Studie

| Auftraggeber | Andres Scheidegger |
| --- | --- |
| Projektleiter | Michael Günter, Elias Schmidhalter |
| Autor | Michael Günter, Elias Schmidhalter |
| Klassifizierung | Intern |
| Status | Freigegeben |
|  |  |

Änderungsverzeichnis

| Datum | Version | Änderung | Autor |
| --- | --- | --- | --- |
| 03.02.2015 | 0.1 | Erstfassung | Michael Günter |

Inhaltsverzeichnis

1 Situationsanalyse 3

1.1 Ausgangslage 3

1.2 Stärken 4

1.3 Schwächen 4

2 Ziele 6

3 Liste der Stakeholder 7

4 Anforderungen 9

5 Lösungsvarianten 11

5.1 Variantenübersicht 11

5.2 Beschreibung der Varianten 11

5.2.1 Variante A: Mobile Lösung mit normalen Elementen 11

5.2.2 Variante B: Mobile Lösung mit chemischen Elementen 12

5.2.3 Variante C: Native mobile Lösung 13

6 Bewertung der Varianten (Tabelle) 14

7 Lösungsbeschreibung 15

7.1.1 Spielprinzip 15

7.1.2 Benutzeroberfläche 16

7.1.3 Anforderungsabdeckung 16

8 Projektplanung 17

8.1.1 Beschreibung der Arbeitspakete 17

9 Empfehlung 19

10 Projektfreigabe 19

# Situationsanalyse

## Ausgangslage

Es ist vorgesehen, dass wir, wie in dem Initialisierungsauftrag geschildert, ein Spiel entwickeln, in welchem man verschiedene Elemente oder Gegenstände kombinieren und splitten kann, um neue Elemente zu schaffen. Mit den neuen Elementen lassen sich wiederum durch dieselben Prinzipien neue, andere Elemente erschaffen. Das grundsätzliche Ziel eines solchen Spiels, ist es den Benutzer zu unterhalten.

Momentan gibt es schon einige Spiele, welche dieses Prinzip nutzen. Sowohl im Browser als auch im mobilen Bereich.

* Little Alchemy, für Google Chrome, Android und iOS.
* Alchemie, für Android und iOS



Diese Spiele haben über die Zeit sehr viele neue Elemente erhalten. Bei Little Alchemy sind es beispielsweise, momentan ca. 550 verschiedene Elemente. Dies wird nicht unser Ziel sein. Wir werden uns auf ein erweitertes Spielprinzip konzentrieren, welches auch vorerst mit ca. 30 – 50 Elementen auskommt. Unser Spiel soll später auch in die App-Stores kommen, um möglichst viele Benutzer anzusprechen.

## Stärken

Die Stärken der aktuell existierenden Spiele sind vor allem die unglaubliche Vielfalt an Elementen. Ausserdem macht es Spass, herauszufinden, welche Elemente kombiniert werden können. Auch verschiedene Lustige Aspekte wurden beachtetet, indem zum Beispiel bei der Kombination bestimmter Elemente, etwas Unerwartetes, Lustiges herauskam.

Da wir uns um ein etwas abgeändertes Konzept bemühen werden, ist es nicht von Nöten, dass wir eine immense Vielfalt an Elementen implementieren. Viel wichtiger ist jedoch, dass es dem Spieler nicht langweilig wird, und wir auch ähnlich wie in den bestehenden Spielen, einige lustige Elemente einbringen.

Zusammenfassung:

1. Vielfalt der Elemente
2. Spass beim Kombinieren
3. Lustige Aspekte

## Schwächen

Die grösste Schwäche an der heutigen Situation ist eigentlich nur eine einzige. Die heute existenten Spiele berufen sich allesamt auf dasselbe Spielprinzip. Somit gibt es viele Spiele, die zwar auf den ersten Blick einzigartig erscheinen, jedoch sich nachher als Kopie eines anderen Spiels, lediglich mit etwas abgeänderten Elementen, entpuppt.

In unserem Projekt werden wir genau diesen Schwachpunkt verstärkt behandeln. Wir wollen ein neues, überarbeitetes Konzept einbringen. In den aktuellen Spielen ist es nämlich so, dass man einen Elemente-Vorrat hat, welcher unbegrenzt ist. Sobald man ein Element entdeckt hat, ist es in unbegrenzter Masse verfügbar.

Dies wollen wir mit Unsustainable Alchemy ändern. Wir wollen, dass der Spieler immer gleich viele Elemente auf dem Spielfeld hat, und nur diese zum Kombinieren zur Verfügung hat. Ausserdem soll auch das Splitten von Elementen möglich sein, damit dem Spieler ermöglicht wird, in der Element-Hierarchie hoch und runter zu navigieren.

Ausserdem versuchen wir verschiedene Erfolge einzubauen, um auch zwischendurch, beim Erlangen bestimmter Elemente, dem Benutzer das Gefühl zu übermitteln, erfolgreich zu sein.

Zusammenfassung:

1. Alle Spiele basieren auf demselben Konzept
2. In manchen Spielen, ist eine gute Bedienbarkeit nicht gewährleistet.
3. Es gibt keine Möglichkeit, zu verlieren.
4. Man hat erst Erfolg, wenn man alle Elemente gefunden hat.

# Ziele

Der folgende Abschnitt stellt eine Übersicht über alle Ziele dar, die wir während des gesamten Projekts erreichen möchten.

**Schwächen bezogene Ziele**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ziel** | **Schwäche** | **Beschreibung** | **Erreichung** | **Kategorie** |
| **1** | 1, 3, 4 | Wir wollen, während der Konzeptionsphase, ein neues, überarbeitetes Spielprinzip, welches sich von den bisherigen Prinzipien unterscheidet, erarbeiten, und später in der Realisationsphase implementieren. | Das Ziel ist erreicht, sobald wir das Spielprinzip implementiert haben. | Muss |
| **2** | 2 | Wir wollen eine übersichtliche Bedienbarkeit gewährleisten, welche besser als bei manchen anderen Spielen ist. Diese wollen wir während der Konzeptionsphase erarbeiten und während der Realisation implementieren. | Das Ziel ist erreicht, sobald wir die Benutzeroberfläche benutzerfreundlich implementiert haben. | Kann |

**Andere Ziele**

|  |  |  |  |
| --- | --- | --- | --- |
| **Ziel** | **Beschreibung** | **Erreichung** | **Kategorie** |
| **3** | Wir wollen dem Spieler primär ein unterhaltsames Erlebnis liefern, während er unser Spiel spielt. | Das Ziel ist erreicht, wenn das Spiel unterhaltsam ist. | Kann |
| **4** | Wir wollen während der Realisierung unser Wissen im Bereich App- und Spielentwicklung erweitern. | Das Ziel ist nach der Realisierungsphase erreicht. | Muss |
| **5** | Wir wollen, während wir die die Grafiken für das Spiel erstellen, erlernen, wie man Pixelgrafiken schön aussehen lässt. Dabei sollen schöne Pixelgrafiken erstellt werden. | Das Ziel ist erreicht, wenn wir schöne Pixelgrafiken erstellen können. | Kann |
| **6** | Das Spiel soll am Ende der Realisierung auf den gängigsten Mobilplattformen (Android und iOS) lauffähig sein. | Das Ziel ist erreicht, sobald die App auf beiden Plattformen läuft. | Muss |
| **7** | Das Spiel soll am Ende der Realisierung flüssig laufen und nicht ruckeln. | Das Ziel ist erreicht, sobald das Spiel flüssig läuft. | Kann |
| **8** | Wir wollen am 17.02.2015 mit der Konzeptionsphase beginnen, welche am 3.3.2015 abgeschlossen werden soll, damit wir danach mit der Realisationsphase beginnen können. | Das Ziel ist erreicht, wenn der Zeitrahmen eingehalten werden kann. | Muss |
| **9** | Das Spiel wird am Ende in den Stores veröffentlicht. | Das Spiel ist veröffentlicht. | Kann |

**Rahmenbedingungen**

Wir werden am 17. Februar mit der Konzeptionsphase des Projekts beginnen. Diese wollen wir, während ca. 3 Wochen beenden. Sobald diese Phase beendet ist, gilt es die Software zu realisieren. Für die Konzeptionsphase werden wir keine zusätzlichen Aufwände ausserhalb der Schulzeit verbuchen müssen, da die Zeit, die wir zur Verfügung gestellt bekommen vollkommen ausreicht. Die meiste Zeit werden wir uns also in der Gibb mit dem Projekt beschäftigen. Während der Realisierungsphase wird es von Nöten sein, ausserschulische Aufwände zu betreiben, welche meist Zuhause oder Unterwegs vollführt werden. Die Kommunikation und Organisation erfolgt dabei auf den heute gängigen Kommunikationsmitteln. Source Code und Dokumente werden auf GitHub verwaltet und zur Verfügung gestellt.

**Abgrenzung**

Wir wollen unser Projekt vorerst nicht zu Umfangreich gestalten. Das bedeutet, dass wir uns vorerst auf ca. 30 verfügbare Elemente beschränken werden. Wir wollen mit dem Projekt nicht unsere Grafikkünste ins Rampenlicht stellen, sondern uns geht es eher um das ausgefeilte Spielprinzip. Unser Vorhaben soll vorerst eine Basis für ein gutes Spiel bieten, welches man später mit zusätzlichen Elementen erweitern kann.

# Liste der Stakeholder

* Andres Scheidegger, Auftraggeber
* Elias Schmidhalter, Entwickler und Projektleiter
* Michael Günter, Entwickler und Projektleiter
* Kunden aus den App-Stores

# Anforderungen

|  |  |  |
| --- | --- | --- |
| **Anforderung** | **Ziel** | **Beschreibung** |
| **1 (Neues Konzept)** | 1 | Als Spieler will ich mit einem neuen Konzept überrascht werden, um neue Dinge auszuprobieren. |
| **2 (Kombinieren)** | 1 | Als Spieler will ich Elemente kombinieren und teilen, um neue Elemente zu erschaffen. |
| **3 (Glücksfaktor)** | 1 | Als Spieler erwarte ich einen gewissen Glücksfaktor, damit nicht alles von Strategie und Nachdenken abhängt. |
| **4 (Misserfolg)** | 1 | Als Spieler will ich, dass ich verliere wenn die Energiebar keine Energie mehr enthält. Damit ich ein Gefühl des Misserfolgs erhalte. |
| **5 (Erfolge)** | 1 | Als Spieler will ich gelegentlich mit Erfolgen belohnt werden, um Motivation, Spass und ein Erfolgsgefühl zu erlangen. |
| **6 (UI)** | 2 | Als Spieler will ich ein übersichtliches, gut aussehendes, Benutzerinterface bedienen, damit ich einen professionellen Eindruck des Spiels erhalte. |
| **7 (Elemente Übersicht)** | 2 | Als Spieler will ich eine Übersicht über die erhaltenen Elemente. Die Übersicht soll Informationen über die einzelnen Elemente enthalten, damit der Spieler immer informiert über den Fortschritt ist. |
| **8 (Erfolgs  Übersicht)** | 2 | Als Spieler will ich eine Übersicht über die erzielten Erfolge. Die Übersicht soll Informationen zu erhaltenen Erfolgen und Anweisungen zu freischaltbaren Erfolgen enthalten, damit der Spieler weiss, was es als nächstes zu holen gibt. |
| **9 (Unterhaltung)** | 3 | Als Spieler will ich von dem Spiel unterhalten werden, um die Langeweile zu eliminieren. |
| **10 (Motivation)** | 3 | Als Spieler will ich lustige Aspekte in dem Spiel vorfinden, damit ich motiviert werde, noch mehr lustige Elemente zu finden. |
| **11 (Weiterbildung)** | 4 | Die Entwickler entwickeln ein mobiles Spiel, um Wissen im Bereich App- und Spielentwicklung zu erweitern. |
| **12 (Grafiken)** | 5 | Die Entwickler geben sich Mühe, schöne Pixelgrafiken für die Elemente zu erstellen, damit ein angemessenes Spielerlebnis für den Spieler ermöglicht wird. |
| **13 (Android/iOS)** | 6 | Als Spieler will ich das Spiel auf Android und iOS spielen können, um eine optimale Kompatibilität auf allen Plattformen abzudecken. |
| **14 (Performance)** | 7 | Als Spieler will ich, dass das Spiel flüssig und performant läuft, damit es nicht zu unerwünschten Unterbrechungen kommt. |
| **15 (Zeitplan)** | 8 | Als Projektleiter wollen wir, dass wir die geplante Zeit für die Konzeptionsphase einhalten können, damit wir pünktlich mit der Realisation beginnen können. |

# Lösungsvarianten

## Variantenübersicht

* Variante A: Mobile Lösung mit normalen Elementen
* Variante B: Mobile Lösung mit chemischen Elementen
* Variante C: Native mobile Lösung

## Beschreibung der Varianten

### Variante A: Mobile Lösung mit normalen Elementen

Die Variante A soll mit Apache Cordova umgesetzt werden, und Webtechnologien für die Darstellung verwenden. Durch die Webtechnologien ist man etwas eingeschränkter, als mit einer nativen Lösung, die Entwicklung geht jedoch schneller und effizienter voran.

Die Lösung soll ca. 30 Elemente beinhalten, welche bei bestimmten Elementkombinationen auch etwas Lustiges ergeben. Es werden keine chemischen Elemente, sondern eher Elemente aus dem Alltag verwendet.

Der Benutzer hat die Möglichkeit verschiedene Erfolge zu erreichen, und diese in einem separaten User-Interface anzuzeigen.

**Vorteile**

* Schnelle Entwicklung durch Webtechnologien
* Gleicher Code für alle Plattformen
* Lustige Aspekte, durch Kombinationen vorhanden

**Nachteile**

* Eingeschränkte Möglichkeiten durch Webtechnologien

### Variante B: Mobile Lösung mit chemischen Elementen

Die Variante B soll ähnlich wie die Variante A aufgebaut werden.

Jedoch sollen anstelle der normalen Elemente, chemische Elemente verwendet werden. Dadurch gehen jedoch möglicherweise die lustigen Aspekte der Kombinationen verloren. Man könnte die Elemente statt in einer klassischen Liste, in einem Periodensystem auflisten.

**Vorteile**

* Schnelle Entwicklung durch Webtechnologien
* Gleicher Code für alle Plattformen

**Nachteile**

* Lustige Aspekte sind nicht vorhanden
* Eingeschränkte Möglichkeiten durch Webtechnologien

### Variante C: Native mobile Lösung

Eine Native Applikation ist eine, welche direkt für ein Gerät geschrieben wurde. Dies wird normalerweise in einer vom Hersteller bereitgestellten Programmiersprache gemacht.

Eine Native Applikation ist sehr vorteilhaft, wenn man oft auf Systemfunktionen wie zum Beispiel die Kamera zugreift. Ausserdem ist die native Programmierung sehr performant und gut integriert. Die Hersteller legen viel Wert darauf, dass es gute und detaillierte Guidelines und vorgefertigte Bausteine gibt. Der grosse Nachteil dieser Variante ist, dass für jede Plattform, also in diesem Fall iOS und Android, die ganze App neu entwickelt werden muss.

**Vorteile**

* Alle Möglichkeiten durch native Entwicklung
* Performant

**Nachteile**

* Plattformabhängige Entwicklung in verschiedenen Sprachen
* Erhöhter Zeitaufwand

# Bewertung der Varianten (Tabelle)

Anhand folgender Tabelle werden die verschiedenen Lösungsvarianten miteinander verglichen. Sie dient als Entscheidungsgrundlage für die Wahl der Variante.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Variante** | **Zeitaufwand (2x)** | **Unterhaltung (1x)** | **Realisierbarkeit (2x)** | **Score** |
| Variante A | 8 | 9 | 8 | **41** |
| Variante B | 7 | 4 | 8 | 34 |
| Variante C | 4 | 9 | 4 | 25 |

Die Tabelle bewertet die 3 Lösungsvarianten nach ihrem Zeitaufwand, der Unterhaltung des Endprodukts und der Realisierbarkeit des Projekts.

Je höher der Wert für den Zeitaufwand desto weniger Zeit nimmt die Realisierung in Anspruch. Je höher der Wert für die Unterhaltung desto mehr wird der Benutzer des Spiels unterhalten. Je höher der Wert für die Realisierbarkeit desto einfacher oder besser ist es realisierbar.

Wir treffen anhand des Gesamt-Scores unsere Entscheidung auf die Variante A, weil sie den niedrigsten Zeitaufwand bietet, den grössten Unterhaltungsfaktor sowie, durch die Webtechnologien einfach und schnell realisierbar ist. Der Unterhaltungsfaktor ist bei Variante A grösser als bei Variante B, weil es mehr Spass macht, Elemente die man kennt zu kombinieren, anstatt völlig unbekannte chemische Elemente zu kombinieren.

# Lösungsbeschreibung

Unsere Lösung wird mit Apache Cordova, das heisst Javascript, HTML und CSS, umgesetzt. Apache Cordova ist ein Framework, welches die Entwicklung von mobilen Anwendungen mit Webtechnologien ermöglicht und Schnittstellen zu Anbindungen (bsp. Kamera) bereitstellt. Wir haben uns dazu entschieden, weil es eine schnelle und effiziente Möglichkeit bietet, die Lösung zu realisieren.

### Spielprinzip

Wir berufen uns auf ein bereits vorhandenes Spielprinzip und ändern dieses ab, um eine gewisse Frische des Spiels zu gewährleisten.

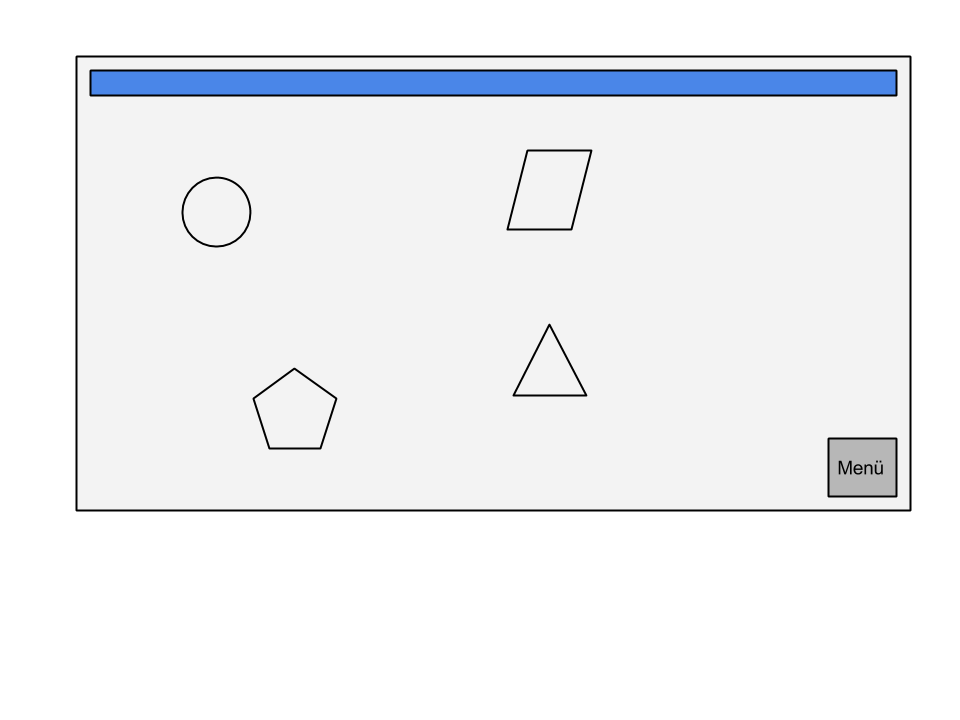
Grundsätzlich stehen dem Spieler am Anfang eine bestimmte Anzahl (ca. 5 oder 6) Elemente zur Verfügung. Diese werden als Grundelemente bezeichnet. Die Grundelemente lassen sich durch andere Grundelemente austauschen. So kann man durch die verschiedenen Grundelemente durchwechseln.

Alle Elemente lassen sich kombinieren. Aus zwei Elementen entstehen 2 andere, jedoch gleiche Elemente. Das Resultat der Kombination kann manchmal logisch, manchmal aber auch lustig sein. Diese beiden Elemente lassen sich nicht mehr auswechseln. Es lassen sich lediglich die Grundelemente auswechseln. Sie lassen sich jedoch stattdessen aufteilen. Durch die Aufteilung wird möglicherweise zufällig gewählt welches der Eltern-Elemente durch die Aufteilung entsteht. Man kann durch das Aufteilen jedoch nur die Eltern-Elemente erhalten.

Es existiert eine zusätzliche Energiebar. Diese verliert bei verschiedenen Aktionen Energie. (Welche genau, müssen wir noch in der Konzeptionsphase festlegen). Wenn die Energie aufgebraucht ist, ist das Spiel zu Ende und man muss wieder mit den Grundelementen anfangen.

Das Ziel des Spiels ist es durch diese Möglichkeiten, so viele Elemente wie möglich zu finden. Jedes gefundene Element wird in einer Liste eingetragen und detaillierte Informationen über das Element stehen zur Verfügung. Beim Erhalt bestimmter Elemente werden Erfolge freigeschaltet.

### Benutzeroberfläche



Grundsätzlich kombiniert der Spieler die verschiedenen Elemente durch ziehen und verschieben, der auf der Fläche herumliegenden Elemente. Werden zwei Elemente übereinander losgelassen, so kombinieren sie sich. Werden sie lange berührt, so splitten sie sich. Oben soll die Energiebar eingeblendet werden.

Durch den Menü Button soll der Benutzer eine Übersicht über die erhaltenen Elemente erhalten und die Erfolge einsehen können.

### Anforderungsabdeckung

|  |  |
| --- | --- |
| **Anforderung** | **Erfüllung** |
| **1 (Neues Konzept)** | Ja |
| **2 (Kombinieren)** | Ja |
| **3 (Glücksfaktor)** | Ja |
| **4 (Misserfolg)** | Ja |
| **5 (Erfolge)** | Ja |
| **6 (UI)** | Ja |
| **7 (Elemente Übersicht)** | Ja |
| **8 (Erfolgs Übersicht)** | Ja |
| **9 (Unterhaltung)** | Ja |
| **10 (Motivation)** | Ja |
| **11 (Weiterbildung)** | Ja |
| **12 (Grafiken)** | Kann |
| **13 (Android/iOS)** | Ja |
| **14 (Performance)** | Kann |
| **15 (Zeitplan)** | Ja |

# Projektplanung

Der Zeitplan und eine Übersicht sind in dem separaten Gantt-Diagramm ersichtlich. Bitte dazu das Excel-Dokument öffnen.

### Beschreibung der Arbeitspakete

*Konzeptionsphase*

**Definition Spiellogik**

In diesem Arbeitspaket wird die Spiellogik genauestens und möglichst detailliert definiert. Es gilt mögliche Schwächen an dem bisherigen Konzept zu identifizieren und auszubauen. Es gilt ein frisches, neues Spielprinzip und Möglichkeiten, diese zu implementieren, zu finden.

**UI Design**

Alle Benutzerinterfaces müssen designt werden und mögliche Steuerungsmöglichkeiten, die das UI betreffen müssen gefunden werden. Vor allem das Haupt UI, wo sich die Elemente befinden, als auch das Menü, wo die verschiedenen Auflistungen angezeigt werden können, müssen definiert und skizziert werden.

**Daten Konzept**

Es muss definiert werden, in welcher Form die Daten abgespeichert werden sollen. Soll eine Datenbank verwendet werden, und wenn ja welche Entitäten werden benötigt?

**Elemente und Kombinationen definieren**

Es muss eine detaillierte hierarchische Struktur von Elementen definiert werden. Ausserdem sollte eine Liste von Elementen angefertigt werden, mit zugehöriger Beschreibung. Die Kombinationen sollten aus der hierarchischen Struktur heraus ersichtlich sein. Möglicherweise gibt es auch eine übersichtlichere Variante der Darstellung. Am Ende sollte man eine Auswahl an ca. 30 Elementen haben, die wir dann auch in die Datenbanktabelle einfügen können.

**Technologie Auswahl**

Es muss eine Technologie bestimmt werden und genau beschrieben werden, welche Frameworks verwendet werden. Es muss auch definiert werden welches Datenbankmanagementsystem verwendet werden kann.

**Game Grundstruktur**

Die Grundstruktur bzw. das Projekt sollte angelegt werden und eine Dateistruktur sollte existieren. Ausserdem kann man definieren wo man künftig die Eventbasierte Spiellogik ablegen möchte.

**UI Grundstruktur**

Hängt fest mit der Game Grundstruktur zusammen. Die grundlegenden Masken und Ansichten der Software sollten stehen, und grundlegende Aktionen sollten von der Gamelogik her steuerbar sein.

**Daten Grundstruktur**

Die verschiedenen Entitäten sollten in der Datenbank existieren und einige Testdaten, möglicherweise auch schon die Daten für die Elemente und Elementkombinationen enthalten. Der Datenzugriff von der Gamelogik her sollte gewährleistet sein und eine einfache Abfrage sollte auf dem UI ausgegeben werden können. Ein Datenbank-Diagramm wurde erstellt.

*Realisierungsphase*

**Gamelogik: Kombinieren von Elementen**

Die Logik sowie die UI Aktionen für das Kombinieren von Elementen müssen umgesetzt werden. Ein Element muss per Drag & Drop verschoben werden können und sich mit einem anderen Element kombinieren, wenn es über dem anderen Element losgelassen wird. Aus den beiden Elementen entstehen zwei weitere, identische Elemente.

**Gamelogik: Aufteilen von Elementen**

Die Logik sowie die UI Aktionen für das Aufteilen von Elementen müssen umgesetzt werden. Wenn eine lange Berührung auf ein Element stattfindet, muss es sich in ein zufällig ausgewähltes Eltern-Element verwandeln. Damit wird auch das Austauschen der Grundelemente umgesetzt.

**Gamelogik: Energieleiste**

Die Logik sowie das UI für die Energieleiste müssen umgesetzt werden. Bei bestimmten Aktionen muss die Energieleiste an Energie verlieren. (Welche genau, müssen noch definiert werden). Wenn die Energieleiste leer ist, ist das Spiel vorbei und der Benutzer erhält erneut, die zur Verfügung stehenden Grundelemente. Die zuvor erhaltenen Elemente bleiben jedoch in der Auflistung der Elemente vorhanden.

**Gamelogik: Element Liste**

Die Logik sowie das UI für die Auflistung der Elemente. Im Hauptmenü des Spiels, sollen alle bisher erhaltenen Elemente aufgelistet werden. Zu jedem Element sollen jeweils das Bild, der Name sowie eine kurze Beschreibung ersichtlich sein.

**Gamelogik: Erfolg Liste**

Die Logik sowie das UI für die Auflistung der Erfolge. Im Hauptmenü des Spiels, sollen alle bisher freigeschalteten Erfolge, sowie eine bestimmte Anzahl nicht freigeschalteter Erfolge ersichtlich sein. Pro Erfolg sollen der Name und eine kurze Beschreibung ersichtlich sein. Bei nicht abgeschlossenen Erfolgen soll der Name und das Ziel, um den Erfolg freizuschalten, ersichtlich sein.

**Grafiken erstellen und einbinden**

Allgemeine Grafiken, wie beispielsweise der Titelbildschirm oder der Game-Over Bildschirm, sowie alle Grafiken für alle Elemente müssen erstellt (Pixelart) und später dem entsprechenden Datensatz zugeordnet werden. Ausserdem müssen die Grafiken an den entsprechenden Orten im Spiel angezeigt werden. Die Erstellung der Grafiken erfolgt nebenbei, während der gesamten Realisierungsphase bis der zugehörige Meilenstein erreicht ist.

**Tests**

Die einzelnen hinzugefügten Funktionen müssen fortlaufend, während der Entwicklung getestet werden. Ausserdem muss am Schluss ein Systemtest folgen, der möglichst alle Situationen und Elementkombinationen abdeckt.

**Entwicklungsdokumentation (Realisierungsbericht)**

Die Entwicklungsdokumentation wird fortlaufend, während der Entwicklung geführt und aktualisiert. Sie sollte die Architektur sowie wichtige Aspekte der Lösung abdecken.

*Einführungsphase*

**Zusatzinfos für Publishing schreiben**

Es müssen Zusatzinformationen, die für das Veröffentlichen auf den Stores benötigt werden, verfasst werden. Dazu gehören, Screenshots, ein App-Icon, eine Beschreibung möglichst auf Englisch und Deutsch.

**Einführungsbericht**

Der Einführungsbericht muss verfasst werden.

**Einführungspräsentation**

Die Präsentation muss vorbereitet werden, und ein PowerPoint muss angefertigt werden. Danach muss das Produkt präsentiert werden.

# Empfehlung

Wir empfehlen den Projektentscheid zu Gunsten der Variante A zu treffen und das Projekt freizugeben.

# Projektfreigabe

Hiermit bestätigt der Auftraggeber die Freigabe des Projektes:

­­­­­­­­­­­­­­­­­­­­­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Der Auftraggeber Der Projektleiter

(Ort, Datum, Unterschrift) (Ort, Datum, Unterschrift)