# Memory - Dokumentation

A - Team
Technische Hochschule Mittelhessen

11. September 2009

## Inhaltsverzeichnis



# 1 Prolog

Die Gruppe A besteht aus Markus Kretsch, Frank Kevin Zey, Florian Thomas und Hagen Lauer.

## 1.1 Anforderungen

Muss

- Memory-Spielfeld mit sinnvoller Größe (z.B. 8x8 Karten/Felder) für 2 bis 6 Spieler.
- Jeder Spieler bekommt einen Namen, der über Spielsitzungen hinweg gespeichert wird.
- Über jeden Spieler wird eine Statistik angezeigt, wie z.B. Anzahl gewonnener verlorener Spiele, oder Anzahl richtiger Treffer pro 100 Züge.

Kann

- Mehrspielermodus über mehrere Smartphones innerhalb eines LANs.
- Weitere Spielkarten können z.B. von der SD-Karte nachinstalliert werden.

### 1.2 Lösung / Idee

- Bilder sollen mit Grid und Imageview dargestellt werden. Dabei bieten diese gute Möglichkeiten Klicks zu erkennen und entsprechend zu behandeln.
- Für die Spieler wird eine SQLite Datenbank verwendet.
- Wir haben gute Bibliotheken gefunden um die Daten der Spieler wie gewünscht statistisch auszuwerten und darzustellen.
- Netzwerkspiele werden über WiFi und JavaSockets realisiert, dabei soll es einen Host und mehrere Clients pro Sitzung geben. Das Spielsystem muss also die entsprechende Flexibilität für lokale und Netzwerkspiele mitbringen.
- Spielkarten sollen per .zip File von der SD Karte des Geräts nachladbar sein. Bilder werden in einer Datenbank gespeichert. Das Spiel lädt die Bilder für das Spielfeld aus der Datenbank.

## 2 Architektur

Wir haben uns selbst als Ziel gesetzt, dass wir in 2 Richtungen entwickeln: Das Spiel Memory als sehr spezifische Implementierung und ein "Framework", das alle typischen Funktionen für ein rundenbasiertes Spiel mitbringt. So konnten wir mit entsprechenden Oberklassen (Game.java) und abgeleiteten Klassen (z.B. Memory.java) garantieren, dass am Ende beide Zweige zusammen führen. Zum Framework gehört Game.java als Oberklasse aller implementierbaren rundenbasierten Spiele, eine Engine die im Wesentlichen Datenbankzugriffe kapselt, ein einfaches Menüsystem, das leicht konfigurier- und erweiterbar ist und einen statistischen Teil der mit der Datenbank zusammen arbeitet. Wir haben dann als Konkrete Implementierung für Memory die Klasse Memory.java von Game erben lassen und entsprechend für Memory angepasst. Ebenso wird die Netzwerkimplementierung eine von Game abgeleitete Klasse sein, natürlich noch mit einigen spezifischen Anpassungen zur Kommunikation der Geräte.

### 2.1 Design

Wie in Abbildung 1 zu sehen ist entsteht das Framework völlig unabhängig vom zu implementierenden Spiel.

### 2.2 Engine / Datenbanken

### 2.3 Engine und Datenbank

Die Engine umfasst die Typisierung der Spieler und Datenbankschnittstellen.

#### 2.3.1 Engine

#### 2.3.2 Datenbank

### 2.4 Statistik

crunch

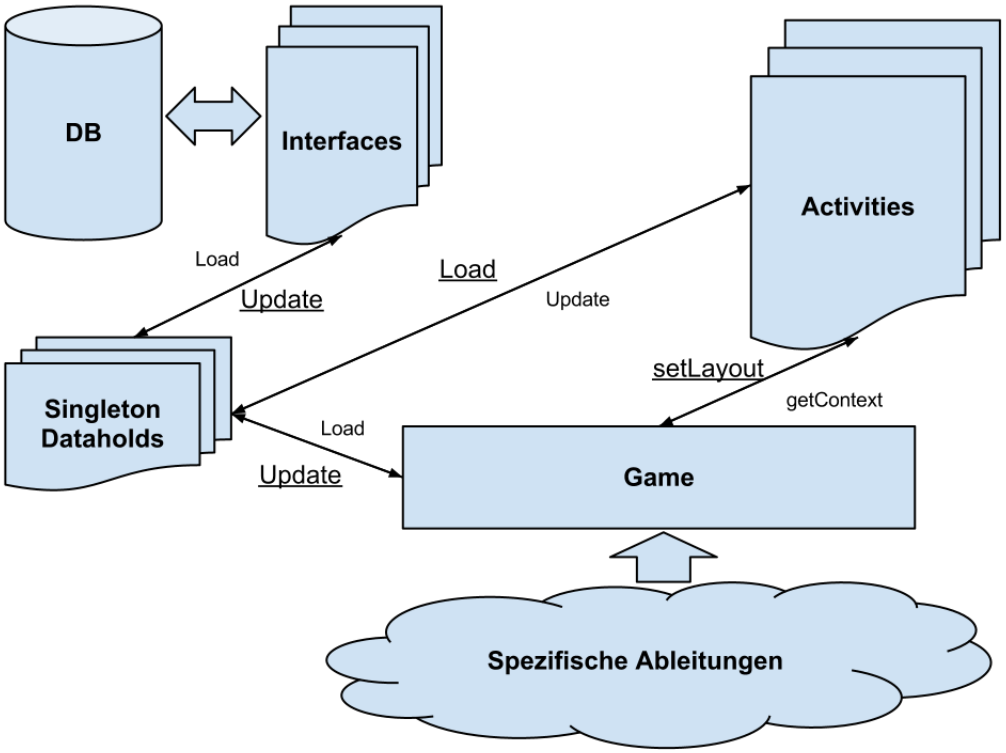


Abbildung 1: Konzept

## 2.5 Memory

markus

## 2.6 Netzwerk

crunch

### Benutzerdokumentation

Die Benutzerdokumentation wird per *Help* in der App bereitgestellt.

### API Dokumentation

Die Code Dokumentation ist per JavaDoc im Quelltext abgewickelt.