

Berufsakademie Sachsen  
Staatliche Studienakademie Leipzig

## **Pflichtenheft**

### **Android-Spielesammlung**

Seminargruppe: IT2011

Auftraggeber: Prof. Dr. Christian heller  
BA Leipzig  
Schönauer Straße 113a  
04207 Leipzig

Auftragnehmer: André Heinicke  
Paul Jähne  
Martin Kimmel  
Sascha Kühnel

Leipzig, 24.01.2014

---

# Inhaltsverzeichnis

<b>1 Zielbestimmung.....</b>	<b>4</b>
1.1 Musskriterien.....	4
1.2 Wunschkriterien.....	4
1.3 Abgrenzung.....	5
<b>2 Produkteinsatz.....</b>	<b>5</b>
2.1 Anwendungsbereich und Zielgruppe.....	5
2.2 Betriebsbedingungen.....	6
<b>3 Umgebung.....</b>	<b>6</b>
3.1 Produktumgebung.....	6
3.1.1 Software.....	6
3.1.2 Hardware.....	6
3.2 Entwicklungsumgebung.....	7
3.2.1 Software.....	7
3.2.2 Hardware.....	7
<b>4 Produktinformationen.....</b>	<b>7</b>
4.1 Anwendungsbereiche.....	7
4.2 Benutzerfunktionen.....	8
4.3 Persistenz.....	8
<b>5 Produktdaten.....</b>	<b>9</b>
<b>6 Produktleistungen.....</b>	<b>9</b>
<b>7 Benutzeroberfläche.....</b>	<b>9</b>
7.1 Hauptmenü.....	9
7.2 Sudoku.....	10
7.3 Mines.....	10
7.4 Spaceslider.....	11
<b>8 Testszenarien und Tests.....</b>	<b>11</b>

---

<b>9 Verzeichnisse.....</b>	<b>14</b>
9.1 Abkürzungsverzeichnis.....	14
9.2 Abbildungsverzeichnis.....	14

# 1 Zielbestimmung

In der zu entwickelnden Spielesammlung sollen dem Nutzer verschiedene Spiele in einer Applikation (App) für das Android-Betriebssystem bereitgestellt werden. In dieser App wählt der Nutzer das gewünschte bereitgestellte Spiel und es werden ihm verschiedene Optionen zum Spielen bereitgestellt.

## 1.1 Musskriterien

- Nutzer kann das gewünschte Spiel im Hauptmenü wählen
- Übersicht der Bestleistung im jeweiligen Spiel („Highscore“)
- Spiele
  - Sudoku
  - Mines (Minesweeper Nachbau)
  - Spaceslider (Ausweichspiel)
- Spieloptionen
  - Beim Start des Spiels wählt Nutzer, ob ein neues Spiel gestartet werden soll oder ob ein zuvor gespeichertes Spiel weitergespielt werden soll
  - Spiele sollen gespeichert, geladen und neu erstellt werden können
  - Nutzer soll mit Hilfe von Toucheingaben bzw. den Beschleunigungssensoren das Spiel bedienen können
  - Beim beenden eines laufenden Spiels kann das Spiel nach Wahl gespeichert oder verworfen werden
- Spielstände werden persistent hinterlegt
- Sprachliche Internationalisierung: Deutsch und Englisch

## 1.2 Wunschkriterien

- Benutzerfreundliche Installationsroutine
- Nutzer kann mehr als ein Spielstand zu jedem Spiel anlegen

- Unterschiedliche Schwierigkeitsgrade zum Spiel
- Modulare Entwicklung zum Nachträglichen einfügen von weiteren Spielen
- Optionale Hilfe bzw. Lösungsvorgaben
- Anpassung der Oberfläche auch an Android-Tablets
- Mehrere Highscores pro Spiel

### **1.3 Abgrenzung**

- Kein Nutzerhandbuch
- Keine Erklärung der Spielregeln
- Keine Online-Hilfen
- Keine Cloud-Anbindung und zugehörige Funktionen für die Spiele bzw. deren Highscores und Speicherstände
- Kein Multi-User-Support
- Android-Versionen unter Version 4.0 werden nicht berücksichtigt
- Android-Tablets im Sinne der Spieloberfläche sollen vorerst nicht beachtet werden
- Eine Portierung bzw. Entwicklung für iOS, Windows Mobile 8 oder anderen Mobilien Betriebssystemen ist nicht vorgesehen

## **2 Produkteinsatz**

### **2.1 Anwendungsbereich und Zielgruppe**

Die Software dient der kurzweiligen Beschäftigung für Personen im Alter ab 8 Jahren. Das Lesen Deutscher oder Englischer Sprache ist dabei Voraussetzung. Der Anwender sollte die Bedienung seines Gerätes mittels Touchscreen und Bewegungssensoren beherrschen.

Die Spielregeln für Mines, was einen Minesweeper-Nachbau darstellt, und insbesondere für Sudoku sollten im Sinne einer sinnvollen Nutzung bekannt sein.

## **2.2 Betriebsbedingungen**

Die Software muss auf einem Android-Smartphone mit einer Android-Version von mindestens 4.0 laufen. Rechte zum Speichern lokaler Daten müssen vorhanden sein.

Auf einen Dauerhaften Betrieb wird kein Wert gelegt. Dennoch soll es vermieden werden, unvorhersehbare Bedingungen herzustellen. Bei Texteingaben sollen möglichst nur sinnvolle Werte eingearbeitet sein. Wird z.B. eine Zahl erwartet, so soll die mit dem Eingabefeld verknüpfte virtuelle Tastatur lediglich über Zifferntasten verfügen.

## **3 Umgebung**

### **3.1 Produktumgebung**

Die Spielesammlung soll nur für das Android Betriebssystem entwickelt werden. Hauptaugenmerk wird dabei auf Android-Smartphones gelegt. Für Android-Tablets werden die generellen Produktfunktionen unterstützt. Grafisch müssen die Spiele bei unterschiedlichen Bildschirmauflösungen scharf und verzerrungsfrei dargestellt werden.

Die Geschwindigkeit des Spielablaufs soll geräteunabhängig auf einem nahezu gleichem Niveau liegen. Diese Bedingung gilt nicht bei Geräten, welche die minimalen Hardwareanforderungen nicht erfüllen.

Weiterhin ist das Produkt für den Einsatz im privaten Umfeld konzipiert.

#### **3.1.1 Software**

- Android Version ab 4.0 vorhanden
- Rechte zum Speichern von Dateien auf dem Smartphone benötigt

#### **3.1.2 Hardware**

- Smartphone
- Unterstützung von Beschleunigungssensoren
- Ausreichend Speicher und Rechenleistung
- ARM-Architektur

- Bildschirmauflösung: mind. 320 x 480 Pixel

## **3.2 Entwicklungsumgebung**

### **3.2.1 Software**

Die App soll unter Windows 7 mit Hilfe der Programmiersprache Java umgesetzt werden. Dazu wird als Entwicklungsumgebung Eclipse mit den Android SDK genutzt.

Die UML Diagramme wurden in Bouml umgesetzt. Das Tool Pencil wurde für den Oberflächenentwurf der App genutzt.

GitHub soll zur Versionsverwaltung verwendet werden.

### **3.2.2 Hardware**

Es wird ein internetfähiger Rechner für die Entwicklung vorausgesetzt.

## **4 Produktinformationen**

Die Spiele Sudoku und Mines genügen den jeweiligen Regeln dieser Spiele. Da Spaceslider sich nicht an einem bekannten Spiel orientiert folgt hier eine kurze Spielbeschreibung:

Der Spieler steuert mittels Bewegungssensor (Kippen) ein virtuelles Objekt (Raumschiff) auf einer horizontalen Bildschirmachse hin und her. Dabei gilt es, anderen entgegenkommenden Objekten auszuweichen. Diese Objekte werden vom Spiel in definierter Menge an zufälligen Positionen generiert. Falls verschiedene Schwierigkeitsgrade umgesetzt werden, ist dieser durch die Bewegungsgeschwindigkeiten und die Anzahl entgegenkommender Objekte bestimmt. Unabhängig vom eingestellten Schwierigkeitsgrad nimmt die Schwierigkeit mit der Spielzeit zu. Die erreichte Punktzahl ergibt sich aus der Zeit, welche der Spieler das von ihm gesteuerte Objekt vor der Zerstörung durch Kollision bewahren kann.

### **4.1 Anwendungsbereiche**

Die Benutzungsmöglichkeiten sollen in den folgenden Abbildungen in einem Use-Case-Diagramm und der Ablauf der Interaktion mit der App in einem Aktivitätsdiagramm dargestellt werden.

#### **UML-Diagramme**

## **4.2 Benutzerfunktionen**

### **/F100/ Auswahlmenü**

Dem Benutzer wird beim Starten der Anwendung eine Menüoberfläche angezeigt, in welcher er zwischen den 3 verfügbaren Spielen auswählen kann. Außerdem wird zu jedem Spiel der aktuelle „Highscore“ angezeigt.

### **/F200/ Speichern von Spielständen**

Der Benutzer soll die Möglichkeit besitzen den aktuellen Spielstand speichern zu können.

### **/F210/ Laden von Spielständen**

Es soll möglich sein mindestens einen Spielstand zu laden und diesen fortzusetzen.

### **/F220/ Neustarten von einem Spiel**

Es soll für den Nutzer die Möglichkeit geben seine Eingaben im aktuellen Spielstand bei dem Spiel Sudoku zu löschen. Außerdem soll es möglich sein bei „Spaceslider“ und „Mines“ das Spiel neu zu starten.

### **/F300/ Erstellung von „Highscores“**

Wird eine neue Bestleistung in einem Spiel erreicht, so soll diese hinterlegt werden.

### **/F400/ Schwierigkeitsgradeinstellung (Wunschkriterium)**

Um den Spielspaß zu gewähren, wäre es wünschenswert den Schwierigkeitsgrad des jeweiligen Spiels anpassen zu können. So werden mit Erhöhung des Schwierigkeitsgrades bei „Sudoku“ bei Start des Spieles weniger Zahlen angezeigt, bei „Mines“ mehr Minen erstellt und der Streuradius erniedrigt und bei „Spaceslider“ die Spielgeschwindigkeit erhöht.

## **4.3 Persistenz**

Für die Persistenz sollten die unter Android üblichen Wege gegangen werden. Denkbar ist z.B. die Nutzung von SharedPreferences zur Speicherung von Schlüssel-Wert-Paaren, d. h. es werden lokale XML-Dateien zur Datenhaltung verwendet.



---

Zu speichernde Daten sind Höchstpunktzahlen, Spielernamen und Spielstände.

## 5 Produktdaten

### /D100/ Highscore-Daten

- Höchstpunktzahl
- Spiel
- Spielername
- Schwierigkeitsgrad

### /D200/ Spielstand-Daten

- Spiel
- Schwierigkeitsgrad
- Spielfläche
- Punktestand

## 6 Produktleistungen

Es werden keine qualifizierbaren Leistungsanforderungen an die Spielesammlung gestellt.

## 7 Benutzeroberfläche

### 7.1 Hauptmenü

Wenn die App gestartet wurde, sieht der Nutzer eine Übersicht (Dashboard) über die verfügbaren Spiele. Die Spiele werden Zeilenweise dargestellt, zuerst steht der Name und dann der jeweilige Highscore. Am Ende der Zeile ist ein Plus-Symbol mit dem sich das jeweilige Spiel starten lässt. Falls ein gespeichertes Spiel vorhanden ist, folgt eine Abfrage, ob dieses fortgesetzt werden soll. Ein Modell ist in Fehler: Referenz nicht gefunden dargestellt. Die Sprache passt sich der eingestellten Systemsprache an. Unterstützte Sprachen sind Deutsch und Englisch.

**Abbildung Hauptmenü**

## 7.2 Sudoku

Die Benutzeroberfläche des Sudokus soll so gestaltet sein, dass alle Funktionen ohne Schwierigkeiten per Toucheingabe bedient werden können. Nachfolgend ist ein Entwurf der Benutzeroberfläche bei Start eines neuen Spieles dargestellt. Das Plus links unten im Bild soll zur schnellen Erstellung eines neuen Spieles dienen. Dabei soll nachgefragt werden, ob das bereits angefangene Spiel gespeichert werden soll. Das Kreuz daneben soll die Funktion zum Löschen aller bisher gemachten Eingaben übernehmen. Neben diesem Symbol befindet sich das Menü Icon, welches bei Toucheingabe das Menü des Sudokus öffnen soll. Außerdem soll neben diesem die Zeit seit Beginn des Spiels dargestellt werden. Aus dieser ergibt sich die Bestleistung. Wird ein Spiel neu gestartet soll der Nutzer zur Auswahl zwischen einem neuen Spiel und einem gespeicherten Spielstand aufgefordert werden.

### Abbildung Sudoku

Die folgende Abbildung soll eine schematische Eingabe widerspiegeln. Diese soll über die in Android implementierte Tastatur des Gerätes verwirklicht werden. Bei der Auswahl eines Feldes soll diese erscheinen. Wurde die Eingabe getätigt, soll diese für eine bessere Übersicht wieder ausgeblendet werden.

### Abbildung Sudoku mit Eingabe

Wird der Menüknopf auf der rechten unteren Hälfte des Bildschirmes betätigt, soll ein Menü aufklappen, welches ähnlich wie in dieser Abbildung dargestellt sein soll. Dieses enthält die Funktionen „Neues Spiel“, „Laden“, „Speichern“. Diese sollen wie in den Benutzerfunktionen F200, F210, F220 beschrieben fungieren.

### Abbildung Sudoku mit Menü

## 7.3 Mines

Die Oberfläche besteht aus dem Spielfeld und einer Menüleiste. Im Spielfeld befindet sich ein Feld von Schaltflächen, die durch Antippen aktiviert werden und dadurch das entsprechende Feld aufdecken. Falls darunter eine Zahl steht, wird nun diese angezeigt (farbige Zahlen?) und bei einem leeren Feld werden alle umliegenden Felder aufgedeckt. Wenn Allerdings eine Mine unter dem Feld ist, hat man das Spiel verloren und das Spielfeld wird aufgedeckt. Wenn das Spiel geschossen wird, erfolgt eine Abfrage, ob der Spielsand gespeichert oder verworfen werden soll. Ein Entwurf ist in Fehler: Referenz nicht gefunden zu sehen.

### **Abbildung Mines**

In der Menüleiste befindet sich ein Plus-Symbol zum schnellen Starten eines neuen Spiels. Mit dem Kreuz-Symbol kann zwischen dem normalen Aufdecken des Feldes und dem Setzen eines Fähnchens zum Markieren von Minen umgeschaltet werden. Rechts davon kann über das Menüsymbol auf die Optionen zugegriffen werden. Darin befindet sich das Erstellen eines neuen Spiels, das Laden und Speichern des Spiels und eine Option zum Anzeigen des Highscores. Rechts in der Menüleiste befindet sich eine Zeitanzeige und die Anzahl der verbleibenden Minen. Falls verschiedene Schwierigkeitsgrade implementiert werden, unterscheiden sich diese durch verschieden große Felder und Anzahl von Minen: einfach mit einem 9x9-Feld und 10 Minen und normal mit einem 16x16-Feld und 40 Minen. Ein Beispiel mit aufgeklapptem Menü ist in Fehler: Referenz nicht gefunden zu sehen.

### **Abbildung Mines mit Menü**

## **7.4 Spaceslider**

Während des Spiels ist ein schwarzer Hintergrund zu sehen. Im unteren Bereich ist das vom Spieler gesteuerte Objekt platziert. Von oben kommen helle Punkte (Sternenhimmel) und die ebenfalls hellen Fremdobjekte.

Der Spieler muss während des Spiels stets den Zustand seines Objektes und die bisher erreichte Punktzahl ablesen können. Diese Daten befinden am Bildrand, wo sie das Spielgeschehen nicht negativ beeinflussen.

### **Abbildung Spaceslider**

Wird das Menü des Spiels dargestellt, ist der Spielfluss zu unterbrechen. Deshalb muss das Menü den Großteil des Spielgeschehens verdecken, damit der Spieler aus dem Standbild keinen Vorteil gewinnen kann.

### **Abbildung Spaceslider mit Menü**

## **8 Testszenarien und Tests**

### **/T100/ Installationstest**

Dieser Test soll sicherstellen, dass die App mit jedem Android-Gerät mit einer Version ab 4.0 genutzt werden kann.

**/T200/ Spiel auswählen**

Es kann eines der verfügbaren Spielen ausgewählt werden.

**/T300/ Neues Spiel erstellen**

Es kann ein neues Spiel erstellt werden.

**/T310/ Spielstand speichern**

Es kann der aktuelle Spielstand gespeichert werden.

**/T320/ Spielstand laden**

Es kann mindestens ein gespeicherter Spielstand ausgewählt und geladen werden.

**/T330/ Spiel neu starten**

Es kann das aktuelle Spiel neugestartet werden.

**/T400/ Schwierigkeitsgrad ändern (Wunschkriterium)**

Nach Implementierung von Schwierigkeitsgraden können diese korrekt eingestellt werden.

**/T500/ Erstellung von „Highscores“**

Mindestens eine Bestleistung wird pro Spiel gespeichert und kann übertroffen werden.

**/T600/ Auswahlmenü**

Das Startmenü, die Spiele und die Menüs der Spiele können problemlos via Toucheingabe des jeweiligen Gerätes bedient werden.

**/T700/ Spielregeln**

Die Spiele halten sich an die bekannten Spielregeln und es ist möglich das Spiel zu gewinnen.

**/T800/ Reaktion auf Fehleingaben**

Da bei dem Spiel „Sudoku“ die eingebaute Tastatur aufklappt, wäre es möglich nicht zulässige Eingaben zu machen. Dies soll verhindert und darauf reagiert werden.

**/T900/ Blackbox Test**

Die Spielesammlung soll auch von Leuten außerhalb der Entwicklergruppe ohne vorherige Einweisung getestet werden.

## **9 Verzeichnisse**

### **9.1 Abkürzungsverzeichnis**

App                                      Applikation

### **9.2 Abbildungsverzeichnis**