

Swift mit Xcode

Ein Benutzerhandbuch für die Anfänger.



Projekt im Basislehrjahr

1 Semester

15.12.22 – 13.01.23

Adligenswil, 04.01.2023

Eingereicht bei: Urs Nussbaumer

Allgemein

Mindestanforderungen

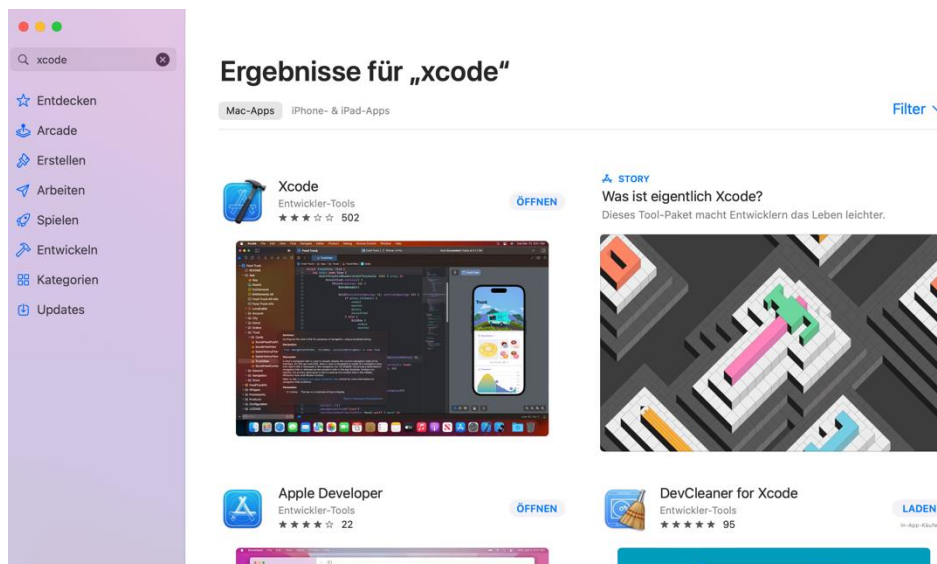
Du startest neu mit Xcode & Swift. Bevor du die Programmiersprache erlernst, empfehle ich dir «Allgemein» durchzulesen. Wichtige Grundelemente sind enthalten.

Um mit Swift & Xcode arbeiten zu können brauchst du mehrere Anforderungen.

- Mindestens MacOS BigSur installiert haben.
- Um aktuelle Software zu entwickeln, solltest du mindestens ein Mac von 2015 haben.
- Ein Mac mit noch mindestens 100 Gigabyte offenen Speicher.

Wenn dein Mac diese Anforderungen erfüllt, kann du beginnen!

Lade dazu Xcode vom App Store hinunter. Der Download könnte einige Zeit gehen. Eine stabile Internetverbindung ist von Vorteil.



Während dem Download wirst du gefragt für welches Gerät (Mac, TV, iPad, iPhone) du die Installation durchführen möchtest. Wähle eines oder mehrere aus um für das Gerät zu entwickeln.

Was ist nun der Unterschied zwischen Swift & Xcode?

Um Programme zu entwickeln, brauchst du nur Xcode. Xcode ist die Entwicklungsumgebung, wo du mit der Programmiersprache Swift, C++ oder Python entwickeln kannst.

Das heisst Swift ist in Xcode enthalten.

Xcode beinhaltet eine Reihe von Werkzeugen & Funktionen. Diese erleichtern dir das Entwickeln. Es beinhaltet ein Editor zum Schreiben von Code, ein Debugger, ein Simulator zum Testen von Anwendungen auf verschiedenen Geräten und Betriebssystemen und viele andere.

Swift ist eine moderne, sichere und schnelle Programmiersprache von Apple, die es Entwicklern ermöglicht, Anwendungen für macOS, iOS, iPadOS, watchOS und tvOS zu erstellen. Es wurde 2014 eingeführt und hat schnell an Popularität gewonnen, da es einfacher zu lesen und zu schreiben ist als Objective-C, die frühere Hauptsprache von Apple für die Entwicklung von iOS- und macOS-Anwendungen.

Swift ist eine statisch typisierte Sprache, was bedeutet, dass Variablen und Konstanten einen bestimmten Datentyp haben müssen, bevor sie verwendet werden können. Es unterstützt auch die Verwendung von optionals, die es ermöglichen, dass Variablen und Konstanten entweder einen Wert oder "nil" (keinen Wert) enthalten können.

Swift hat auch erweiterte Funktionen wie Closures, Generics und Enumerations. Es ist auch sicherer als Objective-C, da es automatisch verhindert, dass bestimmte Arten von Programmierfehlern auftreten, die in Objective-C häufig sind, wie zum Beispiel überflüssige Nullzeiger.

Es gibt auch eine Open-Source-Version von Swift, die es Entwicklern ermöglicht, auf einer Vielzahl von Plattformen wie Linux und Windows zu arbeiten, und die auch für andere Zwecke als die Entwicklung von Apple-Anwendungen verwendet werden kann.

- + leicht zu lesen und zu verstehen: Swift hat eine einfache und klare Syntax, die für Einsteiger und erfahrene Entwickler gleichermaßen leicht zu lesen und zu verstehen ist.
- + sicherer: Swift enthält viele Sicherheitsfunktionen, die dazu beitragen, Fehler und Sicherheitslücken in Ihrem Code zu vermeiden. Zum Beispiel ist Swift type-safe, was bedeutet, dass jedes Objekt in Swift einem bestimmten Datentyp zugeordnet ist, was dazu beiträgt, Typfehler zu vermeiden.
- + schnell: Swift ist eine schnelle Programmiersprache, die dazu beitragen kann, dass Ihre Anwendungen schneller laufen.
- + offen: Swift ist eine offene Quelle Sprache, was bedeutet, dass jeder den Quellcode einsehen und verbessern kann.
- Noch neu: Swift ist im Vergleich zu älteren Sprachen wie Objective-C noch neu und es gibt weniger Ressourcen und Bibliotheken, auf die Entwickler zurückgreifen können.
- möglicherweise nicht so weit verbreitet: Da Swift noch eine neue Sprache ist, wird sie möglicherweise nicht von allen Unternehmen und Entwicklern unterstützt.
- möglicherweise nicht so gut unterstützt: Da Swift noch eine neue Sprache ist, gibt es möglicherweise noch nicht die gleiche Unterstützung von Drittanbietern und Tools wie bei älteren Sprachen.

Xcode

Swift & Swift UI

Allgemeine Erklärung

Emojirama (eigene App)

Was habe ich gemacht?

Als Projekt nahm ich mir vor eine eigene App für das iPhone zu erstellen. Dies wollte ich mit der offiziellen Software von Apple erstellen. Mit der Programmiersprache Swift & dem Programm Xcode. Zuerst lernte ich wie die Programmiersprache Swift funktioniert & wie man mit ihr startet. Ich schaute 2 Tage lang Videos & recherchierte, um wichtiges genauer zu wissen. Ich lernte sehr viel dazu. Zudem lernte ich wie Xcode funktioniert & wie man es bedient. Xcode ist ein sehr vielseitiges & komplexes Programm.

Nachdem ich viel gelernt habe, begann ich mit meinen ersten kleinen Apps um zu trainieren. Ich erstellte eine App mit einer Weltkarte, eine App wo man die Uhrzeiten auf der Welt sieht, & eine mit wenig Sinn. Dort trainierte ich. Zudem habe ich die Apps probiert auf mein iPhone zu laden. Dies war sehr anspruchsvoll & kostete viel Zeit. Es braucht Geduld.

Danach startete ich mit dem Storyboard auf Trello. Ich musste zuerst aber überlegen, was ich erstellen möchte & was ich mit dem Erlernten erstellen kann. Zudem recherchierte ich im Internet. Ich kam zum Schluss ein Memory zu erstellen. Dies sollte aus Schwierigkeitsstufen bestehen. Zudem sollten nach jedem Spiel neue generiert werden. Alles sollte animiert & benutzerfreundlich sein. Nachdem man verloren oder gewonnen hat sollte auch eine Seite angezeigt werden, ob man gewonnen oder verloren hat. Zudem sollte man auch noch wählen können, ob man zum Homescreen zurück möchte oder das Spiel wiederholen möchte. Das das Spiel schwieriger wird sollte auch ein Zeitlimit sein. Somit hat man nur begrenzt Zeit.

Schlussendlich konnte ich dies alles umsetzen & was sehr zufrieden.

Was ging gut & was nicht?

Ich habe meine methodischen Ziele vollkommen erreicht & bin sehr zufrieden mit mir. Ich kam meistens gut voran & habe jeden Tag viel erreicht. Somit musste ich mich immer konzentrieren & mich auf meine Arbeit fokussieren. Zudem habe ich viel neues gelernt & die Zeit immer gut genutzt.

Meine Projektziele vom Trello habe ich auch zu 90% erfüllt. Alle Ziele wo «must» sind habe ich vollumfänglich erfüllt & sind in der App sichtbar. Die «should» habe ich auch ganz erreicht. Die «nice to have» habe ich nicht ganz geschafft. Einige sind trotzdem abgeschlossen. Andere angefangen oder noch nicht bearbeitet. Dies finde ich nicht schlimm. Ich bin immer noch sehr zufrieden.

Beim Erstellen der App (Spiel) kam ich meistens gut voran. Meine grosse Vorbereitung hat sich sehr gelohnt. Mit der Vorbereitung konnte ich viele Probleme verhindern. Trotzdem gab es schwierigere Momente. Das Animieren der Karten war sehr aufwendig. Zudem war das Erstellen der Karten schwierig. Es war sehr aufwendig den Cod zu schreiben, sodass die Karten auch funktionieren. Dazu gab es aber viele Videos auf YouTube.



Homescreen

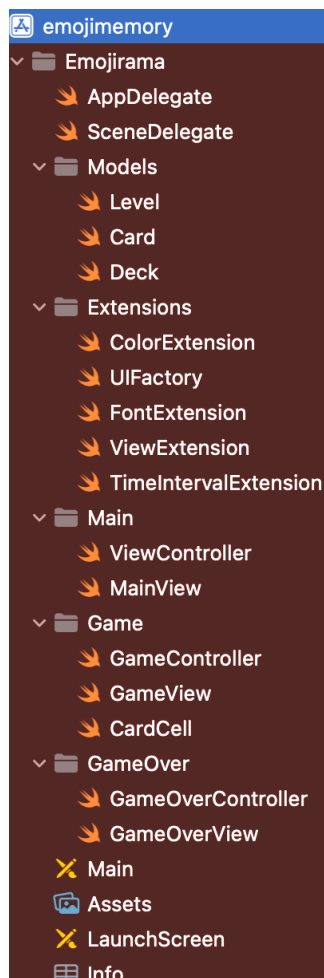


Gameplay



Losescreen

Wie ist der Cod aufgebaut?



Links sieht man die verschiedenen Dateien. Diese zusammen bilden die ganze App.

Ich erstelle ein Ordner für das Game selbst & deren Benutzeroberfläche, für das Gameplay oder auch für, wenn man das Spiel verloren & gewonnen hat.

Alles ist mit Variablen verbunden & funktioniert nur so.

Unter dem LaunchScreen (2 von unten) kann man den Ladevorgang selbst gestalten, sodass es schön aussieht. Ich entschied mich für ein Zahnrad.

Unter Assets kann man noch für alle Optionen das Bild für die App wählen.

Unter Models habe ich zudem noch die Level & Schwierigkeit Optionen erstellt.

Anhang

Videos

Trello Storyboard

<https://trello.com/b/2llbgqVJ/blj-abschlussprojekt>