Inhaltsverzeichnis

[Einleitung 2](#_Toc474402806)

[Hinweise zur Lesbarkeit 2](#_Toc474402807)

[Rechtliche Hinweise 2](#_Toc474402808)

[Abkürzungen 2](#_Toc474402809)

[Abbildungen 2](#_Toc474402810)

[Tabellen 3](#_Toc474402811)

[1. Projektbeschreibung 3](#_Toc474402812)

[1.1 Projektumfeld 3](#_Toc474402813)

[1.2 Projektziel 3](#_Toc474402814)

[1.3 IST-Zustand 3](#_Toc474402815)

[1.4 SOLL-Zustand 3](#_Toc474402816)

[1.5 Personen 3](#_Toc474402817)

[2. Projektplanung 4](#_Toc474402818)

[2.1 Zeitplanung 4](#_Toc474402819)

[2.2 Sachmittelplanung 4](#_Toc474402820)

[3. Projektdurchführung 5](#_Toc474402821)

[3.1 Erstellen der Level-Klasse 5](#_Toc474402822)

[3.2 Erstellen der Main Activity 5](#_Toc474402823)

[3.3 Erstellen der Grafiken 5](#_Toc474402824)

[3.4 Anzeigen der Grafiken im Spiel R 6](#_Toc474402825)

[3.5 Erstellen der Datenbank-Klasse K 6](#_Toc474402826)

[3.6 Erstellen des Logins und der Spielmodus-Auswahl R 6](#_Toc474402827)

[3.7 Nutzen der Datenbank für Login R 6](#_Toc474402828)

[3.8 Implementieren der Level-Auswahl RK 6](#_Toc474402829)

[3.9 Hinzufügen der Zeitmessung R 6](#_Toc474402830)

[3.10 Anpassen des Layouts K 7](#_Toc474402831)

[3.11 Erweitern der Datenbank-Funktionalität RK 7](#_Toc474402832)

[3.12 Erweiterung der Spielmodus-Auswahl auf 2 Spieler R 7](#_Toc474402833)

[3.13 Implementieren der Highscore-Klasse RK 7](#_Toc474402834)

[4. Projektergebnis 7](#_Toc474402835)

[4.1 Fehlerbehebung 7](#_Toc474402836)

[4.2 Abweichungen 7](#_Toc474402837)

Einleitung

Diese Dokumentation wurde im Rahmen einer schulischen Projektarbeit an der Berufsschule in Freising im Fach Anwendungsentwicklung und Programmierung zur Berufsausbildung zum Fachinformatiker für Anwendungsentwicklung erstellt. Sie beinhaltet die Erstellung einer Applikation für Smartphones mit den hierbei aufgekommenen Problemen und deren Lösungsansätzen. Ziel des Projektes ist es, die Lerninhalte des Unterrichts zu vertiefen und ein realitätsbezogenes Projekt umzusetzen.

Katharina Ambrosch, Robert Kais

# Hinweise zur Lesbarkeit

Zur besseren Lesbarkeit wurde der Quellcode durch eine gesonderte Schriftart hervorgehoben. Klickanweisungen wurden ebenfalls mit einer eigenen Schriftart hervorgehoben.

# Rechtliche Hinweise

Die Spiele-App sowie die dazugehörigen Dokumentations-, Präsentations- sowie alle weiteren Unterlagen sind geistiges Eigentum von Ambrosch Katharina und Kais Robert. Diese Dokumentation darf ohne Einverständnis der o. g. Personen weder vollständig noch auszugsweise kopiert, übertragen, vervielfältigt, veröffentlicht, geändert oder dupliziert werden. Ungeachtet der o. g. Bestimmungen sind Lehrer der Berufsschule Freising dazu berechtigt, zum schulischen Gebrauch die Dokumentation zu veröffentlichen, vorausgesetzt es bleibt ersichtlich, dass das Urheberrecht bei o. g. Personen liegt.

# Abkürzungen

Folgende Abkürzungen tauchen in der Dokumentation auf:

* AEP – Anwendungsentwicklung und Programmierung
* App – Applikation
* SP – Service Pack
* FMG – Flughafen München GmbH
* Abb. - Abbildung

# Abbildungen

Abb. 1 – Zeitplanung  
Abb. 2 – Level-Boden  
Abb. 3 – Spielfigur  
Abb. 4 – Crate Maze-Logo  
Abb. 5 – Button für Einspieler-Modus  
Abb. 6 – bla  
Abb. 7 – bla  
Abb. 8 – bla  
Abb. 9 – bla  
Abb. 10 – bla

# Tabellen

Tabelle 1 – Hardware  
Tabelle 2 – Notebook 1  
Tabelle 3 – Notebook 2  
Tabelle 4 – Software  
Tabelle 5 – Level-Definitionen

# 1. Projektbeschreibung

## 1.1 Vorstellung

Das Projekt wurde von Ambrosch Katharina und Kais Robert durchgeführt. Beide machen eine Ausbildung zum Fachinformatiker für Anwendungsentwicklung und sind im 3. Lehrjahr.

Ambrosch Katharina arbeitet bei der Flughafen München GmbH. Der gesamte Campus umfasst in etwa 30.000 Mitarbeiter, davon arbeiten ca. 250 in der IT der FMG. Der Flughafen wurde am jetzigen Standort am 17.05.1992 eröffnet. Der Geschäftsführer ist Dr. Michael Kerkloh. 2015 wurde der Flughafen „Franz Josef Strauß“ zum ersten 5-Sterne-Airport Europas gekürt. In Deutschland zählt er zum zweitgrößten Flughafen. 2016 wurde ein neuer Passagierrekord mit 42 Mio. Fluggästen erreicht.

Kais Robert arbeitet bei der com2 Communications & Security GmbH. Die Firma besteht aus 15 Personen, wovon acht der IT angehören. Die com2 existiert seit 1992 und setzt auf maßgeschneiderte, sichere IT-Kommunikationslösungen. Die Geschäftsführer sind Boehlke Axel und Kleibömer Thomas.

## 1.2 Projektumfeld

Das Projekt wurde an der staatlichen Berufsschule in Freising umgesetzt.

Zur Realisierung des Projektes benötigte Mittel:

|  |  |
| --- | --- |
| Hardware | Beschreibung |
| Motorola XT1072 (Android 6.0) | Zum regelmäßigen Testen der App |
| Notebook 1 (siehe Tabelle 2) | Zur Implementierung bei Katharina |
| Notebook 2 (siehe Tabelle 3) | Zur Implementierung bei Robert |

*Tabelle 1*

|  |  |
| --- | --- |
| Notebook 1 | Beschreibung |
| Prozessor | Intel Core i5-4310M CPU 2,70 GHz |
| Arbeitsspeicher | 8 GB |
| Betriebssystem | Windows 7 Enterprise SP 1 (64 Bit) |

*Tabelle 2*

|  |  |
| --- | --- |
| Notebook 2 | Beschreibung |
| Prozessor | Intel Core i7 CPU 1,73 GHz |
| Arbeitsspeicher | 4 GB |
| Betriebssystem | Windows 8.1 Pro (64 Bit) |

*Tabelle 3*

|  |  |
| --- | --- |
| Software | Beschreibung |
| Android Studio | Programm zur Android-Implementierung |
| Paint.net | Programm zur Erstellung der Grafiken |

*Tabelle 4*

Ansprechperson bei Fragen zur Implementierung / zu Android-Studio:

Albrecht Pfleiderer, Lehrer im Fach AEP

## 1.3 Projektziel

Hauptziel des Projekts ist eine funktionierende Spiele-App für das Android-Betriebssystem. Das Spiel ist eine Art Labyrinth, in dem man Kisten geschickt aus dem Weg schieben muss, um an den Zielpunkt zu gelangen. Verwirklicht wird dies im Zweispieler-Modus. Mit einem Login kann man sich in der App anmelden und bei Bedarf über das Netzwerk (mit zwei Smartphones) gegeneinander spielen.

## 1.4 IST-Zustand

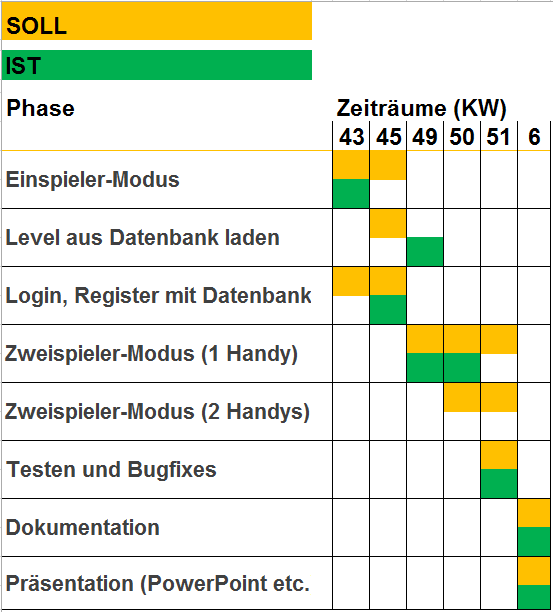
Derzeit steht kein Spiel als Android App zur Verfügung. Es sind bisher keine Programcodes dafür vorhanden.

## 1.5 SOLL-Zustand

Es soll eine funktionierende Spiele-App existieren. Diese soll mit einer Datenbank verknüpft sein, um einen Login bzw. eine Neuanmeldung zu ermöglichen und den Highscore zu speichern. Mit dem Zweispieler-Modus soll es möglich sein, gegeneinander über ein bzw. über zwei mobile Geräte spielen zu können.

# 2. Projektplanung

## 2.1 Zeitplanung



*Abb. 1*

# 3. Projektdurchführung

## 3.1 Erstellen der Level-Klasse

Jede Grafik im Level erhält eine Nummer, um das Layout richtig darstellen zu können. Folgende Definitionen wurden für die Darstellung eines Levels festgelegt.

|  |  |
| --- | --- |
| Nummer | Grafik |
| 0 | Boden |
| 1 | Mauer |
| 2 | Kiste |
| 3 | Startpunkt |
| 4 | Zielpunkt |

*Tabelle 5*

## 3.2 Erstellen der Main Activity

Bla

## 3.3 Erstellen der Grafiken

Alle Grafiken, die im Spiel zu sehen sind, werden selbst erstellt. Hierzu dient das Programm „paint.net“. Als Format eignet sich .PNG, da die Qualität besser ist, als beispielsweise bei .JPEG. In den Levels werden für die Mauern sowie für die zu verschiebenden Kisten Grafiken benötigt. Auch der Boden, der Zielpunkt und die Spielfigur müssen erstellt werden. Bei der Figur ist zu beachten, dass um sie herum die gleiche Farbe ist, die der Boden des Levels auch hat.



*Abb. x Abb. x*

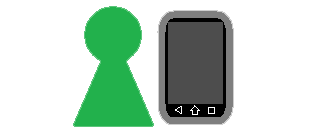
Die Richtungspfeile, welche seitlich bei den Levels zur Steuerung dienen, werden ohne weißer Hintergrundebene erstellt, damit keine farblichen Abweichungen mit dem Untergrund auftreten können.

Beim Login und bei der Auswahl des Spielmodus befindet sich oben ein Schriftzug, der den Namen des Spiels anzeigt. Dieser ist in der Schriftart „Showcard Gothic“ als Grafik abgespeichert mit weißem Hintergrund. Um Farbabweichungen zu vermeiden, wird bei der App-Oberfläche der Hintergrund auch auf weiß gestellt.

Für das App-Logo, das im Menü des Smartphones angezeigt wird, dienen die Anfangsbuchstaben „C“ und „M“ und die bereits erstellte Kiste. Hier darf es keine Hintergrundebene geben. So ist es möglich, dass das Bild überall platziert werden kann ohne farbliche Abweichungen. Auf folgender Abbildung wird dies erkennbar.



*Abb. x*

Die Buttons zur Auswahl des Spielmodus sind ebenfalls Grafiken ohne eine Hintergrundebene. Es gibt jeweils Spielfiguren und ein bzw. zwei Smartphone-Abbildungen. Diese Bilder werden mithilfe von Formen in paint.net erstellt. Beispielsweise besteht eine Spielfigur aus einem Kreis und einem darüber gelegten Dreieck.

*Abb. x*

## 3.4 Anzeigen der Grafiken im Spiel R

Bla

## 3.5 Erstellen der Datenbank-Klasse K

In diesem Projekt wird mit der SQLite-Datenbank gearbeitet. Zunächst wird eine eigene Klasse erstellt, die DatabaseManager genannt wird. Diese erbt von SQLiteOpenHelper. Am Anfang wird die Datenbank selbst erzeugt mit folgendem Befehl.

private SQLiteDatabase sqldb;

Dann werden Klassenvariablen deklariert und initialisiert. Beispielsweise benötigt man den Datenbanknamen.

private static final String DB\_NAME = "Crate\_Maze\_db";

Dieser, sowie die Datenbank-Version und die beiden Tabellen „player“ und „level“ werden als private deklariert. Zudem werden zwei SELECT-Befehle auf die Tabellen initialisiert. Der Aufbau ist gleich den privaten Klassenvariablen von oben, nur dass diese beiden jetzt public sind.

Anschließend werden die Spalten für die beiden Tabellen definiert. Für die „player“-Tabelle gibt es Spalten für die ID, den Namen, des Passworts und dem aktuellen Level. Bei der „level“-Tabelle benötigt man die ID, den Level-Aufbau, die Zeit und den Highscore. Bei den ID-Namen ist zu beachten, dass ein Unterstrich vorangeführt ist, da einige Klassen der Android-API dies so erwarten.

In der Methode onCreate werden die Tabellen in der Datenbank erstellt. Es werden jeweils ein String Player und ein String Level initialisiert. Der Aufbau anschließend ist wie bei MySQL.

CREATE TABLE <tabellenname> (<spaltenname> <datentyp>);

Nach der Erstellung werden die Tabellen mit der Methode execSQL() abgeschickt. Für die vollständige Methode siehe Anhang, Abb. **X**

Beim Aufruf der Methode close() wird die Datenbank geschlossen. Der Konstruktor wird von der Mutterklasse übernommen mit dem super-Befehl.

## 3.6 Erstellen des Logins und der Spielmodus-Auswahl R

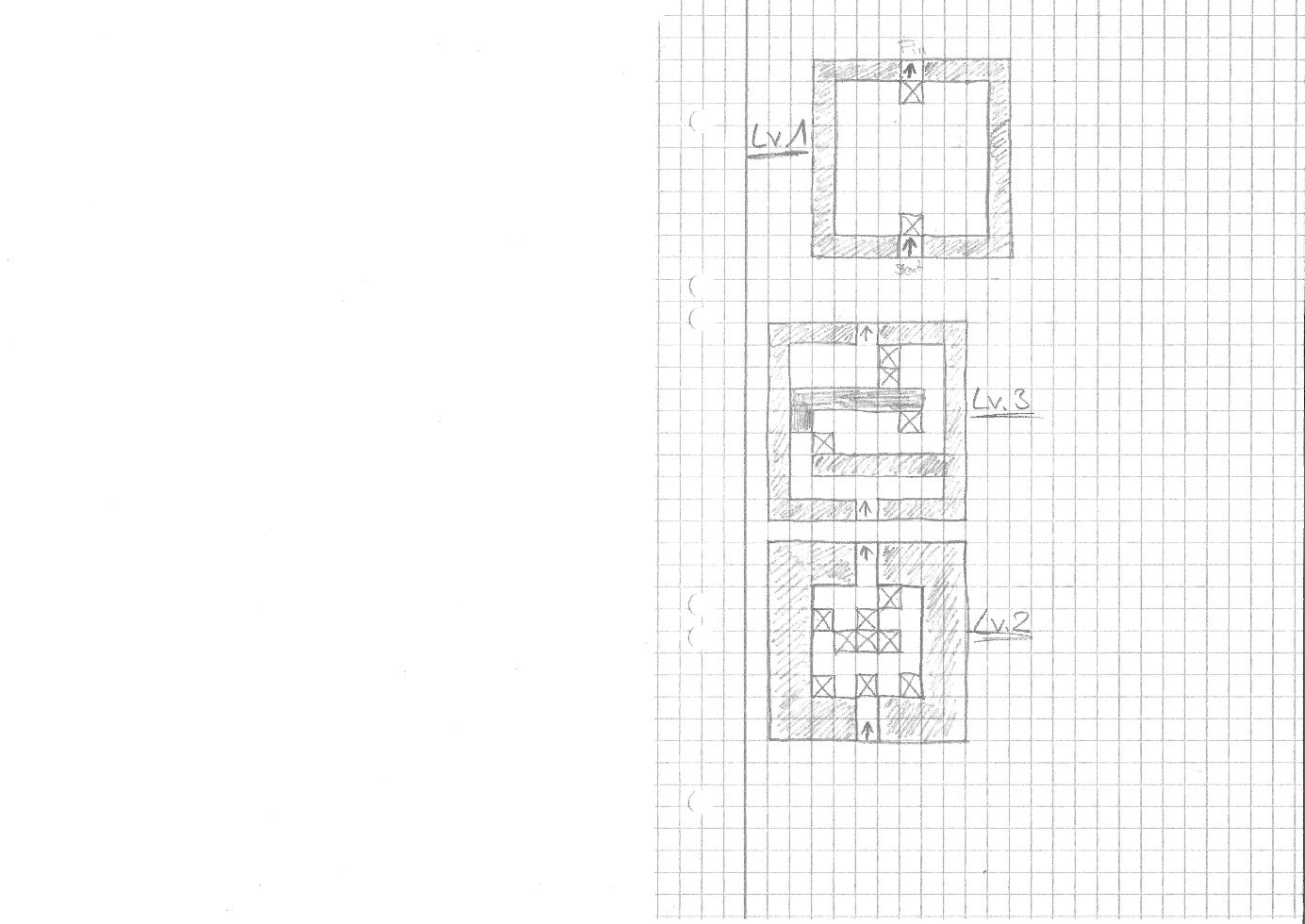
bla

## 3.7 Nutzen der Datenbank für Login R

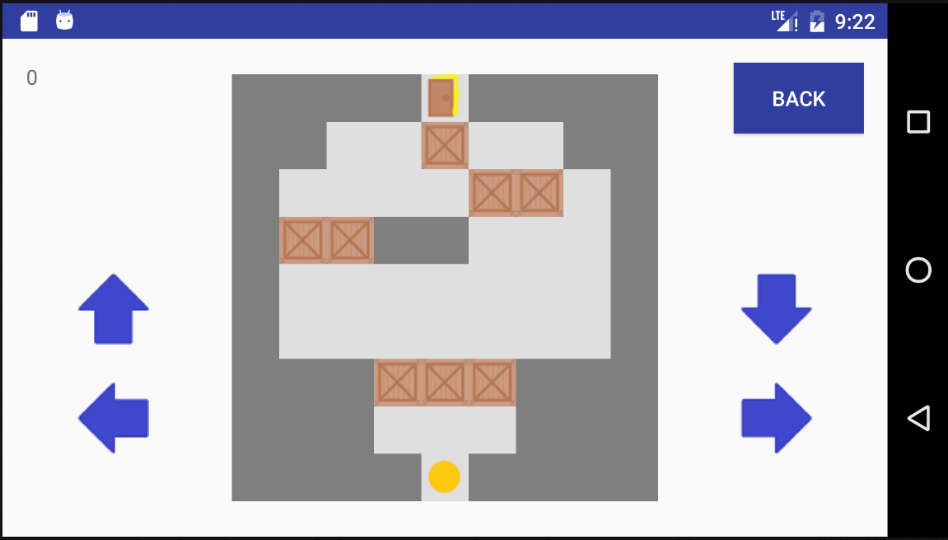
bla

## 3.8 Implementieren der Level-Auswahl RK

Im ersten Schritt werden die Levels auf einem Blatt skizziert. Wichtig hierbei ist, dass darauf geachtet wird, ein Level so zu gestalten, dass es am Ende nicht unmöglich zu lösen ist. Im besten Fall steigert sich der Schwierigkeitsgrad allmählich.



Die fertigen Levels werden in die Main-Klasse implementiert nach dem Muster, das im Punkt 3.1 genauer erläutert ist. In folgenden Abbildungen sieht man beispielsweise Level 5 aus dem Programmcode und daneben die dazugehörige Spieldarstellung.

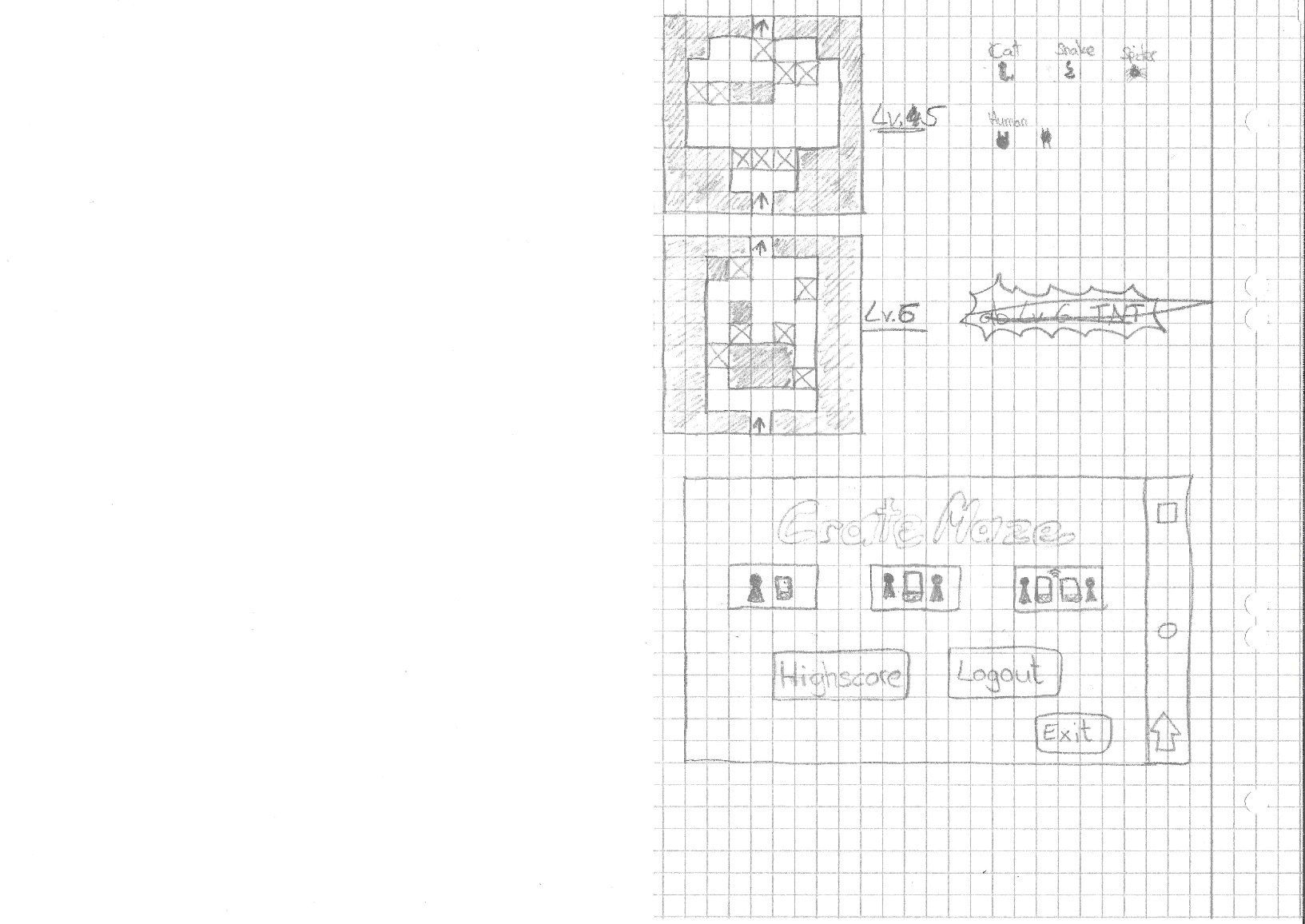
1 1 1 1 4 1 1 1 1   
1 1 0 0 2 0 0 1 1  
1 1 0 0 0 2 2 0 1  
1 2 2 1 1 0 0 0 1  
1 0 0 0 0 0 0 0 1  
1 0 0 0 0 0 0 0 1  
1 1 1 2 2 2 1 1 1  
1 1 1 0 0 0 1 1 1  
1 1 1 1 3 1 1 1 1  
 *Abb. x*

## 3.9 Hinzufügen der Zeitmessung R

bla

## 3.10 Anpassen des Layouts K

Sowohl für den Login-Bildschirm als auch für die Spielmodus-Auswahl werden zuerst auf einem Blatt Skizzen gefertigt, um besser planen zu können, wie diese Menüs gestaltet werden sollen.



## 3.11 Erweitern der Datenbank-Funktionalität RK

bla

## 3.12 Erweiterung der Spielmodus-Auswahl auf 2 Spieler R

bla

## 3.13 Implementieren der Highscore-Klasse RK

Die Klasse für den Highscore wird als Erstes erstellt.

# 4. Projektergebnis

## 4.1 Fehlerbehebung

### 4.1.1 Fortschritt wird nicht gespeichert R

Bla

### 4.1.2 Spiel stürzt nach Level 10 ab und verliert Fortschritt R

bla

### 4.1.3 Sonstige kleine Fehler

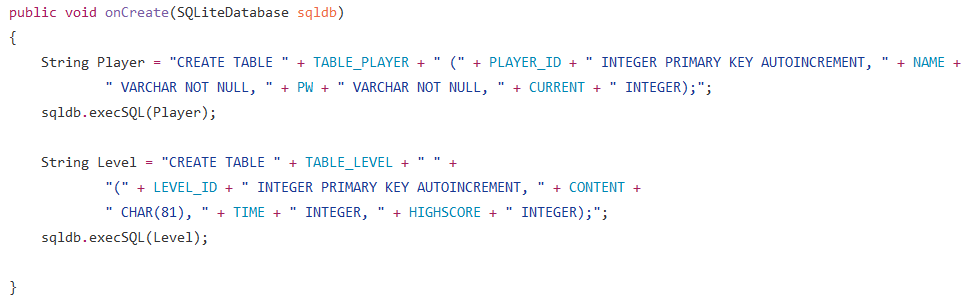
Wenn man einen Fehler in einem Level gemacht hat, kann man nicht neu anfangen, da kein „Zurück“-Button vorhanden ist. Dieser war bei der vorherigen Implementierung nicht vorgesehen. Der Button kann in dem Level-Layout hinzugefügt werden und muss anschließen in der Level Activity initialisiert werden und eine Funktion erhalten. Beim Drücken des Buttons kommt man nun zurück zur Level-Auswahl.

## 4.2 Abweichungen

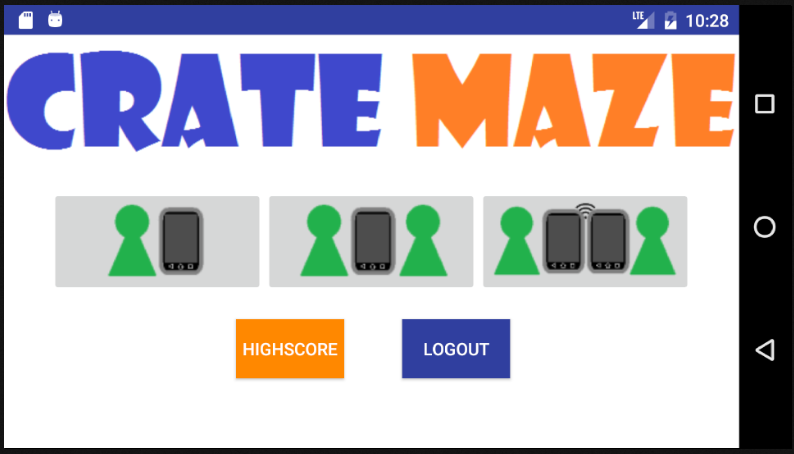
Der Zweispieler-Modus über zwei unterschiedliche Smartphones konnte in der vorgegebenen Zeit nicht mehr verwirklicht werden. Dafür wurde ein Highscore implementiert, der ursprünglich nicht in der Planung vorgekommen ist. Engpässe traten vor allem in der Kalenderwoche 51 auf, da ein Krankheitsfall eingetreten ist. Im Vergleich der IST- und SOLL-Zeit wird ersichtlich, dass die Planung ansonsten im Großen und Ganzen korrekt durchgeführt wurde. Abweichungen sind vor allem zu sehen, weil mehrere Aufgaben lediglich über einen ganzen Berufsschulblock geplant wurden und dadurch eine Woche früher beendet bzw. eine Woche später angefangen wurden. Das Laden der Levels aus der Datenbank konnte erst etwas später umgesetzt werden, da die Datenbank-Klasse unerwartete Fehler aufwies.

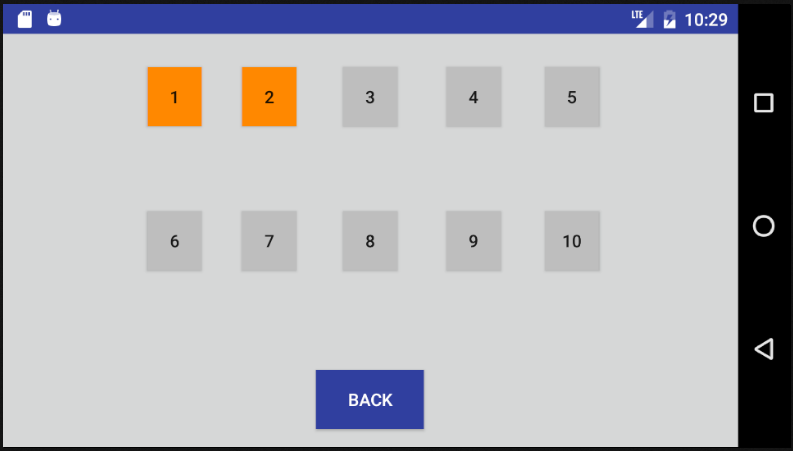
# 5. Anhang

## 5.1 Screenshots

  
*Abb. x*

  
*Abb. x*

  
*Abb. x*

  
*Abb. x*