

## Überblick über die Materialien zu „Algorithmisches Problemlösen mithilfe des App-Inventors“ sowie didaktische Überlegungen

### Lehrermaterial

Im Folgenden werden die Materialien zu „Algorithmisches Problemlösen mit dem App-Inventor“ beschrieben sowie jeweils zugehörige didaktische Überlegungen formuliert.

Als Werkzeug wird der App Inventor<sup>1</sup> verwendet. Mit dieser kostenlosen Software können blockbasiert eigene Apps entwickelt werden.

### Ausführen und Testen der selbst erstellten Materialien

Die selbst programmierten Apps können in Echtzeit auf mobilen Geräten wie Smartphones oder Tablets getestet werden. Entsprechende Companion Apps (siehe <https://appinventor.mit.edu/explore/ai2/setup> - Link vom 03.06.2021), die die Ausführung der selbst erstellten Apps in einer geschützten Umgebung ermöglichen, gibt es sowohl für Android- als auch iOS-Systeme. Zum Zeitpunkt der Erstellung dieser Materialien ist die iOS-Variante relativ neu, insbesondere die Ausführung von Apps beispielsweise mit geräteeigenen Beschleunigungssensoren oder die Darstellung komplexerer Layouts funktionieren daher noch nicht ganz zufriedenstellend. Da die Werkzeuge kontinuierlich weiterentwickelt werden ist hier in der nächsten Zeit mit Verbesserungen zu rechnen. Das Testen auf Android-Systemen ist davon nicht betroffen und funktionierte zum Zeitpunkt der Erstellung der Materialien problemlos.

### Inhaltliche und technische Voraussetzungen

Die vorliegenden Materialien richten sich an Lernende, die bereits erste Erfahrungen im algorithmischen Problemlösen in einer blockbasierten Sprache wie beispielsweise Scratch oder Snap! sammeln konnten. Sie können auch in Lerngruppen ohne Programmiererfahrungen eingesetzt werden, dann müssen evtl. an gegebenen Stellen im Unterricht Aspekte zur Programmierung vertieft werden.

Die Implementierung erfolgt im App Inventor, wahlweise unter Nutzung eines Google-Accounts mit der Möglichkeit der Speicherung der eigenen Projekte unter <http://ai2.appinventor.mit.edu> (Zugriff vom 15.01.2020) oder anonym unter <http://code.appinventor.mit.edu/login/> (Zugriff vom 15.01.2020).

Die selbst entwickelten Apps können in einem Emulator getestet werden (hierzu muss eine entsprechende Software auf dem Rechner installiert werden). Eine Alternative und sicherlich für Lernende motivierender ist das Testen der Apps in Echtzeit auf einem Smartphone oder Tablet. Erklärungen hierzu findet man beispielsweise unter <https://appinventor.mit.edu/explore/get-started> (Zugriff vom 15.01.2020).

---

<sup>1</sup> <https://appinventor.mit.edu/> (Zugriff vom 15.01.2020)

Nutzungsbedingungen vgl. <https://appinventor.mit.edu/about/termservice> (Zugriff vom 15.01.2020)

## Mögliche Einstiegshürden zur Erstellung eigener Apps

Die Entwicklung eigener Apps mit dem App Inventor ist trotz der blockbasierten Programmierung nicht ganz so intuitiv wie das Programmieren mit anderen blockbasierten Sprachen wie Scratch. Folgende Konzepte müssen daher von den Schülerinnen und Schülern verinnerlicht werden, damit sie sich in dem Werkzeug zurechtfinden:

- *Kennenlernen des Werkzeugs*: Im App-Inventor gibt es zwei verschiedene Modi, den Designer-Modus und den Blöcke-Modus. Diese trennen die Gestaltung einer App von der Implementierung ihrer Funktionalitäten. Dabei ist insbesondere für Lernende, die bisher nur blockbasierte Programmiersprachen wie Scratch oder Snap verwendet haben, die Gestaltung einer App im Designer-Modus neu.
- *Unterschiede zu Scratch und Snap!*: die farbliche Gestaltung und die Formen der einzelnen Blöcke im AppInventor ähneln denen in Scratch und Snap!. Jedoch gibt es zusätzlich für jedes Objekt (wie beispielsweise Buttons, Labels, ...) weitere Blöcke. So hat beispielsweise ein Objekt `Button` Blöcke, die nur für Buttons Sinn machen. Dieses Konzept der lokalen Eigenschaften und Methoden ist möglicherweise neu.
- *Testen der selbst erstellten Apps über die Companion App oder den Emulator*: auch hier muss das Testen selbst erstellter Apps erlernt werden.

Die Materialien sind daher so angelegt, dass sofort erste Apps programmiert und getestet werden, um dadurch die grundsätzliche Verwendung des App Inventors kennenzulernen. Auf einen vollständigen Überblick der vielfältigen Möglichkeiten des Werkzeugs wird dagegen verzichtet. Die unterschiedlichen Komponenten und ihre Verwendung können bei Bedarf von den Lernenden im Anschluss eigenständig erkundet werden.

## Spracheinstellungen

Man kann im App Inventor auswählen, in welcher Sprache die Komponenten und Blöcke angezeigt werden sollen. Dabei ist die Übersetzung vom Englischen ins Deutsche möglicherweise nicht immer gelungen, inzwischen (Stand Mai 2021) aber insgesamt schon sehr weit fortgeschritten. Je nach Lerngruppe ist trotzdem abzuwägen, ob man eine Spracheinstellung empfiehlt oder die Auswahl den Lernenden selbst überlässt.

## Überblick über die Materialien

### 01\_App-Inventor

Dieses Dokument soll lediglich als Hintergrundwissen fungieren und einen kurzen Überblick über die Benutzeroberfläche ermöglichen. Es bietet sich an, die aufgeführten Menüpunkte im Unterrichtsgespräch jeweils genau dann zu thematisieren, wenn sie relevant werden. So möchte man möglicherweise zunächst im Unterricht eine eigene Oberfläche gestalten und stellt deswegen die Palette (3), den Betrachter (4) und das Eigenschaften-Fenster (6) vor, verzichtet aber vielleicht zunächst auf einer Erklärung zum Verbinden (2) oder den verschiedenen Fenstern (9).

#### Möglicher Einstieg am Beispiel einer Mini-App:

Ein Einstieg in das Werkzeug sollte mithilfe eines einfachen Beispiels erfolgen. Denkbar ist z.B. eine Mini-App mit einem Button und einem Begrüßungstext oder ein Bild, welches nach Anklicken des Buttons erscheint. Anhand eines solchen Beispiels können wesentliche Elemente der Benutzeroberfläche wie Palette, Betrachter, Komponenten-Fenster, Eigenschaften-Fenster, Designer-Modus, Blöcke-Modus und das Verbinden mit einem Smartphone oder Tablet veranschaulicht werden. Anschließend können die Lernenden durch Implementieren einer ähnlichen Mini-App das Werkzeug eigenständig erkunden.

### 02\_Paint-App

Mithilfe dieser Arbeitsblätter wird schrittweise eine Mal-App implementiert. Dabei gestalten alle ein gleiches Grundprogramm, welches das Zeichnen von Punkten und Linien sowie die Auswahl einer von drei Farben ermöglicht. Das Grundprogramm kann dann um zusätzliche Funktionalitäten erweitert werden, deren Umsetzung sich unterschiedlich schwierig gestaltet und so eine innere Differenzierung erlaubt.

Ziele sind:

- weitere Erkundung des Werkzeuges und Implementierung von einfachen Funktionalitäten
- Kennenlernen der Zeichenflächen-Komponente und weiterer Komponenten (z.B. Schieberegler, Kamera, Benachrichtigung, ...) sowie zugehöriger Blöcke
- Zunehmend komplexere Gestaltung einer Oberfläche unter Zuhilfenahme von Gestaltungselementen der Kategorie Anordnung
- Analyse eines gegebenen Programms
- Zunehmend komplexere Implementierung von Funktionalitäten

### 03\_Sensoren\_im Smartphone

Im Zusammenhang mit der Programmierung von Apps bietet es sich an, auch über die in Smartphones enthaltenen Hardware-Komponenten zu sprechen. Erfahrungsgemäß haben Lernende hierzu erstaunlich wenig allgemeines Vorwissen bzw. verfügen nur über ausgewähltes Spezialwissen: bekannt sind zwar zum Teil sehr vertiefte Details zu Kamera, Display, Akku, Prozessor und Speicher, in Smartphones enthaltene Sensoren sind dagegen kaum bekannt.

Diese sind aber bereits für einfachste Anwendungen wie etwa das automatisierte Anpassen des Bildschirms beim Drehen des Smartphones oder das Abdunkeln des Displays beim Telefonieren

zwingend notwendig. Solche einfachen Beispiele könnten mithilfe eines Calliopes oder anderer geeigneter Werkzeuge vor oder während einer Unterrichtssequenz zum Algorithmischen Problemlösen mithilfe des App-Inventors rekonstruiert werden.

Die hier angebotenen Materialien bieten eine Thematisierung direkt innerhalb einer Unterrichtssequenz zum Thema „Algorithmisches Problemlösen mit dem App Inventor“ an. Dazu werden beispielhaft zwei Sensoren („Ortssensor“ und Beschleunigungssensoren) in mobilen Geräten untersucht und ihre Verwendung im App Inventor thematisiert.

### Didaktische Anmerkungen zur Komplexität des Werkzeugs

Der App Inventor verfügt über eine große Vielzahl an Komponenten, die in eigenen Apps verwendet werden können. Mithilfe der Beispiele 01 – 03 werden exemplarisch einige vorgestellt. Auf eine Thematisierung weiterer Komponenten wird an dieser Stelle verzichtet. Die Erfahrung zeigt, dass Schülerinnen und Schüler bei der Bearbeitung der Aufgaben oder der Entwicklung eigener Projekte eigenständig die Möglichkeiten des App Inventors weiter erkunden. Da die Verwendung einiger Komponenten im App Inventor nicht immer selbsterklärend ist, können diese dann bei Bedarf im Unterrichtsverlauf vorgestellt und erläutert werden. Dabei wird empfohlen, das Testen und die Erkundung der gewünschten Komponenten „vorzuleben“: sowohl die Lernenden als auch die Lehrkraft müssen nicht alle Komponenten des App Inventors und ihre Verwendung genau kennen. Spezielles Werkzeugwissen kann bei Bedarf immer noch durch gemeinsames Ausprobieren, ggf. durch Internet-Recherche oder gemeinsame Analyse der vom App Inventor zur Verfügung gestellten Hilfen erworben werden.

### 04\_FeelFine-App

Mithilfe dieser Arbeitsblätter wird sukzessiv eine „Gesundheits-Ratgeber-App“ zur Erfassung der Gefühlslage aufgebaut. Hierbei wird eine bereits vorhandene Vorlage schrittweise um Funktionalitäten erweitert. Dabei können die Lernenden diese je nach Vorwissen (Operationen auf Listen, Zeichenketten) aus verschiedenen Möglichkeiten auswählen. Außerdem werden basierend auf den umgesetzten Funktionalitäten Chancen und Risiken solcher Ratgeber-Apps diskutiert.

Ziele sind:

- Analyse der Komponenten einer Vorlage
- Implementierung zunehmend komplexerer Funktionalitäten.
- Zunehmend komplexere Gestaltung bzw. Erweiterung der Oberfläche einer App.
- Diskussion von Chancen und Risiken solcher Ratgeber-Apps unter Berücksichtigung informatischen Wissens.

Je nach Lerngruppe und sind weitere Ziele:

- Verwendung von Variablen und Listen bei der Implementierung.
- Auswahl und Verwendung geeigneter Sensoren zur Erweiterung der Funktionalitäten der App.

## Hinweise

Die Inhalte der Leitfäden und didaktischen Hinweise bringen ausschließlich Ansichten und Meinungen der Autorin zum Ausdruck.

Für die korrekte Ausführbarkeit der Quelltexte wird keine Garantie übernommen. Auch für Folgeschäden, die sich aus der Anwendung der Quelltexte oder durch eventuelle fehlerhafte Angaben ergeben, wird keine Haftung oder juristische Verantwortung übernommen.

Die Idee für eine App zum Erfassen der Gefühlslage (03\_FeelFine-App) stammt aus den Materialien von <https://appinventor.mit.edu/explore/app-building-guides> (Zugriff am 15.01.2020) zum Thema „Mood Ring“,

[http://ai2.appinventor.mit.edu/?locale=en&repo=http://appinventor.mit.edu/yrtoolkit/yr/aiaFiles/moodCounter/yr\\_mood\\_starter.asc](http://ai2.appinventor.mit.edu/?locale=en&repo=http://appinventor.mit.edu/yrtoolkit/yr/aiaFiles/moodCounter/yr_mood_starter.asc) (Zugriff am 15.01.2020), beides lizenziert unter einer [Creative Commons Namensnennung - Weitergabe unter gleichen Bedingungen 4.0 International Lizenz](#).

Übernommen wurde die Idee einer Erfassung der aktuellen Gefühlslage.

Alle weiteren Materialien wie beispielsweise die zur Verfügung gestellten Projekte, die zugehörigen Arbeitsaufträge zur Implementierung dieser und weiterer Funktionalitäten und Diskussion von Chancen und Risiken wurden selbst entwickelt.

Dieses Werk ist lizenziert unter einer [Creative Commons Namensnennung - Nicht kommerziell - Keine Bearbeitungen 4.0 International Lizenz](#). Sie erlaubt Download und Weiterverteilung des vollständigen Werkes unter Nennung meines Namens, jedoch keinerlei Bearbeitung oder kommerzielle Nutzung.