Anforderungskatalog

Inhalt

[Pojektziele 2](#_Toc436574333)

[System Kontext 3](#_Toc436574334)

[Personen / Stakeholders 3](#_Toc436574335)

[Hardware 3](#_Toc436574336)

[Rahmenvorgaben 3](#_Toc436574337)

[Abgrenzung 3](#_Toc436574338)

[Anforderungen 5](#_Toc436574339)

[1: Karten speichern 5](#_Toc436574340)

[2: Karten lösen (Solver) 5](#_Toc436574341)

[3: Sokoban spielen 5](#_Toc436574342)

[4: Spieler schnell verschieben 5](#_Toc436574343)

[5: Karte erstellen/editieren 5](#_Toc436574344)

[6: Karte teilen 5](#_Toc436574345)

[7: Solver Performance 5](#_Toc436574346)

[8: Anwendungsstabilität 5](#_Toc436574347)

[Glossar 6](#_Toc436574348)

# Pojektziele

Ziel des Projekts ist die Implementation eines realitätsnahen Prototypen des Sokoban-Spiels (gemäss Wikipedia <https://en.wikipedia.org/wiki/Sokoban> ) als Entscheidungsgrundlage für eine spätere Kommerzialisierung des Produkts. Die vorgegebene Software-Plattform ist Java.

Ebenfalls soll in diesem Projekt die Funktion und Performance eines automatischen Lösungsalgorithmus aufgezeigt werden.

Es sollen weitere Features implementiert werden, welche im Rahmen dieses Projekts genauer definiert werden. Das Endprodukt besteht aus einem Prototypen der Anwendung inklusive Dokumentation.

|  |  |
| --- | --- |
| Kategorie | Ziele |
| Wirtschaftlich | 1. Ermöglicht unserer Firma als Prototyp die Evaluation eines Produkts im Bereich der Sokoban-Spiele. Aufgrund des Produkts wird die Entscheidung getroffen, ob wir ein Sokoban-Spiel kommerzialisieren wollen. 2. Liefern von Erfahrungswerten für weitere Projekte: Auf Basis der Erfahrungswerte dieses Projekts kann die (Aufwands-)Planung weiterer Projekte verbessert werden. |
| System | 1. Das Produkt beziehungsweise der Prototyp unterstützt uns bei der Auswahl des besten Lösungsalgorithmus für Sokoban. 2. Der Prototyp zeigt auf welche der beiden Java-GUI-Frameworks für dieses Produkt optimal ist: Swing oder JavaFX. 3. Das Projekt zeigt auf, welche Features in einem Sokoban-Spiel mach- und vermarktbar sind. |
| Personell | 1. Die beteiligten Entwickler gewinnen Erfahrung mit GUI-Entwicklung und können somit flexibler eingesetzt werden. 2. Marktforschung: Finden, was Kunden von einem Sokoban-Spiel erwarten könnten. |
| Vorgehen | 1. Erstellung eines Prototypen, somit möglichst Ressourcen- und Funktionsorientierte Entwicklung. |
| Qualität | 1. Das Produkt ist als Sokoban-Spiel brauchbar. 2. Das Produkt verfügt über die nötigsten Sicherheitschecks, um stabil zu laufen. |

# System Kontext

## Personen / Stakeholders

|  |  |  |
| --- | --- | --- |
| Name | Rolle | Email |
| Peter Schwab | Stakeholder | peter.schwab@bfh.ch |
| Pascal Ammon | Auftragnehmer | [pascal.ammon@gmail](mailto:pascal.ammon@gmail).com |
| Gabriel Wyss | Auftragnehmer | gabriel.wyss@gmail.com |

## Hardware

Das Produkt ist für folgende Hardware bestimmt:

|  |  |
| --- | --- |
| Version | Hardware |
| Desktop | Intel basiert  Windows 7 oder neuer |
| Mobil / Android | HTC ONE M8 |

## Rahmenvorgaben

Die Rahmenvorgaben sind durch das Modul Projekt1 gegeben.

## Abgrenzung

Dieses Projekt umfasst die Entwicklung der Java-Applikation für Desktop und Android. Die Hardware, Betriebssystem oder andere Software gehören nicht zu diesem Projekt.

Das Produkt hat Schnittstellen mit dem Benutzer und dem darunter liegenden Betriebssystem. Es sind keine Schnittstellen mit weiteren Software-Systemen vorgesehen.

Sokoban-Projekt

Betriebssystem:

Windows oder Linux/MAC

Hardware

Java Runtime Environment

Sokoban Prototyp

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Anforderungen/Leistungsmerkmale der Software Sokoban** (2015-10-29) | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
| **Nr.** | | | **Kurzbezeichnung** | **Typ** | **Status** | **Prio** | **Stabilität** | **Komplexität** | **Risiko** | **Quelle** | **Datum** | **Ziele** |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| **1** |  |  | **Karten speichern** | Funktionell | geplant | 2 | 10 | Klein | 0 | Sitzung | 2015-10-29 | 1,2,5,6,7 |
| **2** |  |  | **Karten lösen (Solver)** | Funktionell | geplant | 1 | 10 | Hoch | 5 | Sitzung | 2015-10-29 | 1,2,5,6,7 |
| **3** |  |  | **Sokoban spielen** | Funktionell | geplant | 1 | 10 | Mittel | 0 | Sitzung | 2015-10-29 | 1,2,4,5,6,7 |
| **4** |  |  | **Spieler schnell verschieben** | Funktionell | geplant | 2 | 8 | Klein | 2 | Sitzung | 2015-10-29 | 1,2,5,6,7 |
| **5** |  |  | **Karte erstellen/editieren** | Funktionell | geplant | 1 | 10 | Mittel | 0 | Sitzung | 2015-10-29 | 1,2,5,6,7 |
| **6** |  |  | **Karte teilen** | Funktionell | geplant | 9 | 6 | Hoch | 6 | Sitzung | 2015-10-29 | 1,2,5,6,7 |
| **7** |  |  | **Solver Performance** | nichtfunktionell | geplant | 5 | 10 | Mittel |  | Sitzung | 2015-10-29 | 3,5,9,10 |
| **8** |  |  | **Anwendungsstabilität** | nichtfunktionell | geplant | 4 | 10 | Mittel |  | Sitzung | 2015-10-29 | 1,9,10 |

Funktionale Anforderungen:

Beschreiben Sie (in etwa) 6 funktionale Anforderungen natürlichsprachig.(davon allenfalls 2 auf Stufe Grob-Anforderungen [komplex] und 4auf Stufe Detail-Anforderungen [Komplexität reduziert]; je nach Thema/Ausgangslage).

Unterteilen Sie trotz der (aufgrund der Übungsanlage bewusst) kleinen Menge an Anforderungen die Darstellung in eine „Anforderungsübersicht (Kurzbezeichnung)“ und eine „Detailbeschreibung (textliche Langausführung)“. Führen Sie ggf. die Spalten „Priorität“, „Variabilität“, „Komplexität“ mit einer entsprechenden Bewertung (z.B. drei-stufig) und überlegen Sie sich gestützt darauf eine „Formel“ mit der Sie das „Risiko“ der Anforderung bewerten würden (z.B. drei

-stufig). Vergessen Sie den Zielbezug nicht!

Nicht-Funktionale Anforderungen („Qualitätsanforderungen“):

Beschreiben Sie mind. 2 nicht-funktionale Anforderungen bei Ihrer Themenstellung

# Anforderungen

## 1: Karten speichern

Die Anwendung kann Spielkarten auf dem Dateisystem abspeichern und einlesen. Das im Internet gängige Textformat der Karten soll dazu benutzt werden.

Gemäss: <http://www.sourcecode.se/sokoban/levels>

## 2: Karten lösen (Solver)

Die Applikation löst das Spiel über einen noch zu definierenden Algorithmus und zeigt dem Benutzer die Lösung grafisch dar. Dazu kann der Benutzer während dem Spiel die Lösungsfunktion auslösen. Die Performance des Lösungsalgorithmus wird in Anzahl verschobenen Kisten und Anzahl Spielfigurzügen gemessen.

## 3: Sokoban spielen

Das Sokoban-Spiels kann gemäss den bekannten Spielregeln auf einer grafisch ansprechenden Oberfläche mittels Maus und Tastatur gespielt werden.

Der Benutzer kann seinen Avatar/Spielfigur mit den 4-Pfeiltasten nach links, rechts, rauf und runter navigieren. Der Avatar kann nur auf den Feldern bewegt werden, die gemäss Spielregeln dafür vorgesehen sind.

Der Benutzer kann zwischen mehreren grafischen Styles („Themes“) auswählen.

## 4: Spieler schnell verschieben

Mittels eines Klicks auf die Karte kann der User die Spielfigur augenblicklich auf das betroffene Spielfeld verschieben. Die Aktion wird nur ausgeführt, wenn die mit einem erlaubten Spielzug möglich ist. Es darf zum Beispiel keine Mauer im Weg stehen.

## 5: Karte erstellen/editieren

Der Benutzer kann eigene Karten für das Spiel bauen oder bestehende Karten editieren. Der Editor erlaubt es, eine Kartengrösse (in Spielsteinen), Kartennamen und Kartenschwierigkeit anzugeben Während dem Bau stellt die Applikation die Lösbarkeit sicher. Es dürfen nur lösbare Karten gebaut werden. Beim Bauen der Karte werden automatisch nur lösbare Bauoptionen angezeigt. Die Karte kann im Karteneditor nach Schwierigkeitsgrad klassifiziert werden. Das Endprodukt wird mit mindestens 5 Beispielkarten geliefert.

## 6: Karte teilen

Der Benutzer kann seine selbst erstellten Karten mit anderen Spielern teilen.

## 7: Solver Performance

Durch Nutzung von im Internet frei verfügbaren und selbst programmierten Solver-Lösungen soll der optimale Solver-Algorithmus für Sokoban gefunden werden. Es sind auch Optimierungen von bestehenden Algorithmen denkbar.

## 8: Anwendungsstabilität

Die Anwendung überprüft Benutzereingaben auf Korrektheit, um Anwendungsstabilität zu gewährleisten. Ebenfalls werden die möglichen Exceptions abgefangen, damit der Prototyp möglichst stabil wird.

# Glossar

|  |  |
| --- | --- |
| Wort | Beschreibung |
| GroundField | Grundfeld |
|  |  |