

# Projektarbeit 1

## Erweiterungen des Memory-Spiels



## Allgemeine Informationen

In dieser Aufgabe geht es darum, das bestehende Memory-Spiel um weitere Funktionen zu ergänzen. Hierzu ist es notwendig, dass Sie selbständig nach geeigneten Umsetzungsvarianten recherchieren und sich für die in Ihren Augen geeignetste entscheiden. Bevor Sie die Arbeit einreichen, testen Sie diese bitte auf verschiedenen Simulatoren oder falls vorhanden auch physischen iOS-Geräten. Eventuell geschriebene Tests werden nicht bewertet, dürfen aber selbstverständlich umgesetzt werden.

Offene Fragen werden bei Relevanz für die Gruppe per E-Mail an alle Teilnehmer der Vorlesung beantwortet.

**Spätester Abgabetermin:** 19.11.2020 12:00

## **Bewertungskriterien**

- Einhaltung der Kriterien, die in der ersten Stunde genannt wurden
- Erfüllung der Anforderungen aus Aufgabenstellung
- Einhalten des MVVM-Patterns
- Einhalten von Clean-Code-Prinzipen
- Fehlerfreiheit (Code kompiliert ohne vermeidbare Warnungen, App crasht nicht)
- Abgabe kompatibel mit Xcode 12, iOS 14
- Ausschliessliche Verwendung von SwiftUI
- Keine Verwendung von Third-Party-Frameworks

## Aufgabe 1: Menü

- Erstellen Sie ein Menü, in dem der Nutzer zwischen verschiedene Spieltypen wählen kann, die im Folgenden implementiert werden: «Emoji», «Fotos», «Kontakte».
- Das Menü soll angezeigt werden, sobald der Nutzer ein neues Spiel auswählt, ausserdem initial beim Start der App.
- Zur Darstellung des Menüs bietet sich ein Sheet an, wie unter <https://www.hackingwithswift.com/quick-start/swiftui/how-to-present-a-new-view-using-sheets> gezeigt (optional).

## Aufgabe 2: Schwierigkeitsgrad

- Erweitern Sie das Menü um den Schwierigkeitsgrad des Spiels einstellen zu können: «Leicht», «Mittel», «Schwer».
- Passen Sie das Spiel so an, dass je nach Bildschirmgrösse eine sinnvolle Menge an Karten in das Spiel je nach Schwierigkeitsgrad angezeigt werden.
  - Auf einem iPhone SE lassen sich weniger Karten anzeigen als auf einem iPad mit 12.9 Zoll, welches hier berücksichtigt werden soll.
  - Entscheiden Sie sich hier für eine sinnvolle Anzahl an Karten und für eine geeignete Darstellung der Auswahlmöglichkeit.

### Aufgabe 3: Emoji-Sammlungen

- Erweitern Sie das Spiel um thematisch zusammenhängende Emoji-Sammlungen. Z.B. Natur, Smileys, Nahrungsmittel, etc.
- Jede Sammlung besteht aus einer Liste von Emojis, die zufällig ausgewählt auf die Karten des Spiels aufgeteilt werden.
- Wählt der Nutzer den Kartentyp Emoji, wird eine dieser Sammlungen verwendet. Die Anzahl der Emojis muss die unter Aufgabe 2 implementierte Kartenanzahl abdecken.

## Aufgabe 4: Punkte-System

- Implementieren Sie ein Punktesystem und speichern Sie den Highscore auf dem Gerät.
  - Der aktuelle Punktestand soll im Spiel sichtbar sein und sich beim Lösen von Kartenpaaren aktualisieren.
  - Hierzu soll auch die in der Vorlesung implementierte «Bonustime» berücksichtigt werden, falls Paare schnell genug gefunden werden.

## Aufgabe 5: Bilder

- Erweitern Sie das Spiel um den Spieltyp «Bilder». Hierzu sollen die Spiel-Karten Bilder darstellen.
- Bis zur 8. Vorlesung wurde nicht behandelt, wie Bilder aus dem Internet geladen werden können, daher gilt die Aufgabe als voll erfüllt, wenn Bilder aus dem Asset Catalog (Assets.xcassets) verwendet werden, den Sie mit entsprechenden Fotos anreichern müssen. Als Bildquelle sind Fotos von <https://unsplash.com/> zugelassen.
- Sonderpunkte können gesammelt werden, wenn als Datenquelle das REST-API von <https://unsplash.com/developers> verwendet wird (optional). Anleitungen hierzu finden Sie zahlreich im Internet.
  - Mit diesen Sonderpunkten lassen sich Abzüge an anderer Stelle ausgleichen



## Aufgabe 6: Kontakt-Fotos

- Erweitern Sie das Spiel um den Spieltyp «Kontakte». Hierzu sollen Profilbilder aus dem eigenen Adressbuch zufällig ausgewählt werden.
- Zum Zugriff auf das Adressbuch verwenden Sie bitte das Contacts Framework von Apple:  
<https://developer.apple.com/documentation/contacts>.
- Sofern Sie über kein iOS-Geräte mit Profilfotos verfügen, auf die Sie zugreifen können, können Sie auch problemlos den Simulator verwenden. Dieser verfügt bereits über Dummy-Kontakte, denen Sie Bilder zuweisen können. Per Drag-Drop lassen sich Fotos in den Simulator kopieren und sind dann zur Auswahl des Profilbildes verfügbar.
- Die Dokumentation von Apple ist ein erster Einstiegspunkt, recherchieren müssen Sie noch, wie Sie beispielsweise um Erlaubnis des Nutzers bitten, auf die Kontakte zugreifen zu dürfen.
- Beachten Sie bitte, dass das Auslesen der Fotos eine Weile dauern kann. Das UI sollte in dieser Zeit nicht blockieren und weiter bedienbar bleiben.