Konzeptbericht

|  |  |
| --- | --- |
| **Status** | In Arbeit / In Prüfung / Abgeschlossen |
| **Projektname** | Unsustainable |
| **Projektleiter** | Michael Günter, Elias Schmidhalter |
| **Auftraggeber** | Andres Scheidegger |
| **Autoren** | Michael Günter, Elias Schmidhalter |
| **Verteiler** | Michael Günter, Elias Schmidhalter, Andres Scheidegger |

**Änderungskontrolle, Prüfung, Genehmigung**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Datum | Beschreibung, Bemerkung | Name oder Rolle |
| 0.1 | 24.02.15 | Erstfassung | Elias Schmidhalter |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Definitionen und Abkürzungen**

|  |  |
| --- | --- |
| Begriff / Abkürzung | Bedeutung |
|  |  |

**Referenzen**

|  |  |
| --- | --- |
| Referenz | Titel, Quelle |
| [1] |  |
| [2] |  |
| [3] |  |

**Inhaltsverzeichnis**

[1 Zusammenfassung 3](#_Toc414342776)

[2 Systemanforderungen 3](#_Toc414342777)

[2.1 Entitätstypen 3](#_Toc414342778)

[1. Player (Spieler) 3](#_Toc414342779)

[2. Element 3](#_Toc414342780)

[3. Recipe (Rezept) 3](#_Toc414342781)

[4. Achievement (Erfolg) 3](#_Toc414342782)

[2.2 Anforderungen an die Funktionalität 4](#_Toc414342783)

[1. Drag and Drop 4](#_Toc414342784)

[2. Kombinieren 4](#_Toc414342785)

[3. Teilen 4](#_Toc414342786)

[4. Energie 4](#_Toc414342787)

[5. Energieleiste 4](#_Toc414342788)

[6. Rezepte 4](#_Toc414342789)

[7. Spielende 4](#_Toc414342790)

[8. Neues Spiel 4](#_Toc414342791)

[9. Achievements 4](#_Toc414342792)

[10. Achievements Übersicht 4](#_Toc414342793)

[11. Alchemy Table 4](#_Toc414342794)

[12. Element Details 4](#_Toc414342795)

[13. Elemente Übersicht 5](#_Toc414342796)

[14. Spass Elemente 5](#_Toc414342797)

[15. Grafiken 5](#_Toc414342798)

[16. Logos & Splashscreen 5](#_Toc414342799)

[17. Atmosphäre & Ambiente 5](#_Toc414342800)

[18. Sounds 5](#_Toc414342801)

[19. Texturen & Hintergründe 5](#_Toc414342802)

[20. Plattformen 5](#_Toc414342803)

[2.3 Arbeitspakete 5](#_Toc414342804)

[2.4 Anforderungen an die Informationssicherheit und den Datenschutz 5](#_Toc414342805)

[1. Lokale Datenbank 5](#_Toc414342806)

[2. Datenverbindungen 5](#_Toc414342807)

[3. Datenlöschung 5](#_Toc414342808)

[4. Schützenswerte Daten 5](#_Toc414342809)

[5. Persistenz 5](#_Toc414342810)

[3 Systemarchitektur 6](#_Toc414342811)

[3.1 Gliederung der Lösung in Module 6](#_Toc414342812)

[3.2 Schnittstellen 7](#_Toc414342813)

[1. Persistenz Schicht (Datenbank) 7](#_Toc414342814)

[2. Benutzerschnittstelle 8](#_Toc414342815)

[4 Testkonzept 11](#_Toc414342816)

[5 Weiterführung der Projektplanung 11](#_Toc414342817)

[5.1 Abgleich von Planung und tatsächlichem Verlauf der Phase Konzept 11](#_Toc414342818)

[5.2 Aktualisierung der Risikosituation 11](#_Toc414342819)

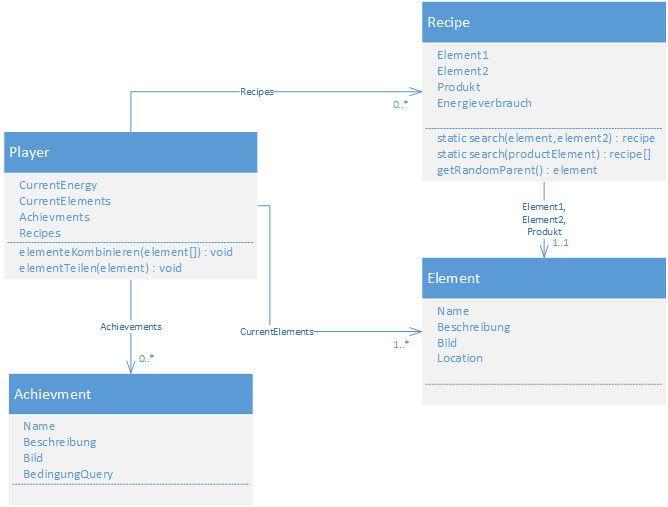
[5.3 Planung der nächsten Phase 11](#_Toc414342820)

# Zusammenfassung

Dieses Dokument dient der Veranschaulichung von Anforderungen und der Systemarchitektur. Die Schnittstellen (u.a. auch die Benutzerschnittstelle) sollen detaillierter beschrieben werden. Ausserdem enthält dieses Dokument ein Testkonzept.

# Systemanforderungen

## Entitätstypen



Wir haben uns für die vier Entitäten Spieler, Rezept, Element und Erfolg entschieden.

### Player (Spieler)

Der Spieler ist die Hauptentität. Er speichert seinen aktuellen Energiestand, die Elemente, die sich auf der Spielfläche befinden, die erhaltenen Achievements (Erfolge) sowie die erhaltenen Rezepte.

Er bietet Operationen an, um Elemente zu kombinieren und zu teilen.

### Element

Das Element besitzt einen Namen, eine Beschreibung, ein passendes Bild sowie, wenn auf der Spielfläche vorhanden, die aktuelle Position. Die genauen Elementdefinitionen sind in dem separaten Dokument ersichtlich.

### Recipe (Rezept)

Das Rezept besteht immer aus zwei Elementen, die für die Kombination benötigt werden. Das Rezept beinhaltet ein Produkt, welches aus den zwei Elementen entsteht. Das Rezept definiert den Energiewert, welcher benötigt wird um ein Element zu kombinieren. Die genauen Elementkombinationen sind in dem separaten Dokument ersichtlich.

### Achievement (Erfolg)

Das Achievement besitzt einen Namen, eine Beschreibung, ein passendes Bild und eine Bedingung, die erfüllt sein muss, damit man das Achievement erhält. Es stellt eine Art freischaltbares Ziel dar.

## Anforderungen an die Funktionalität

Wir haben uns für agile Softwareentwicklung entschieden. Deshalb ersetzen User-Stories die Use-Case Diagramme. In diesem Teil werden die User-Stories beschrieben.

### Drag and Drop

Als Spieler will ich die 6 Elemente auf meinem Spielfeld frei verschieben können. Dazu packe ich ein Element an und ziehe es über den Bildschirm an die gewünschte Position. Ich nehme nun den Finger von dem Bildschirm und das Element bleibt an der Stelle stehen.

### Kombinieren

Als Spieler, will ich Elemente kombinieren, wenn ich sie per Drag and Drop übereinander ziehe, und somit neue Elemente erschaffen. Die neuen Elemente werden anhand von Rezepten erstellt. Ein Rezept enthält immer zwei Teilelemente und das Produkt. Durch das Kombinieren möchte ich zwei Einheiten des neuen Elements erhalten. Die Teilelemente verschwinden beim Kombinieren.

### Teilen

Als Spieler, will ich Elemente teilen, indem ich lange auf ein Element drücke, und somit ein zufälliges Teilelement wieder zurückgewinne. Um die Auswahl an zufällig auswählbaren Teilelementen zu erlangen, wird das Rezept in umgekehrter Richtung angewandt. Das geteilte Element verschwindet und das neue Teilelement kommt zum Vorschein.

### Energie

Als Spieler, möchte ich, dass das Kombinieren und Teilen von Elementen Energie kostet. Teilen soll nur halb so viel Energie kosten wie das Kombinieren.

### Energieleiste

Als Spieler, will ich den aktuellen Energiestand anhand einer Energieleiste ablesen können.

### Rezepte

Als Spieler, will ich dass ich anhand von Rezepten Elemente teilen und kombinieren kann. In diesen Rezepten sollen der Energieverbrauch, die benötigten Teilelemente und das resultierende Element gespeichert werden. Der Energieverbrauch bezieht sich immer auf die Energie, die benötigt wird um ein Element zu kombinieren. Die genauen Kombinationsdefinitionen sind in dem separaten Dokument.

### Spielende

Als Spieler, will ich dass das Spiel endet sobald keine Energie mehr vorhanden ist. Wenn das Spiel zu Ende ist, soll eine Liste der im letzten Spiel erhaltenen Elemente angezeigt werden. Es gibt die Möglichkeit ein neues Spiel zu starten.

### Neues Spiel

Als Spieler, will ich bei Starten eines neuen Spiels eine volle Energiebar, sowie die Grundelemente erhalten. Die Sammlung der Achievements und der Elemente bleibt bestehen.  
Ein neues Spiel wird beim Ersten öffnen der App gestartet, sowie nach dem Spielende und beim klicken auf einen „Neues Spiel“ Button.

### Achievements

Als Spieler, will ich beim Kombinieren von bestimmten Elementen ein Achievement (Erfolg) erhalten. Ebenfalls, will ich nach einer bestimmten Anzahl von Elementen ein Achievement erhalten. Beim Erhalten soll eine kleine Meldung angezeigt werden und das Achievement soll in der Liste als freigeschaltet erscheinen. Die Achievements stellen eine Art freischaltbares Ziel dar.

### Achievements Übersicht

Als Spieler, will ich eine Liste, in der alle erhaltenen und möglichen Achievements beschrieben sind. Bei jedem Achievement sollen ein Bild, ein Name und eine kurze Beschreibung stehen.

### Alchemy Table

Als Spieler, will ich auf dem „Alchemy Table“ meine Elemente sehen, und Elemente teilen und kombinieren. Ich kann Elemente kombinieren indem ich ein Element via Drag and Drop auf ein anderes verschiebe.  
Ich will Elemente teilen indem ich lange auf das entsprechende Element tippe. Es sollen immer sechs Elemente auf dem Alchemy Table sein.

### Element Details

Als Spieler, will ich wenn ich auf ein Element tippe Infos zu diesem sehen. Ich will, Elementname, Beschreibung und die Teilelemente sehen.

### Elemente Übersicht

Als Spieler, will ich eine Übersicht mit allen schon gefundenen Elementen haben. Diese Übersicht soll in einer Liste dargestellt werden. Es soll das Bild des Elements, der Name und die Beschreibung angezeigt werden.

### Spass Elemente

Als Spieler, möchte ich einzelne fiktive oder lustige Spasselemente kreieren können. Auch lustige Elementbeschreibungen sind vorgesehen, die wir als lustig empfinden.

### Grafiken

Als Spieler, möchte ich gut aussehende Grafiken im Spiel haben. Es sollen vor allem Pixelgrafiken verwendet werden. Für jedes Element und jedes Achievement muss eine Grafik vorhanden sein.

### Logos & Splashscreen

Als Entwickler, brauchen wir ein Logo, um das Spiel im Store zu veröffentlichen. Ausserdem brauchen wir ein Splashscreen Bild.

### Atmosphäre & Ambiente

Als Spieler, möchte ich, dass eine etwas ältere, hölzerne und etwas düstere Atmosphäre vermittelt wird.

### Sounds

Als Spieler, will ich, dass beim Kombinieren und Teilen von Elementen passende Geräusche entstehen. Beim Erhalt von Achievements soll auch ein Geräusch abgespielt werden.

### Texturen & Hintergründe

Als Spieler, will ich dass die Texturen und Hintergrundbilder zur Atmosphäre passen. Für den „Alchemy Table“ und für das Menü müssen Texturen vorhanden sein.

### Plattformen

Als Spieler, will ich das Spiel auf iOS und Android spielen können.

## Arbeitspakete

Die Anforderungen wurden im Zeitplan (Gantt-Diagramm) den Arbeitspaketen zugeordnet.

## Anforderungen an die Informationssicherheit und den Datenschutz

Alle Daten werden lokal auf dem Gerät abgelegt. Es werden keine Daten über externe Verbindungen (Internet) übertragen. Wir speichern keine Daten über den Nutzer oder das Nutzungsverhalten. Der Nutzer hat die Möglichkeit alle Daten zu löschen, indem er die App entfernt.

Im Spiel fallen keine schützenswerten Daten an.

Jeder Nutzer ist verantwortlich, dass die Daten nicht durch äussere Einflüsse verändert werden.

Als äussere Einflüsse gelten: das Löschen der App, direktzugriffe auf den Datenspeicher, nicht vorhergesehene und geplante Veränderungen am Spiel.

Vor Datenverlust und Veränderungen durch die App schützt ein Persistenz Layer in der App. Bei einem Absturz können möglicherweise nicht gespeicherte Daten verloren gehen.

### Lokale Datenbank

Als Spieler will ich, dass alle Daten lokal abgespeichert werden.

### Datenverbindungen

Als Spieler will ich, dass meine Daten nicht über externe Verbindungen übertragen werden. Ich will, dass die App keine ein- oder ausgehenden Verbindungen erlaubt.

### Datenlöschung

Als Spieler will ich, dass alle Daten, wenn ich die App deinstalliere, gelöscht werden.

### Schützenswerte Daten

Als Spieler will ich, dass keine schützenswerte Daten gesammelt werden.

### Persistenz

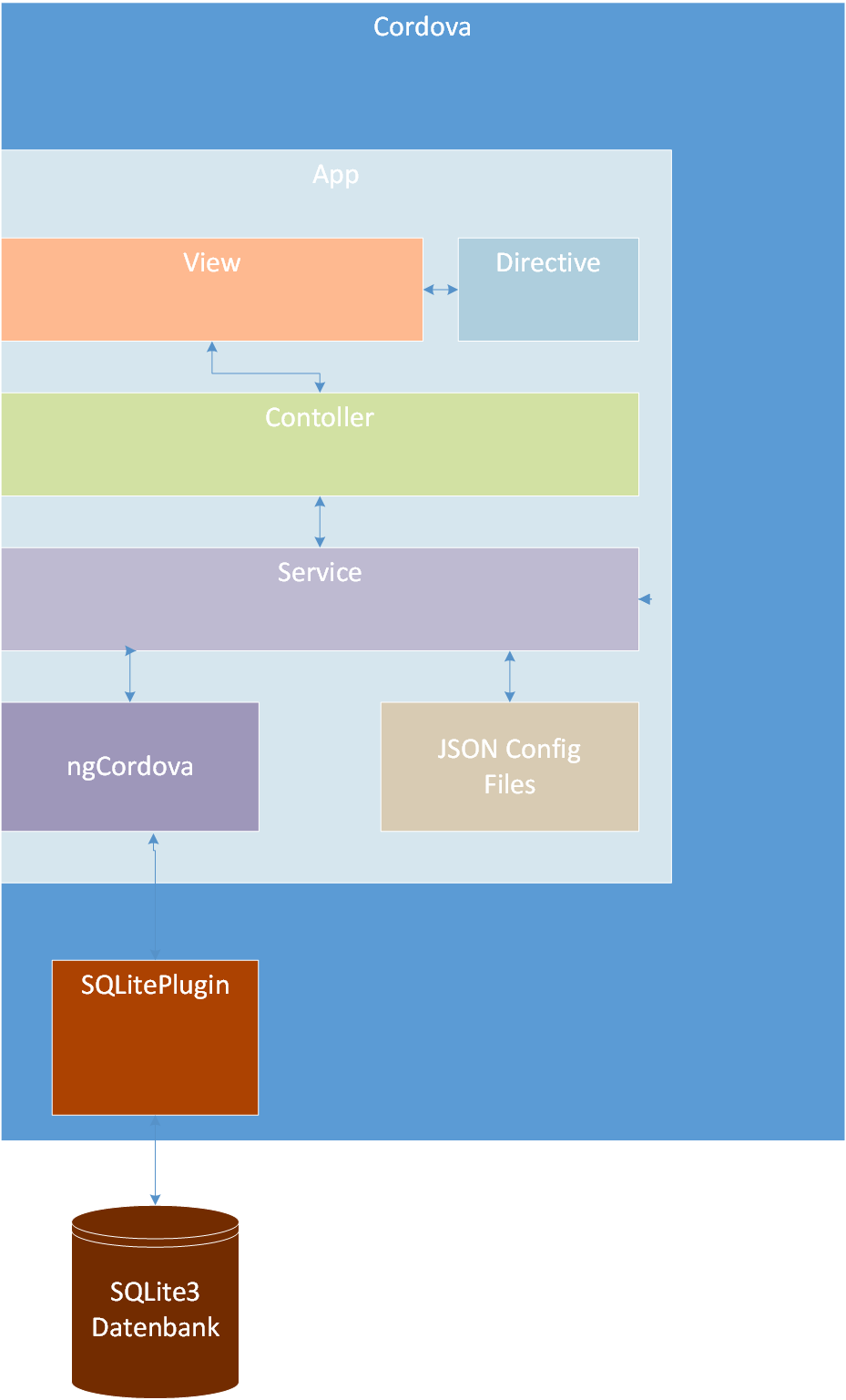
Als Entwickler will ich, dass die Daten persistent abgespeichert werden. Der Spieler ist selber für die Korrektheit der Daten verantwortlich. Der Spieler sollte die Daten nicht ausserhalb des Spiels modifizieren.

# Systemarchitektur

Für die Realisierung des Spiels haben wir die Variante gewählt, die Webtechnologien und normale Elemente verwendet. Umgesetzt soll das Ganze mit AngularJS und Apache Cordova werden.

## Gliederung der Lösung in Module

|  |  |  |
| --- | --- | --- |
| Nummer | Name | Beschreibung |
| 1 | App | App beinhaltet die Konfiguration der Applikation und ist das Hauptmodul. |
| 2 | Controllers | Controllers beinhalten die Controller für die einzelnen Seiten der Applikation. Ein Controller lädt die Daten in eine View. |
| 3 | Services | Services stellen im Normallfall Daten aus der Persistenz Schicht bereit. Services können auch Daten zwischenspeichern. Services sind wiederverwendbare Singleton-Instanzen. |
| 4 | Directives | Directives sind zusammengefasste, wiederverwendbare UI Komponenten. |
| 5 | View | In der View wird das Markup der Seite definiert. |

Die Applikation wird auf einem Mobiltelefon ausgeführt werden. Normalerweise entwickelt man Apps in den für die spezifischen Plattformen vorgesehenen Programmiersprachen (Android: Java, etc.).

Wir haben uns jedoch entschlossen, dass wir die App mit Webtechnologien aufbauen werden. Dazu existiert Apache Cordova, ein Framework welches eine Art Browserfenster zu Verfügung stellt, welches man benutzen kann um die App mit Webtechnologien aufzubauen. Apache Cordova bietet auch verschiedene Plugins, die beispielsweise die Interaktion mit der Kamera ermöglichen.

In unserem Fall umschliesst Apache Cordova unsere gesamte Applikation. Es enthält ein Plugin welches für die Kommunikation mit der lokalen SQLite Datenbank sorgt. Die SQLite Datenbank wird lokal auf dem Telefon abgelegt.

In der App selbst benutzen wir dann ngCordova, eine AngularJS Library zur Kommunikation mit den Cordova Plugins.

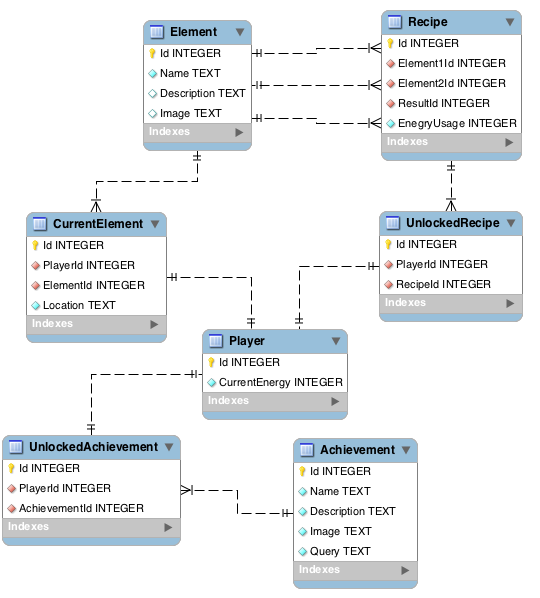
Grundsätzlich ist dann die App jedoch in einzelne Module gegliedert. Das Modul View enthält alle HTML-Dokumente, die benötigt werden um das UI bereitzustellen. Wenn einige Komponenten im UI mehrmals vorkommen, kann man diese Komponenten in ein Directive zusammenfassen. Dann gehört diese Komponente jedoch ins Modul Directives. Der Controller verwaltet dann die Daten, die auf der View angezeigt werden und er kommuniziert mit Services. Diese Services sind wiederverwendbare Singleton-Instanzen und können in den Controller injected werden. Die Services kommunizieren dann mit ngCordova und JSON Config Files um die Daten aus der Datenbank oder aus den Config-Files bereitstellen zu können.

## Schnittstellen

### Persistenz Schicht (Datenbank)

Für die Persistenz Schicht verwenden wir eine SQLite Datenbank. Zusätzlich können JSON Files für Konfigurationen verwendet werden. Der Zugriff auf die Datenbank geschieht über die Services, ngCordova und ein Cordova Plugin (brodysoft/cordova-sqliteplugin).

Anhand des folgenden Datenbank-Diagramms ist die Struktur der Daten in der Datenbank ersichtlich.



Grundsätzlich sind Elemente, Rezepte, Achievements und Player Tabellen die von uns schon vorabgefüllt werden und bei Beginn des Spiels schon Daten enthalten. Die Dinge, die der Spieler freischaltet, also alle Rezepte, alle Achievements und alle Elemente die er zur Zeit auf dem Alchemy Table hat, werden in den Tabellen UnlockedRecipe, UnlockedAchievement und CurrentElement gespeichert.

### Benutzerschnittstelle

Die Benutzerschnittstelle besteht hautsächlich aus dem Alchemy Table und dem Menü, welches aus der Element Übersicht und der Achievement Übersicht besteht.



Auf dem Alchemy Table befinden sich immer 6 Elemente. Am Anfang sind es die Grundelemente und danach sind es die Elemente die der Benutzer kombiniert. Das Konzept der Kombinationen ist bewusst so gewählt, dass sich immer 6 Elemente auf dem Table befinden.

Indem man einmal kurz ein Element berührt, kann man ein kleines Fenster öffnen, welches den Elementnamen, die Elementbeschreibung und die möglichen Elementkombinationen anzeigt.

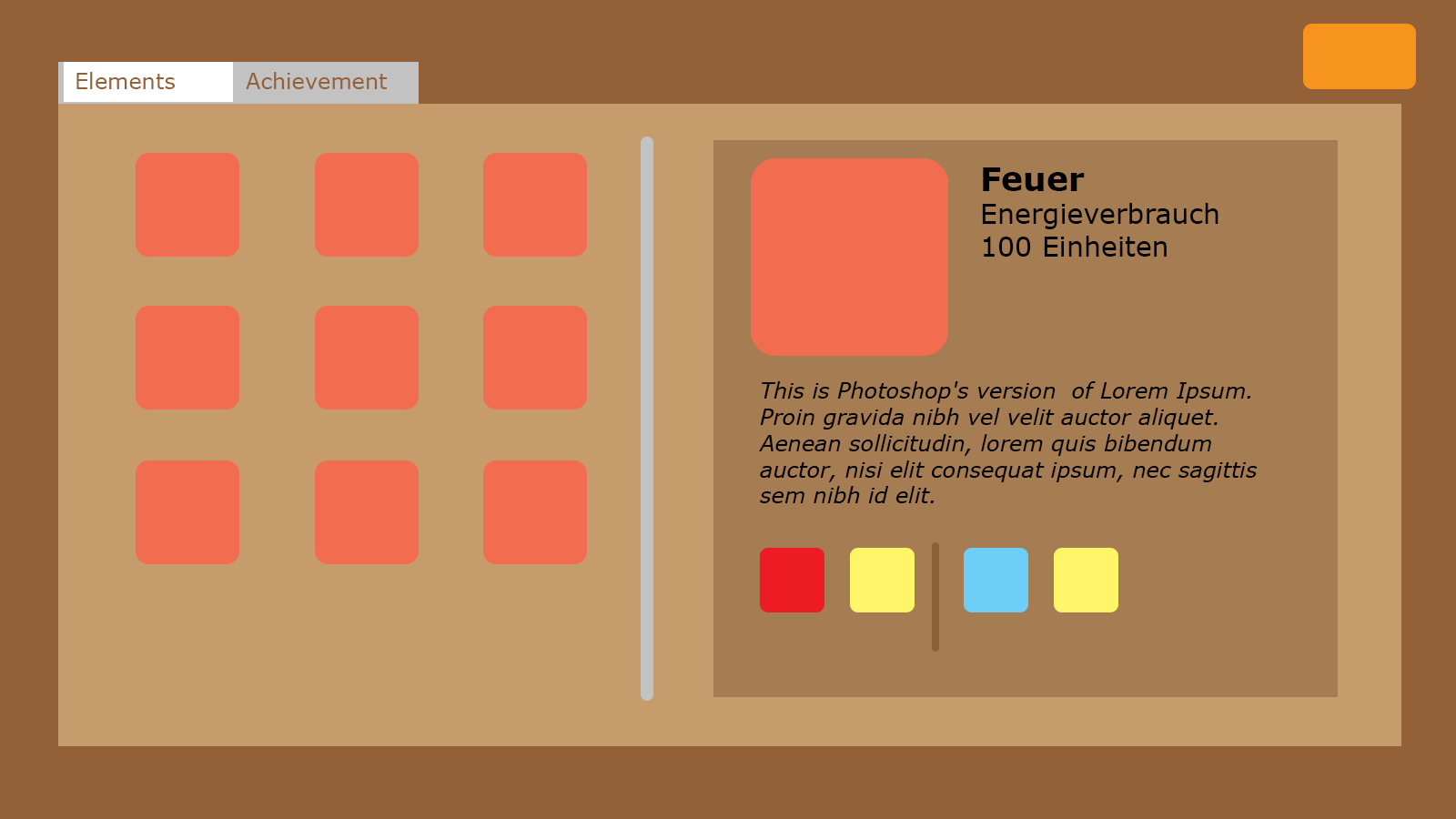
Indem man eine lange Berührung auf ein Element tätigt, wird es aufgeteilt.

Die Elemente lassen sich per Drag and Drop verschieben. Wenn ein Element auf ein anderes Element verschoben wird, werden die beiden Elemente kombiniert und zwei neue resultierende Elemente erscheinen.

Am oberen Bildschirmrand befindet sich die Energieleiste. Diese zeigt den aktuellen Energiestand an. Der Energiestand gibt an, wie viele Elemente noch aufgeteilt oder kombiniert werden können, bevor das Spiel zu Ende ist.

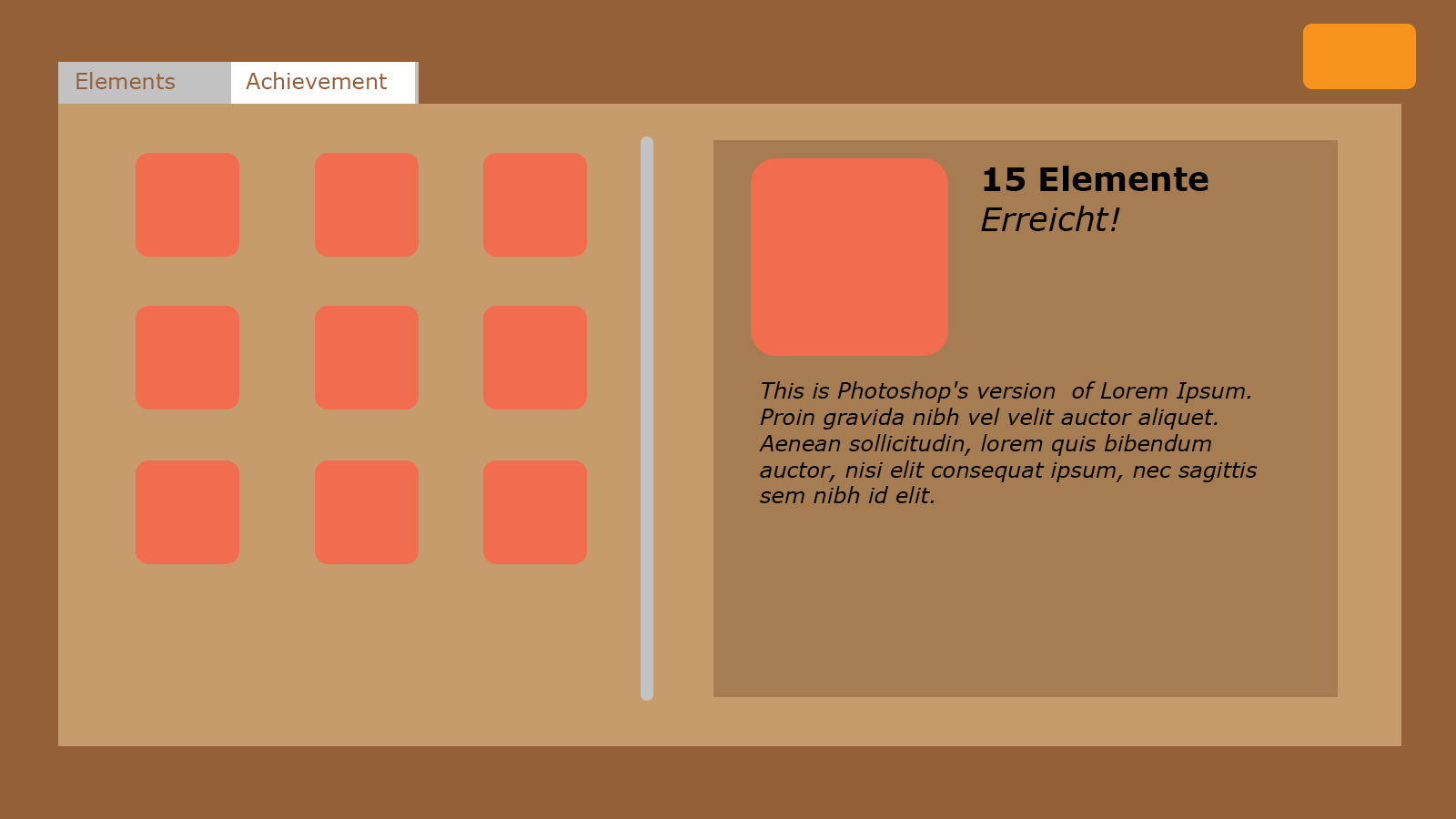
Neben der Energiebar befinden sich die Buttons für die Menüs. Es kann das Menü für die Elemente sowie das Menü für die Achievements aufgerufen werden.

Der gesamte Alchemy Table soll eine hölzerne Textur aufweisen, damit die notwendige Atmosphäre geschaffen wird.



In dem Menü für die Elemente werden auf der linken Seite alle Elemente, die man mittels Rezepten erlangt hat, aufgelistet. Man kann ein Element berühren um detaillierte Informationen zu dem Element zu erhalten.

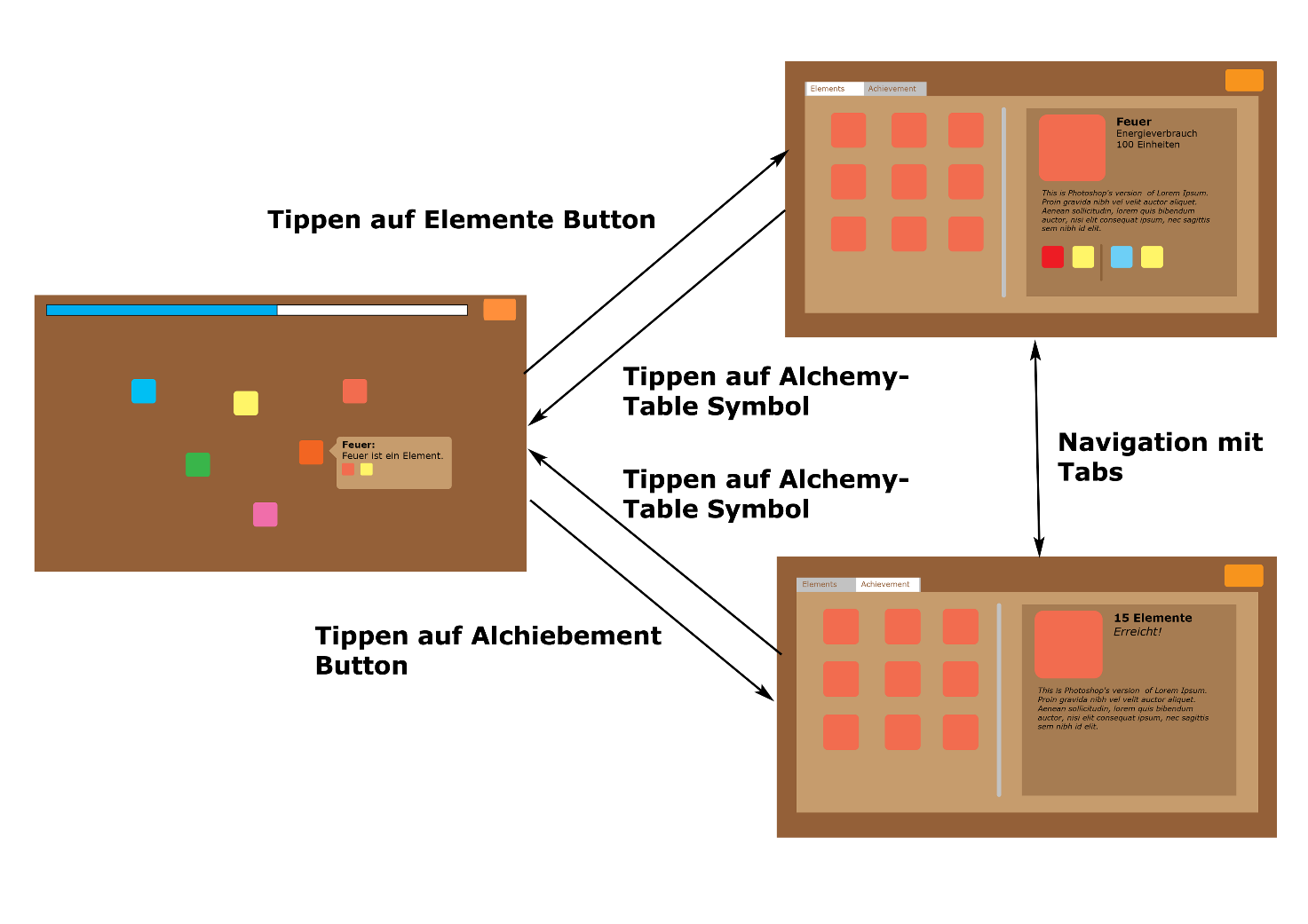
In den detaillierten Informationen sind der Name und der Energieverbrauch neben dem Bild ersichtlich. Unterhalb des Bildes wird eine kurze Beschreibung zu dem Element dargestellt und ausserdem kann man sehen, mit welchen Teilelementen man das Element kreieren kann.



In dem Menü für die Achievements werden auf der linken Seite alle Achievements angezeigt. Man kann ein Achievement berühren um detaillierte Informationen zu dem Achievement zu erhalten.

In den detaillierten Informationen werden das Bild, der Name, sowie die Beschreibung angezeigt. Ausserdem ist ersichtlich, ob das Achievement bereits erreicht wurde oder nicht.

In folgender Abbildung sind nochmals die Navigationspfade ersichtlich.



An der rechten oberen Ecke befinden sich zwei Buttons, mit welchen zwischen den Ansichten navigiert werden kann.

# Testkonzept

Während der Entwicklung wenden wir das Test-Driven-Development an. Das heisst, dass wir erst Testfälle (Unit Tests) schreiben und erst danach die Funktionalität umsetzen. Die Testfälle sind identisch zu den User Stories und sind dann erfüllt wenn alle User Stories erfolgreich und ohne Fehlermeldung durchgespielt werden können.

# Weiterführung der Projektplanung

## Abgleich von Planung und tatsächlichem Verlauf der Phase Konzept

Der Soll-Ist Vergleich (Gantt-Diagramm) wurde auf den aktuellen Stand gebracht und befindet sich im Anhang dieses Dokuments.

## Aktualisierung der Risikosituation

An der Risikosituation hat sich soweit nichts verändert.

## Planung der nächsten Phase

Die Planung für die nächste Phase befindet sich im Zeitplan (Gantt-Diagramm).