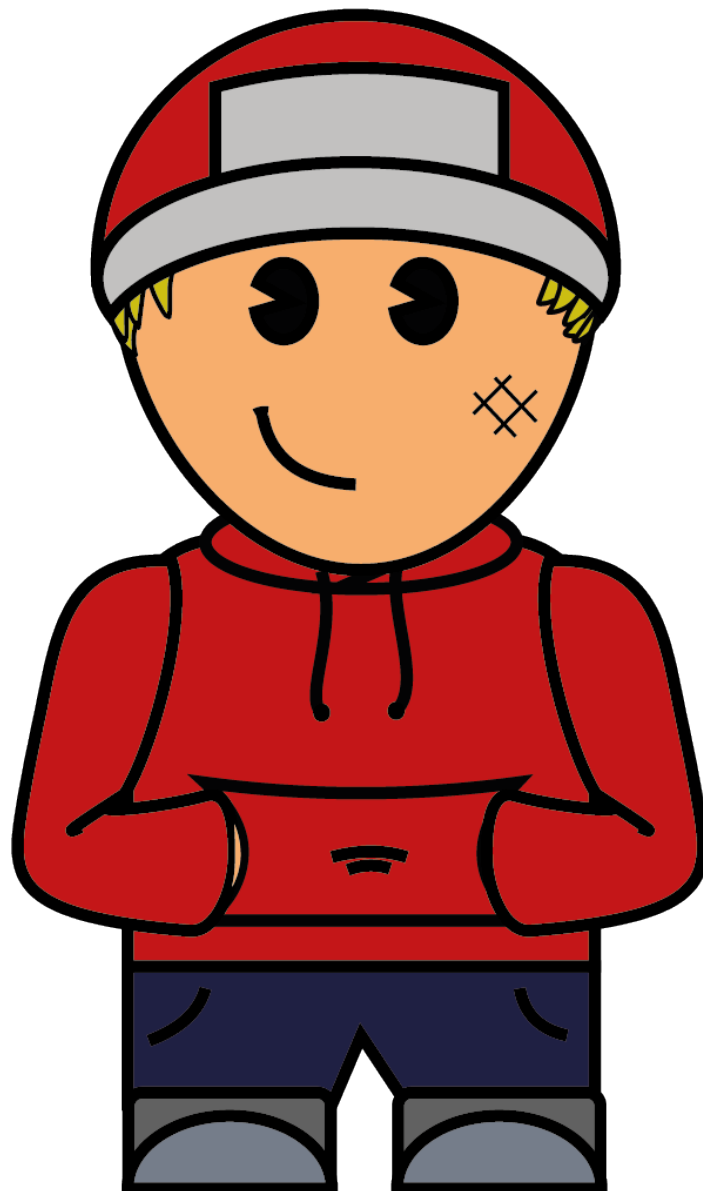


# Bernd bricht aus!

Ein Projekt von:

Fillip Czogallik  
Michael Kettenburg  
Bernd Henke  
Mario Schulz

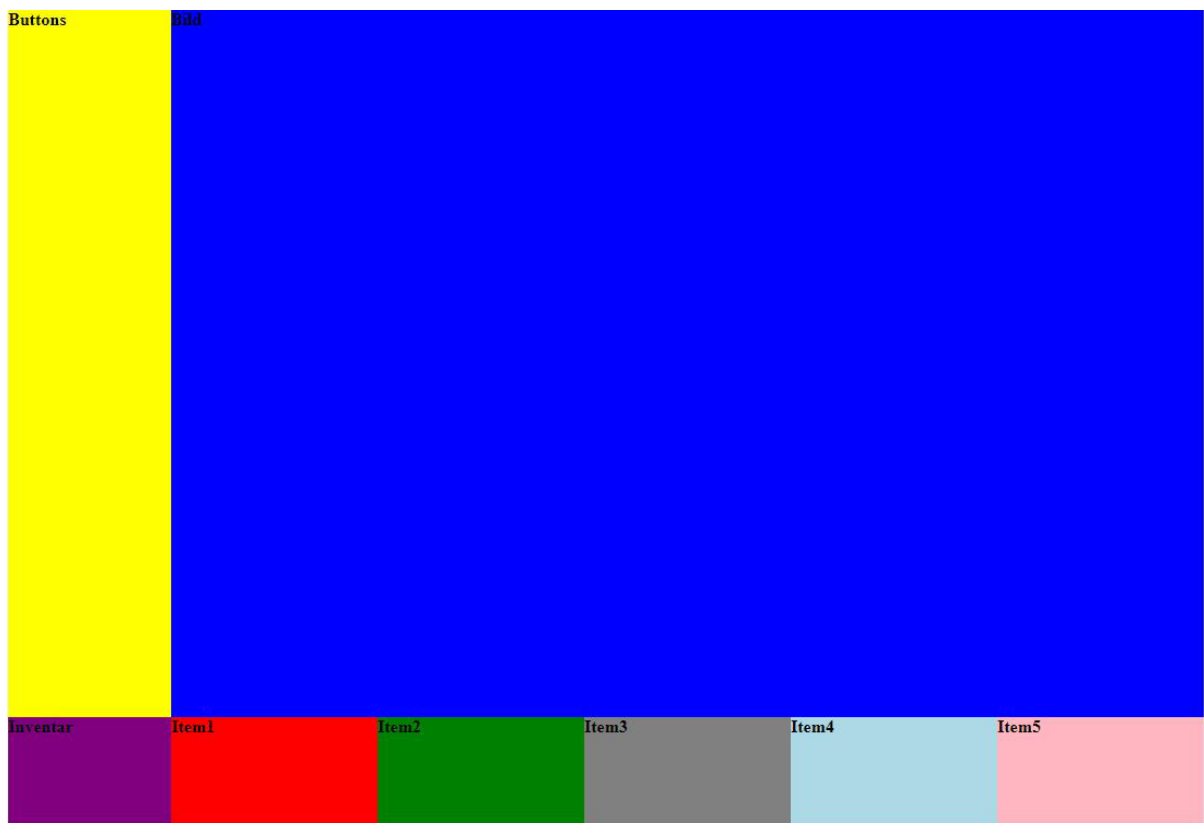


# Die Spielidee

Bei unserer Entscheidung über das Spielgenre kamen wir relativ schnell auf ein Point&Click-Adventure. Die überzeugendsten Gründe für dieses Genre waren die vermutete simple Programmierung und daher vermeintlich mehr Zeit für viele Interaktionen, Räume und Objekte. Wir suchten dann nach guten Vorbild-Spielen um einige Anhaltspunkte zu haben und kamen auf "Edna bricht aus". Der Plan war es am Anfang die wichtigsten Punkte zu sammeln und dann für unser Spiel abzuändern und anzupassen. Außerdem mussten wir uns auch entscheiden ob wir eher seriös oder lustig an die Idee rangehen sollten. Ein lustiger Ansatz erschien aber deutlich naheliegender. Ursprünglich hatten wir auch noch viele Ideen für lustige Räume und Items oder Rätsel, allerdings sind die meisten davon durch den Zeitdruck verloren gegangen.



*der erste Design Test:*



*das fertige Ergebnis*



# Die Handlung

## Story:

In unserem Spiel geht es um den kleinen (sehr kleinen) Jungen Bernd, der in den Sommerferien von seiner Mutter in seinem Zimmer eingeschlossen wurde, um ihn für sein zuletzt angerichtetes Chaos zu bestrafen. Bernd sieht es aber so gar nicht ein, nicht mit seinen Freunden draußen spielen zu dürfen und hat es sich zur Aufgabe gemacht aus seinem Zimmer zu entkommen. Mit Hilfe von verschiedenen sammelbaren Gegenständen und den Hinweisen durch Bernds Gedanken, ist es jetzt Aufgabe des Spielers, dessen Plan in die Tat umzusetzen.

## Interaktion:

Im Spiel kann der Spieler auf verschiedene Arten mit der Spielwelt interagieren. Zunächst können die meisten Objekte im Spiel angesehen werden um mehr über sie zu erfahren und Hinweise aus Bernds wirren Gedanken zu erschließen. Bernd geht dabei immer zu den Gegenständen hin um sie genau betrachten zu können. Man kann außerdem manche Dinge Benutzen, z.B. Schlüssel; dies ist eines der wichtigsten Mittel um im Spiel weiter zu kommen. Das Mitnehmen mancher Objekte bildet eine weitere Art von Spieler-Interaktion. Aufgehobene Items werden im Inventar am unteren Rand des Spielfelds hinzugefügt und können von dort mit anderen festen Gegenständen des Spiel benutzt also "kombiniert" werden.

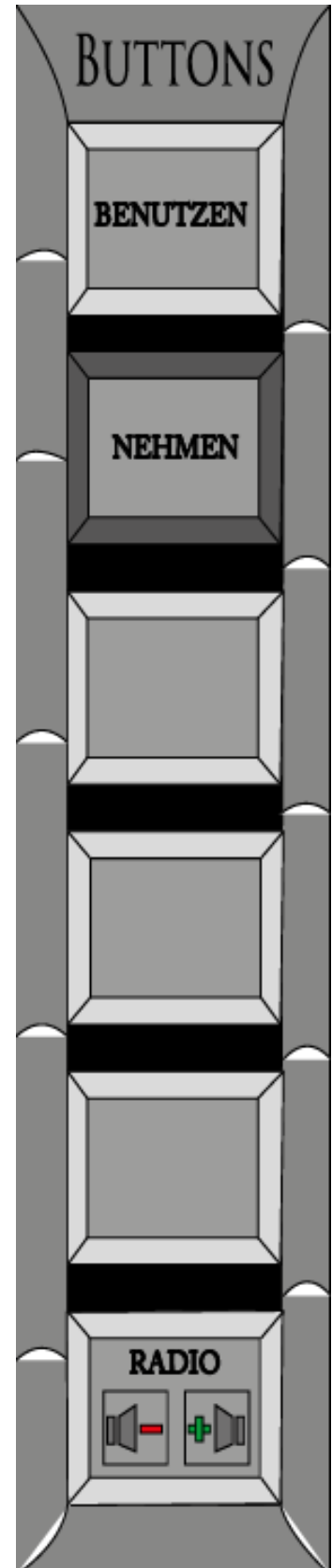
Damit die Orientierung im Raum leichter fällt und man nicht wahllos alles anklicken muss, ändert der Maus-Cursor beim Hover über ein wichtiges Objekt zur typischen Auswahl-Hand und in der oberen rechten Ecke erscheint eine Textbox mit Feedback, welcher Gegenstand gerade bei einem Klick ausgewählt werden würde (z.B. "Bett"). Das Benutzen und Mitnehmen wird über Drücken des jeweiligen Buttons am linken Bildschirmrand und danach das anklicken eine der wie oben beschriebenen markierten Flächen im Raum. Beide Buttons (Benutzen und Nehmen) bleiben "aktiv" bis die jeweils andere oder

# Die Handlung

## Interaktion:

auch dieselbe Taste geklickt wird.

Objekte im Raum können auch den Zustand der Tasten zurücksetzen, aber nur wenn die Interaktion vorgesehen bzw. für das Voranschreiten im Spiel sinnvoll ist. Sollte z.B. versucht werden einen Blumentopf mitzunehmen, wird dies durch Bernd kommentiert und der Nehmen Button bleibt aktiv. Um zu vermeiden dass unklar wird, welche Taste gerade ausgewählt und aktiv ist, ändert sich das Aussehen der Tasten. So bleibt immer ersichtlich in welchem Zustand von Interaktion der Spieler gerade ist. Schließlich kann im ersten Raum des Spiels ein Radio gefunden werden das durch klicken an- und ausgeschaltet werden kann. In der unteren linken Ecke des Interfaces kann man die Lautstärke des Radio-Songs lauter und leiser stellen. Standardmäßig läuft eine Hintergrundmusik welche durch den Radio-Song ersetzt wird, solange das Radio An ist.

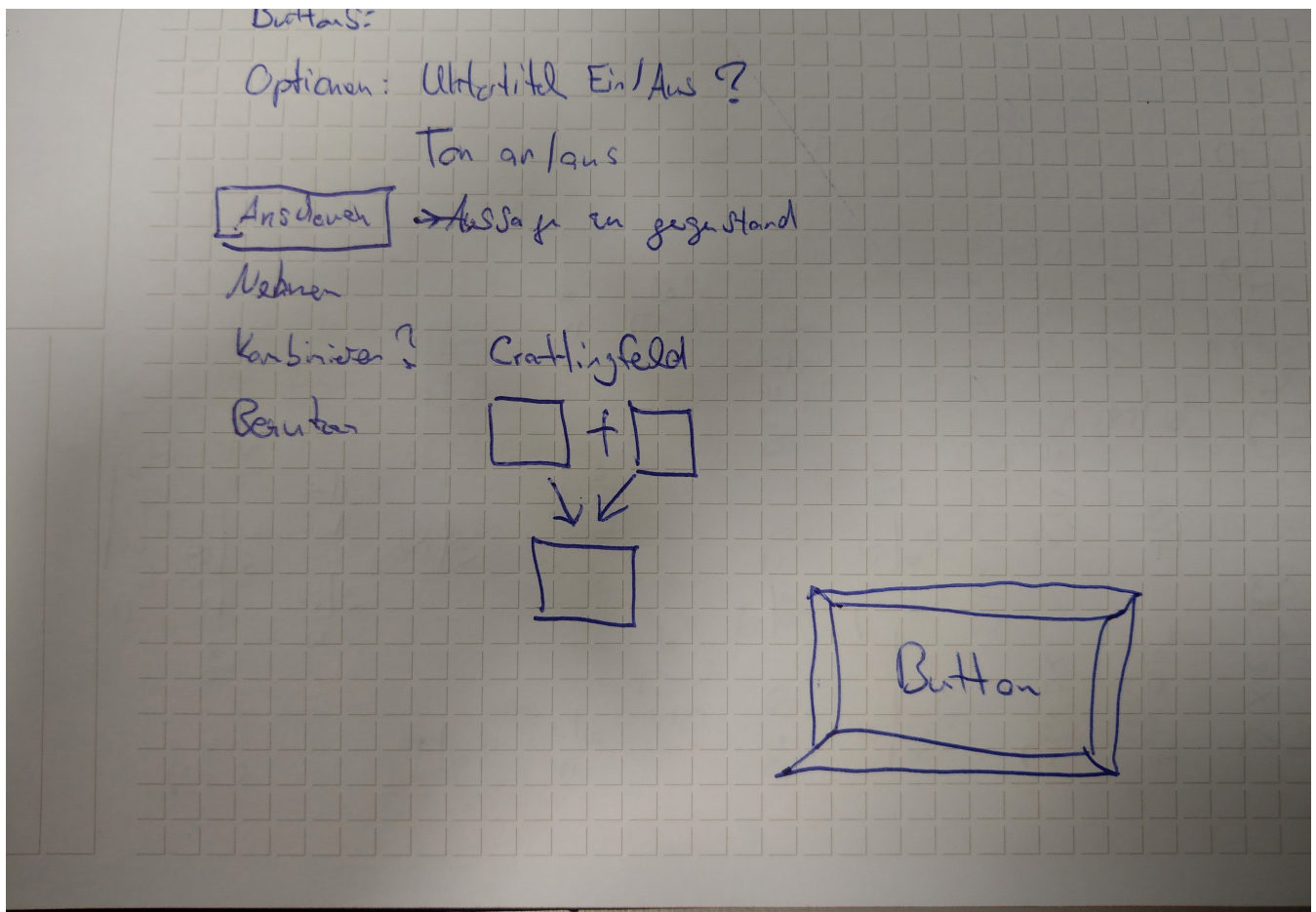


Der „Nehmen“-Button ist aktiviert.



# Die Graphische Gestaltung

Das Layout war eines der ersten besprochenen Elemente. Wir haben diskutiert wie viele und welche Buttons wir einbauen wollen, also eigentlich auch direkt welche Arten von Interaktion wir benutzen wollten. So war das Ansehen von Dingen auch mit eigener Taste in der Überlegung, auch eine "Kombinieren"-Funktion wäre denkbar gewesen.



Da auch ein Inventar immer sichtbar im Spiel bleiben sollte legten wir uns schon einmal auf 2 Kanten des Spielfelds fest an denen Inventar und Buttons angelegt werden würden (Unten und Links). Wir haben dann zur Vorschau ein Testbild erstellt um auch für die Längenverhältnisse ein Gefühl zu bekommen.

# Die Graphische Gestaltung

Insgesamt ist der Spielstil im Comic-Look gehalten. Da wir ja sowieso ein eher lustiges Spiel machen wollten und uns "Edna bricht aus" aus Vorbild genommen hatten, wäre ein realistischer Stil eher fehl am Platz gewesen. Auch die Methode die man in vielen Projekten aus den Vorjahren sehen konnte, nämlich Fotos aus Hochschule und Umgebung mit einem Filter als Hintergrund zu nehmen, sahen wir als unpassend für unser Spiel an. Es sollte alles eher einfach und übersichtlich bleiben und möglichst auf Gegenstände ohne Funktionen verzichtet werden. Zwar haben nicht alle Elemente der Räume einen Nutzen um im Spiel weiter zu kommen, aber sie können dennoch manchmal Hinweise liefern. Schließlich sollte der Raum nicht aus 5 Dingen bestehen die nur in einer bestimmten Reihenfolge gedrückt werden müssen, ohne einige "nutzlose" Deko-Objekte. Der Spielercharakter ist als Folge passend zu diesem Comic-Aussehen des Raumes entstanden, mit Inspiration durch unter anderem "South Park". Items sind, da sie ja oft aus dem Spielfeld eingesammelt werden im gleichen Aussehen wie auch die Statischen Objekte im Spiel designt.

# Die verwendeten Technologien

Bernd

Der Code wurde mit Visual Studio Code geschrieben und später über Github geteilt und verwaltet

## Der Code

Da unser Spiel ein Point&Click-Adventure ist, basiert der Hauptteil des Spiels auf Html-Buttons. Für alle Objekte im Bild, alle Items im Inventar, die Interaktions-Buttons und die Radiosteuerung wurden Button Objekte in Html erstellt. Wir haben uns dafür entschieden die Raumzustände für jede Veränderung einzeln zu speichern, was rückblickend gesehen eher umständlich war. Der Hauptteil der Logik besteht somit darin die verschiedenen Zustände zu unterscheiden in denen der Raum sein kann, je nachdem was der Spieler in welcher Reihenfolge tut. In Räumen mit vielen Optionen sorgt dies für ziemlich viele If-Abfragen um auch immer das richtige Raumbild anzeigen zu können. So kann im ersten Raum z.B. entweder zuerst eine Schublade geöffnet oder ein Hocker bewegt werden. Währenddessen kann aber jederzeit das Radio sowohl an als auch aus geschaltet sein, was die Anzahl der Raumzustände verdoppelte. Man könnte hier also schonmal als Verbesserungsidee für zukünftige Projekte festhalten, dass alle Objekte einzeln verändert werden sollten da so der Code kürzer und auch übersichtlicher würde. Die Sprechblasen sind auch nicht die eleganteste Lösung die möglich gewesen wäre. Das Konzept der Sprechblasen wurde bereits implementiert, bevor in den Vorlesungen Arrays oder Objektorientierte Programmierung erwähnt wur-



# Der Code

de. Beides hätte natürlich naheliegende Vorteile für eine Funktion welche im Grunde immer wieder das gleiche tut: Nach dem klicken eines Button-Objekts ein Bild anzeigen und nach 3 Sekunden wieder ausblenden. Die Time-Out Funktion war hier eine gute Ergänzung des ursprünglichen Codes. Auch das Ausblenden von alten Sprechblasen beim aktivieren eines anderen Gegenstands um zu mehrere gleichzeitige Sprechblasen zu vermeiden war eine sinnvolle Verbesserung. Der Aufwand für viele Sprechblasen bleibt jedoch unnötig hoch, da wir nicht im Nachhinein nicht auf z.B. Arrays umgestellt haben.

## Die Projektplanung und die Arbeitsteilung

Am Anfang war die Idee, mit dem Spiel so ziemlich bis zu Ende Dezember fertig zu werden um für andere Module mehr Freiraum zu bekommen. Dieses Ziel haben wir eindeutig nicht erfüllen können. Unter anderem lag das daran, dass wir dem ersten Raum zu ausführlich programmiert und gestaltet haben, die weiteren Räume des Spiels haben darunter leiden müssen. Da wir uns auf 4 Räume als gute Größe geeinigt hatten und ein Großteil der Zeit für das Schlafzimmer drauf ging werden die Räume stetig simpler im späteren Spielverlauf.

Die Arbeitsteilung sah wie folgt aus: Mario hat sich um den Großteil der Spiellogik also z.B. alle Buttons und die Zustandsabfragen gekümmert. Fillip hat sich um den Spielcharakter also alle Animationen und bewegungen im Raum gekümmert. Bernd hat alle gestalterischen Arbeiten übernommen, dementsprechend alle Bilder für Spielfeld, Items, Interface, Intro und die Interaktions-Buttons. Michael hat die Sprechblasen implementiert.

# Fazit: Fillip

Bei „Bernd bricht aus“ war Ich erst hauptsächlich für die Gestaltung für die Figur „Bernd“ verantwortlich, sowie für die Animationen der Figur und dem Movement in Spiel. Wo ich Bernds design für das erste Zimmer gesehen habe, ist mir direkt „South Park“ in den Sinn gekommen. Der Grundumriss der Figur Basiert auf den Hauptcharakteren dieser Show. Für sein Outfit habe ich mich für ein simples Design entschieden. Ein Rowdy mit einem Lächeln und einem Pflaster im Gesicht. Er trägt eine Cappy und einen Pullover. Seine Pose ist an „Sans“ von „Undertale“ angelehnt und die Laufanimationen an die alten Pokémon spiele.

Doch nach einiger Zeit habe ich die Figur nachbearbeitet, mit einer deutlicheren Farbe, einer Schramme statt einem Pflaster und Ovalen Augen. Die Augen sollen an Cartoons von den 20er 30er Jahren erinnern.

An den Animationen habe ich mich ganz am Anfang schwer getan, da ich im Internet viele verschiedene Möglichkeiten gefunden habe, wie man in CSS/Javascript eine Figur per Knopfdruck bewegen lassen kann. Nach einigen Verzweifeln habe ich die Methode gefunden, die ich für das Spiel verwendet habe. Diese Methode habe ich benutzt, da sie mich sehr an Keyframe Animationen von 3D Animation Programmen erinnert und ich darin schon Erfahrung habe. Ich habe mich dafür entschieden, jeweils eine bestimmte Anzahl an Punkten im Raum zu haben, zu welchen er hingehen kann, statt ein Punkt für jedes Objekt.

Welche „Geh-Animation“ er benutzt ist eine Pure Design-Wahl gewesen, obwohl dies auch mit einer Funktion berechenbar gewesen wäre. Außerdem war ich für die Funktionen der Sprechblasen im Badezimmer und in der Küche verantwortlich. Dort dachte ich mir, bevor ich mich in der bisherigen Funktion für die Sprechblasen verbeisse, schreibe ich eine neue Funktion für die letzten zwei Räume. Im Ganzen hat mir das Projekt gut gefallen, da das Programmieren und das Gestalten der Figur Spaß gemacht hat und ich gute Gruppenmitglieder hatte.

# Fazit: Bernd

Wir haben uns entschieden, ein Point & Click Adventure zu kreieren. Ich war für die Gestaltung an unserem Spiel zuständig. Um auf eine Idee zu kommen, habe ich das Spiel „Edna bricht aus“ als Mustervorlage genommen. Angefangen habe ich mit den Zimmern des Spiels: Schlafzimmer, Flur, Badezimmer, Küche und das „End-Bild“. Die habe ich mit Hilfe von Adobe Illustrator mit den „Perspektivenraster“ erstellen können. Des Weiteren habe ich die Buttons, das Inventar, die Items und die Sprechblasen entworfen.

Ich bin sehr zufrieden mit unserer Gruppe, denn ich konnte mich in der Gestaltung frei entfalten. Zusätzlich fand ich es gut, dass wir unsere Aufgaben aufgeteilt haben. Jeder hat an seinen Aufgaben gearbeitet und wir haben uns gegenseitig helfen können. Wir haben uns von Woche zu Woche Ziele gesetzt. Diese Ziele konnten wir leider nur teilweise erreichen.

Aufgrund der Verteilung der Aufgaben habe ich die meiste Zeit dafür verwendet das Spiel zu Illustrieren. Deshalb habe ich nur sehr wenig an der Programmierung mitarbeiten können. Am Ende könnte ich mir größtenteils nur Code anschauen und nicht selbst schreiben. Dafür habe ich wiederum in Gestaltung viel dazu lernen können.

Außerdem war am Ende unser Zeitmanagement schlecht. Wir haben weniger Zeit investiert und dadurch wurde es noch knapp. Deshalb haben wir nicht alles schaffen können, was geplant war.

Im Großen und Ganzen bin ich zufrieden mit unserem Projekt. Jeder hat sein bestes gegeben und es gab keine Streitigkeiten.

# Fazit: Michael

In unserem Projekt war ich für den Großteil der Sprechblasen-Implementierung verantwortlich. Mario hat die erste Form der Funktion erdacht und ich habe dann weitere Features und die Großzahl der Sprechblasen hinzugefügt. Um ehrlich zu sein war das normale Einfügen von Sprechblasen ohne Nutzung von Arrays unnötig Eintönig und meist nur stumpfes Copy&Paste. So hat jetzt jede Sprechblase mehrere Variablen und Funktionen, die sich alle nur im Namen minimal unterscheiden. Das Ausdenken und Umsetzen von neuen Funktionen für die Sprechblasen und auch mein Radio-Code, waren dagegen aber sehr angenehm zu programmieren. Beides stellte mich vor einige Probleme, die ich dann nach und nach abgebaut habe. Als ich einmal auf die Lösung eines dieser Probleme gekommen war, z.B. wie vermieden wird mehrere Sprechblasen gleichzeitig anzuzeigen, stellte das eigentlich den besten Teil des Projekts dar. Ich hätte rückblickend auf jeden Fall mehr Zeit in ein besseres Konzept für die Logik stecken sollen und weniger in das Kopieren von sehr ähnlichem Code. Zeitmanagement war generell etwas, dass in einem späteren Projekt verbessert werden kann. Wir haben uns auf jeden Fall in unserer Geschwindigkeit überschätzt. Die Arbeit in der Gruppe hat viel Spaß gemacht und gestaltete sich außerdem als effizient. Alle haben ihren Beitrag geleistet und die Kommunikation hat super funktioniert (was ja längst nicht Selbstverständlich ist). Somit habe ich einiges während des Projekts lernen können und hatte insgesamt viel Spaß. 10/10 would programm again.

# Fazit: Mario

Ich habe im Projekt die Logik des Spiels implementiert. Damit sind die verschiedenen Bildwechsel im Spiel gemeint. Wir haben uns dazu entschieden mit Hauptbildern zu arbeiten und nicht mit Objekten die sich im Spiel bewegen. Aufgrund der mangelnden Vorerfahrung mit der Programmierung, habe ich gedacht, dass dies der einfachste Weg ist. Allerdings hat sich herausgestellt, dass es mit einzelnen Objekt deutlich einfacher gewesen wäre. Um die Problematik der Bildwechsel zu meistern, habe ich verschiedene Variablen angelegt, in denen ich die Spielzustände speichere. Im Spielverlauf wird dann auf diese Variablen geprüft, um festzustellen welches neue Bild erscheinen soll. Dadurch sind stellenweise sehr große If-Abfragen entstanden, da in einen Raum bis zu 10 verschiedene Zustände gespeichert werden. Als ich den ersten Raum fertig hatte, viel es mir deutlich einfacher die Logik für die nächsten Räume zu gestalten. Dabei ist mir aufgefallen, dass eine objektorientierte Programmierung die Programmierung deutlich vereinfacht hätte. Jedoch habe ich mich dagegen entschieden, da ich nicht mitten im Projekt anfangen wollte das Programmierkonzept zu ändern. Hätte ich mehr Zeit gehabt, hätte ich in Erwägung gezogen den kompletten Code erneut und sauberer zu schreiben. Ich bin trotzdem sehr zufrieden mit dem Projekt. Mir gehen mittlerweile Arbeiten mit HTML, CSS und JavaScript viel leichter von der Hand, was mir auch in den anderen Modulen zur Hilfe kommt. Gerade in Hinsicht auf die Website die wir im Modul Gestaltung erstellen sollen, habe ich gemerkt, dass es deutlich schneller geht. Außerdem habe ich aus meinen Fehlern gelernt und viel mehr mit Klassen und deren Vorteilen gearbeitet. Mir hat die Programmierung am Spiel viel Spaß bereitet und freue mich schon auf weitere Projekte. Gerade die Lösung von Problemen oder das beheben von Bugs ist ein sehr erfüllendes Gefühl. Durch die Semesterarbeit fiel mir der Einstieg in die Webentwicklung sehr einfach, da ich das Thema sehr spannend fand und wir die Möglichkeit hatten uns frei zu entfalten. Abschließend kann ich sagen, dass ich sehr viel gelernt habe und ich großen Spaß an dem Projekt hatte. Auch die Zusammenarbeit in der Gruppe hat super geklappt und die Rollenverteilung der Gruppe hat gut funktioniert. Die Kommunikation in der Gruppe hat aus meiner Sicht wunderbar funktioniert und ich hoffe das wir auch in Zukunft weiterhin zusammenarbeiten können.