Bernd bricht aus

ein Projekt von

Fillip Czogallik

Bernd Henke

Mario Schulz

Michael Kettenburg

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.

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**

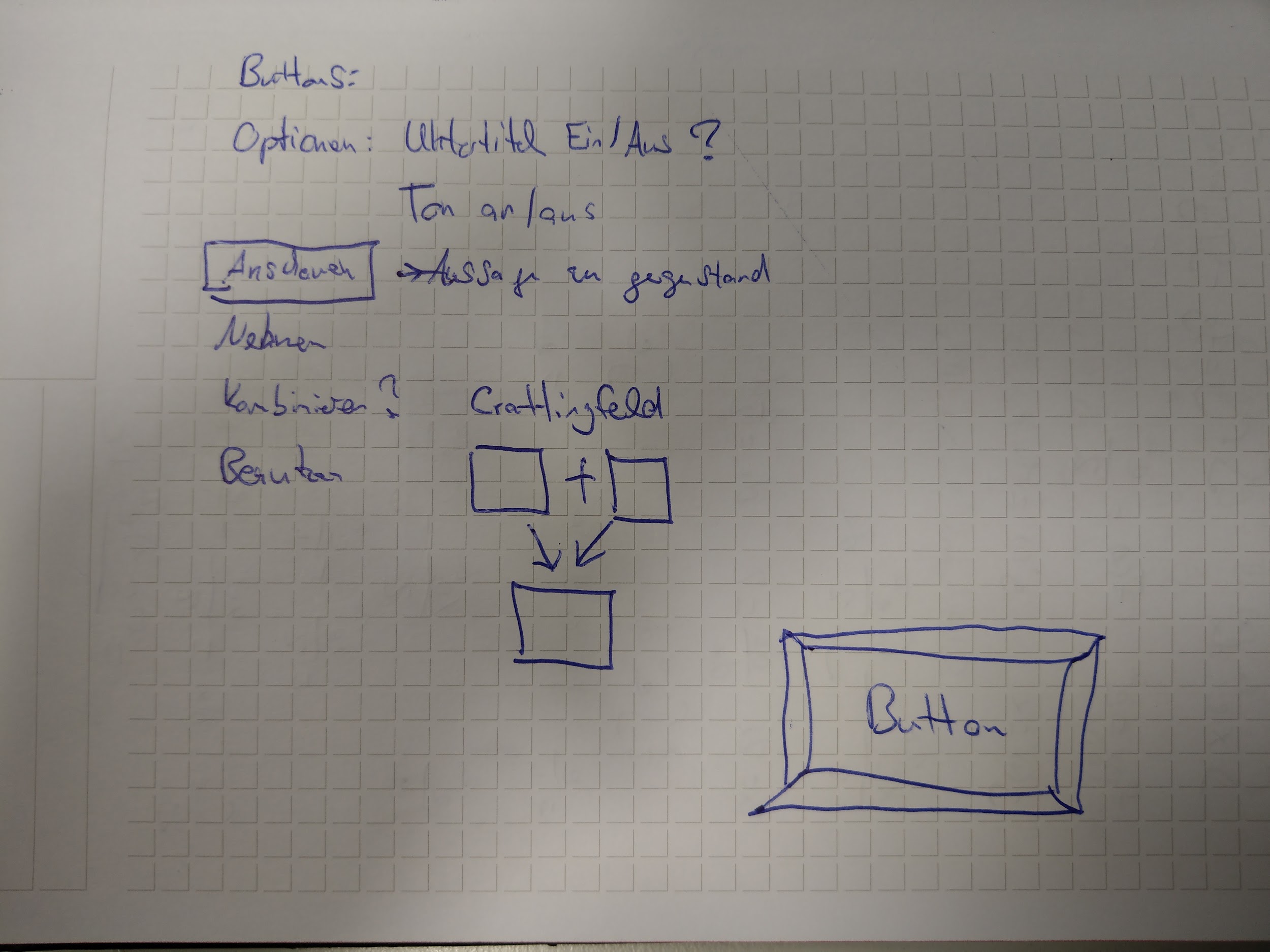
Im Spiel kann der Spieler auf verschiedene Arten mit der Spielwelt interagieren. Zunächst können die meisten Objekte im Spiel angesehen werden um mehr über sie zu erfahren und Hinweise aus Bernds wirren Gedanken zu erschließen. Bernd geht dabei immer zu den Gegenständen hin um sie genau betrachten zu 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.

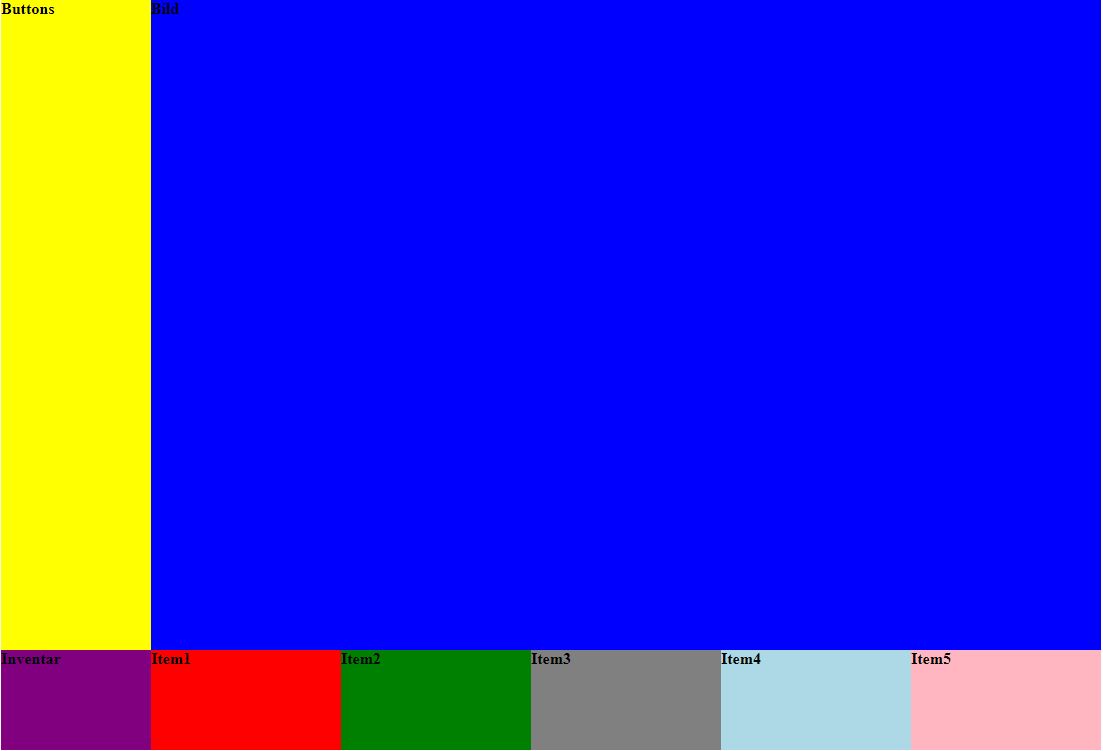
Damit die Orientierung im Raum leichter fällt und man nicht wahllos alles anklicken muss, ändert der Maus-Cursor beim Hover über ein wichtiges Objekt zur typischen Auswahl-Hand und in der oberen rechten Ecke erscheint eine Textbox mit Feedback, welcher Gegenstand gerade bei einem Klick ausgewählt werden 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 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.

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.



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.

Verwendete Technologie

Bernd

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

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 wurde. 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.

Projektplanung und 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.

Mögliche Bilder

