

# **Technische Projektdokumentation**

Entwicklung mobiler Anwendungen  
Sommersemester 2020

Hochschule Worms  
Fachbereich Informatik  
Angewandte Informatik

Oliver Siebert 674899  
Melissa Sands 674721

## Inhaltsverzeichnis

DatenbankKlasse.kt.....	3
StoryDatenbank.kt .....	3
ShopDatenbank.kt .....	3
MusicService.kt .....	3
Menu.kt.....	3
StoryAuswahl.kt.....	4
Spielmechanik .....	4
Shop.kt .....	5
Einstellungen.kt .....	5
EinstellungenAuswahl.kt.....	5

## DatenbankKlasse.kt

In dieser Klasse wird eine Datenbank erstellt, die aus ID, Name, Geld und Musikan besteht. Geld gibt dabei den aktuellen Stand der verdienten Spielwährung an und Musikan gibt an, ob im Hintergrund Musik laufen soll oder nicht.

## StoryDatenbank.kt

Hier wird eine Datenbank mit vorgefertigten Levels erstellt. Diese bestehen aus dem Namen des Levels, ob das Level ausgewählt wurde, die einzelnen Felder des Levels, die als Lösung dienen und dem Wert „solved“, der angibt, ob das Level gelöst wurde. Anschließend werden alle bisher vorhandenen Level hinzugefügt.

## ShopDatenbank.kt

Die Shop Datenbank besteht aus zwei Tabellen: Color und Music. Sie dient dazu den Shop mit Farben und Musik zu füllen, die durch die Spielwährung zu erwerben sind. Die Tabelle Color besteht aus ID, Name, Preis, selected, gekauft, Hintergrund- und Schriftfarbe. Die Tabelle Music besteht aus ID, Name, Preis, selected, gekauft und source. Selected gibt dabei an, welche Farbe oder Musik ausgewählt wurde.

Anschließend werden die Werte der Tabelle Color durch Einfügen in die Datenbank hinzugefügt.

Für die Tabelle Music wird ebenso verfahren, allerdings werden die dafür benötigten mp3-Daten in dem Ordner „raw“ eingefügt.

## MusicService.kt

Hier wurden die Funktionen zum Starten, Pausieren und Stoppen der Musik eingefügt. Über die Funktion lesen() wurde aus der Datenbank heraus die mp3-Datei bestimmt, welche abgespielt werden soll.

## Menu.kt

Der erste Abschnitt dient dazu, dass die Musik abgespielt und ordnungsgemäß beendet wird, falls die Applikation beendet wird. Der zweite Abschnitt befasst sich mit der Erstellung der Activity. Es wird das Layout menu\_layout geladen und anschließend wird beim ersten Starten die Datenbank des Spielers (DatenbankKlasse.kt) initialisiert. Hier wird auch die Menge der Spielwährung ausgelesen.

Im Abschnitt „Musik spielen“ wird ausgelesen, ob die Auswahl „musikan“ getroffen wurde und spielt dementsprechend Musik. Andernfalls wird keine Musik abgespielt.

Im Abschnitt „Farbe auslesen“ wird aus der Shop Datenbank ausgelesen, welche Farbe ausgewählt ist und diese wird dann als Hintergrundfarbe übernommen.

Anschließend werden nur noch die Buttons definiert, die einen Seitenwechsel auslösen.

## StoryAuswahl.kt

Im ersten Abschnitt werden beim Erstellen der Activity die ersten sechs Felder (Level) der Story Datenbank (StoryDatenbank.kt) geladen. Danach wird per Knopfdruck definiert, dass bei Auswahl eines der Level die Story5x5-Activity geladen wird. Hierbei wird in der Datenbank der Parameter „selected“ geändert. Anschließend wird die Hintergrundfarbe, wie im Menü, ausgelesen. Die Vor- und Nach-Buttons dienen dazu, durch das Menü zu navigieren.

## Spielmechanik

Die Activities Story5x5, Farb5x5, Zeit5x5 und Zeit3x3 haben folgende gemeinsame Funktionen:

Die Felder sind so definiert, dass sie per Knopfdruck die Farbe ändern. Anschließend gibt es die Funktion generateNewGame(). Hier werden Zufallsdaten in ein Lösungsarray geschrieben. Danach werden die Buttons definiert, womit man die Felder füllen oder mit einem X markieren kann. Beim Farb5x5 hat man hier die Möglichkeit unterschiedliche Farben auszuwählen.

Nachfolgend werden die Unterschiede der Activities Story5x5, Farb5x5, Zeit5x5 und Zeit3x3 aufgeführt:

Bei der Activity Story5x5 wird das Solution Array mit Lösungswerten aus der Datenbank gefüllt. Anschließend wird das Spiel geladen, d. h. die hellblauen Randfelder werden anhand des Lösungsarrays generiert. Bei jedem Feld, das ausgewählt/markiert wird, wird im Hintergrund überprüft, ob die Auswahl korrekt ist. Hier ist die Besonderheit, dass bei der Überprüfung die direkten Felder des Spielfelds mit dem Lösungsarray verglichen werden. Die Funktion pruefe() überprüft, ob die Bedingungen der Randfelder erfüllt wurden und färbt diese gegebenenfalls grün.

Der Spielmodus Münzmodus ist in den Dateien Zeit3x3 und Zeit5x5 beschrieben. Hier wird zu Beginn das Solution Array mit den Zufallswerten 1 oder 0 gefüllt. Anschließend werden die Randfelder dementsprechend beschriftet. Wie beim Storymodus gibt es hier auch einen Button, der die Einstellung wechselt, ob die Felder ausgefüllt oder mit einem X markiert werden. Die Funktion checkWin() ruft die

Funktion `pruefe()` auf, welche überprüft, ob die Bedingungen der Randfelder erfüllt wurden, da hier mehrere Lösungen möglich sind.

Beim Farbmodus wird zu Beginn das Solution Array auch mit Zufallswerten gefüllt, allerdings mit Werten von 0 bis 3. Anschließend wird die Funktion `beschrifte()` aufgerufen. Hier werden die Randfelder mit der Farbe und entsprechender Anzahl der Felder beschriftet. In der Funktion `checkWin()` wird wieder die Funktion `pruefe()` aufgerufen, die auch überprüft, ob die Bedingungen der Randfelder erfüllt wurden, da es auch hier wieder mehrere Lösungen gibt.

## Shop.kt

Hier wird zu Beginn `generateItems()` aufgerufen, bei dem zwei Items für die Hintergrundfarbe generiert wird. Dabei definiert „colorMax“ die höchste Anzahl an IDs in der Color Datenbank. Anschließend wird ein Item aus der Musik Datenbank generiert. Auch hier wird mit „musicMax“ die höchste Anzahl an IDs in der Musik Datenbank definiert. Daraufhin werden die Werte aus der Datenbank ausgelesen. Wenn etwas bereits gekauft wurde, wird es grau hinterlegt, andernfalls wird Name und Preis angegeben. Bei den Farbelementen passt sich der dazugehörige Hintergrund der Farbe an. Wird ein graues Feld angeklickt, erscheint der Toast „Element bereits gekauft“. Bei Klick auf ein farbiges Feld erscheint ein Dialogfenster, in dem man den Kauf bestätigen kann. Wenn man einen Kauf tätigt, wird die Funktion `kaufen()` aufgerufen. Hier wird zuerst überprüft, ob genug Geld vorhanden ist, um das Element zu kaufen. Wenn ja, wird das Geld abgezogen und das Element wird in der Shop Datenbank als gekauft gekennzeichnet.

## Einstellungen.kt

Zu Beginn wird aus der Spielerdatenbank ausgelesen, ob die Musik gerade spielt oder nicht. Dann werden dementsprechend der Hintergrund und die Beschriftung des Buttons (`music_switch`) angepasst. Der Farb- und der Musikbutton rufen die neue Seite `EinstellungenAuswahl.kt` auf und geben in ihrem Intent den Namen ihrer Datenbank mit.

## EinstellungenAuswahl.kt

Als erstes wird der Datenbankname, der mitgegeben wurde, gespeichert. Anschließend werden die Daten mit der Funktion `loadValues()` geladen. Hier wird aus der Shop Datenbank die Tabelle ausgelesen, die mit dem String übergeben wurde. Zunächst werden alle Felder mit dem String „----“ gefüllt und mit einem grauen Hintergrund hinterlegt. Anschließend werden aus der ausgewählten Datenbank die Daten ausgelesen. Falls ein Element als gekauft gekennzeichnet wurde, wird sein Name angezeigt und die Hintergrundfarbe angepasst. Wenn die Color Tabelle

aufgerufen wurde, werden die Elemente mit ihrer eigenen Hintergrundfarbe hinterlegt. Wenn die Music Tabelle aufgerufen wird, wird eine Standard Hintergrundfarbe ausgewählt. Wenn ein Musiktitel den Wert true bei selected hat, dann wird die Hintergrundfarbe auf grün gesetzt. Bei der Farbe bleibt hier der ursprüngliche Hintergrund erhalten. Wenn das Element nicht gekauft wurde, wird der String „????“ ausgegeben.

Mit den Buttons Vor und Nach kann man die nächsten oder vorherigen sechs Elemente anzeigen lassen.