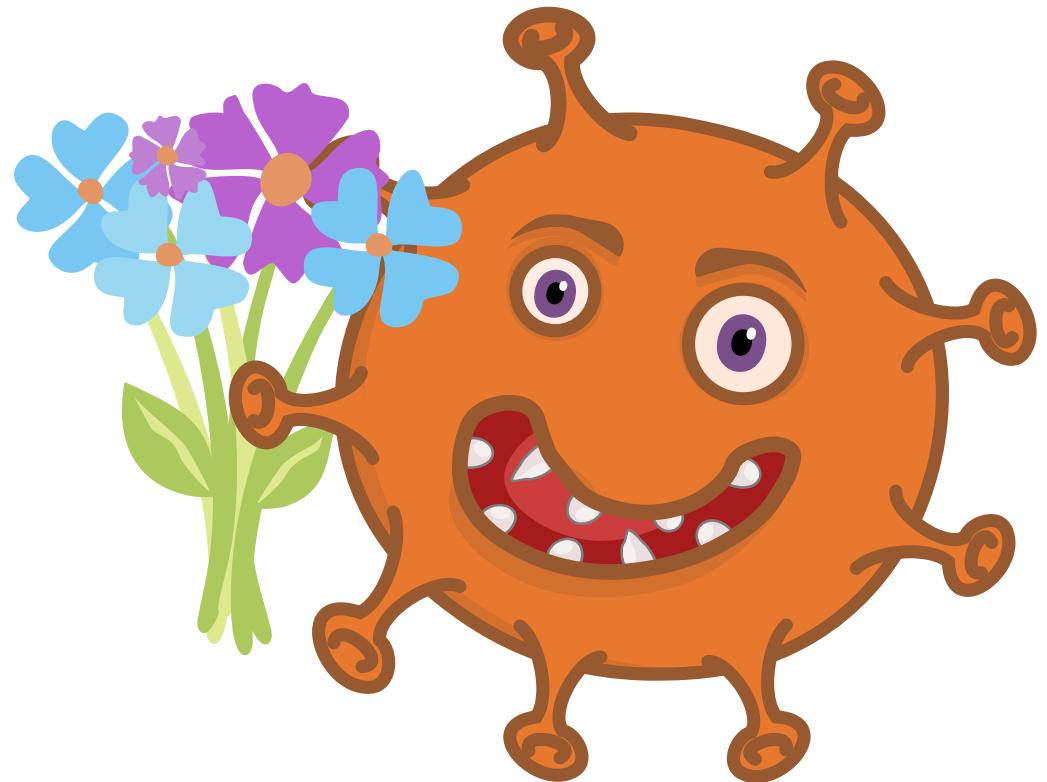
Danksagung

Vielen Dank

In diesem Semester drehte sich alles rund um die COVID-19 Pandemie. Deshalb fand das gesamte Semester präsenzfrei statt. Hier möchten wir einen besonderen Dank an die Dozenten Prof. Andrea Krajewski, Prof. Tsune Tanaka, Garrit Schaap, und Dieter Stasch ausrichten, ohne die dieses Semester nicht möglich gewesen wäre.

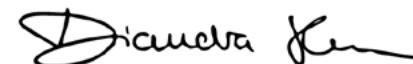
Auch möchten wir den Dozenten Enes Ünal, Max Pelzer, Steffen Kotschi, Tomas Zebis, Martin Haas und Sebastian Haase für Ihre Unterstützung in Vorlesungen und Coachings danken.

Nicht zuletzt ein großes Lob auch an unsere Tutoren Alexander Korolkov, Oliver Ivic-Matijas, Elina Faber, Jan Meininghaus, Anne Schneider und Maximilian Brandl, die uns stets tatkräftig unterstützten.



Plagiatserklärung

Wir versichern, dass wir die Arbeit selbstständig und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt und dass wir alle Stellen der Arbeit, die aus anderen Werken dem Wortlaut oder dem Sinne nach entnommen sind, kenntlich gemacht haben. Darüber hinaus haben wir keine Arbeit mit ähnlichem Inhalt an einer anderen Stelle eingereicht.



Darmstadt, 08. Juli 2020
Diandra Hermann



Eberstadt, 08. Juli 2020
Florian Grünewald



Alzenau, 08. Juli 2020
Marie Sasse

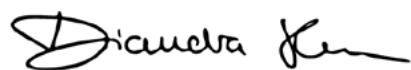


Eberstadt, 08. Juli 2020
Leander Schmidt

DOR & NVS

Delegation of Rights and Non Violation Statement

Be[at] the Virus is a project, developed in the framework of the study course Interactive Media Design of Hochschule Darmstadt - Faculty of Media, during the Summer Term 2020. It is a group work of: Diandra Hermann, Florian Grünewald, Marie Sasse and Leander Schmidt. It was mentored by: Prof. Andrea Krajewski, Garrit Schaap, Dieter Stasch. We herewith delegate the non exclusive and timewise non restricted rights to publish and present the results of the project Be[at] the Virus to the Professors of Hochschule Darmstadt and to the coaches directly connected to the academic supervision of this project, named above. In the same time the student project team declares that with the project no intellectual properties rights of third parties have been harmed.



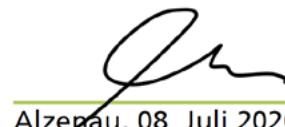
Darmstadt, 08. Juli 2020

Diandra Hermann



Eberstadt, 08. Juli 2020

Florian Grünewald



Alzenau, 08. Juli 2020

Marie Sasse



Eberstadt, 08. Juli 2020

Leander Schmidt

Quellen

Bildquellen

- S. 29 Kunst Vorlage „Designed by pch.vector / Freepik“
- S. 49 Viren Vorlage „Designed by brgfx / Freepik“
- S. 43 Noten Vorlage „Designed by Freepik“
- S. 53 Blumen Vorlage „Designed by lesyaskripak / Freepik“

Zur Erstellung der Simulations-Screens, haben wir frei lizenzierte Bildelemente von Freepik als Vorlage genutzt und abgewandelt eingesetzt.

Startscreen

- Bettdecke „Designed by upklyak / Freepik“
- Poster „Designed by Freepik“

Karte

„Designed by brgfx / Freepik“

Händewaschen

Badezimmer „Designed by upklyak / Freepik“

People Bouncy

- Klassenraum „Designed by upklyak / Freepik“
- Schule außen „Designed by upklyak / Freepik“

Soundquellen

Die meisten Soundeffekte in der Simulation wurden selbst aufgenommen.

Dream Transition sound from www.fesliyanstudios.com

Endscreen tear sound obtained from <https://www.zapsplat.com>

Wind sound obtained from <https://www.zapsplat.com>

Storm sound obtained from <https://www.zapsplat.com>

Children outside sound obtained from <https://www.zapsplat.com>

Classroom full of children sound obtained from <https://www.zapsplat.com>

Videoquellen

- S. 35 Easings von easings.net