

# V4T PROJEKT

## E-LEARNING KURS

### MODUL 3: PROGRAMMIERANWENDUNGEN FÜR MOBILE GERÄTE

*Dieses Modul beginnt mit einem Überblick über die Technologien und Werkzeuge, die für die Programmierung mobiler Anwendungen zur Verfügung stehen, und bietet anschauliche Beispiele, die mit GrundschülerInnen verwendet werden oder werden könnten. Es folgt eine detaillierte Darstellung der Programmierumgebung „App Inventor“. Der Entwicklungsprozess von zwei beliebten Spielen in „App Inventor“ wird beschrieben. Schließlich stellt dieses Modul die „Metaverse Augmented Reality Web“-Plattform vor, die es LehrerInnen und SchülerInnen ermöglicht, ihre eigenen interaktiven „Augmented Reality“-Anwendungen einfach zu erstellen.*

#### Kapitel 1 - Einführung in die APP-Programmierung für mobile Geräte

##### 1.1 – Verfügbare Technologien

Bevor Sie mit der Programmierung Ihrer allerersten App beginnen können, müssen Sie sich entscheiden, auf welchem Softwaresystem Sie Ihre App bereitstellen möchten und welche Programmiersprache Sie verwenden möchten. Die beliebtesten Softwaresysteme sind Android von Google oder iOS von Apple.

Die Programmierung für Android erfordert Kenntnisse der Programmiersprache Java und des Android SDK.

iOS läuft nur auf Apple-Geräten und die Programmierlandschaft XCode ebenfalls, daher benötigen Sie einen Apple-Computer mit Mac OSX. Um eine App für iOS-Geräte zu programmieren, müssen Sie die Sprachen Objective-C oder Swift beherrschen.

Wenn Sie mit der Programmierung beginnen wollen, haben Sie verschiedene Möglichkeiten zu starten. In der folgenden Einheit konzentrieren wir uns auf die verfügbaren Technologien.

##### **Native Apps**

Native Apps sind in den Sprachen der bereits erwähnten Softwaresysteme (Objective-C, Cocoa) geschrieben.

##### **Web Apps (HTML5-Apps)**



Eine Technologie, mit der Sie beginnen können, wenn Sie nicht sicher sind, was Sie programmieren oder wo Sie anfangen sollen. Sie beginnen mit dem Erlernen von Javascript und beginnen mit der Programmierung von Webanwendungen, die mit Hilfe von Emulatoren problemlos in mobile Versionen für mobile Geräte umgewandelt werden können. Ein Emulator ist ein System, das ein Programm aus einer Architektur in eine andere Struktur oder ein anderes System kloniert. Mit einem Emulator können Sie eine Web App in eine für andere Systeme (z.B. Android) verwendbare App umwandeln.

Online-Ressourcen

Erfahren Sie einige Grundlagen über Native Apps -

<https://searchsoftwarequality.techtarget.com/definition/native-application-native-app>

## 1.2 - Entwicklungsumgebung und Programmiersprachen

Wenn Sie mit der Programmierung einer App für Android-Systeme beginnen möchten, können Sie mit Android Studio beginnen, das eine "Integrated Development Environment" kurz IDE bietet.

Android Studio bietet:

- Gradle-basiertes Build-System
- funktionsreicher Emulator
- Entwicklungsumgebung für verschiedene Android-Geräte
- Sofortstart, um Änderungen an Ihre laufende Anwendung zu übertragen, ohne ein neues APK zu erstellen.
- Codevorlagen und GitHub-Integration zur Unterstützung bei der Erstellung gemeinsamer App-Funktionen und dem Import von Beispielcode.
- Test-Tools und Frameworks
- Lint-Tools, um Leistung, Benutzerfreundlichkeit, Versionskompatibilität und andere Probleme zu erfassen.
- C++- und NDK-Unterstützung

Das Benutzerhandbuch ist in einer einfachen Sprache verfasst und motiviert AnfängerInnen, mit der Programmierung ihrer ersten Apps zu beginnen.

Der Einstieg in die Erstellung Ihrer allerersten App ist auch mit Hilfe der Blocktechnologie möglich. Das ist viel einfacher als das Erlernen einer Codesprache, schränkt aber auch Ihre Möglichkeiten ein.

"GoodBarber" ist ein Anbieter eines solchen App Toolkits. Sie erhalten eine 30-tägige Testversion, wenn Sie sich registrieren. Das System ermöglicht es Ihnen, ganz einfach eine App mit netten Funktionen für

zukünftige BenutzerInnen zu entwickeln (Benutzerauthentifizierung, Push-Benachrichtigungen, Monetarisierung und mehr). "Goodbarber" bietet verschiedene Preismodelle ab € 32,- pro Monat an. In Kapitel 2 konzentrieren wir uns auf "App Inventor", eine Entwicklungsumgebung für Android-Systeme von Google.

#### Online Ressourcen

**Android Studio** - <https://developer.android.com/studio>

Laden Sie Android Studio herunter und lesen Sie die Bedienungsanleitung, um mit der Programmierung zu beginnen.

**GoodBarber** - <https://www.goodbarber.com/>

Toolkit-Anbieter, um eine App mit Hilfe von Blöcken zu erstellen.

### 1.3 - Apps und neue Technologien für die Grundschule

Wie können unsere Kinder auf das digitale Zeitalter vorbereitet werden, das uns alle bereits umgibt? Wir müssen komplexe Inhalte in einfache Nutzungsmöglichkeiten überführen. Das Verständnis von Code und Algorithmus ist eine Kernkompetenz in einer digitalen Welt.

Es gibt viele Möglichkeiten, Kindern digitale Kompetenzen zu vermitteln oder besser, sie Erfahrungen mit digitalen Umgebungen machen zu lassen.

#### Lernen mit Apps, Spielzeugen und Plattformen

Es gibt bereits viele Apps, Online-Plattformen und Spielzeuge, mit denen die ersten Schritte im Coding gemacht werden können (siehe Online-Ressourcen). Diese Apps, Plattformen und Spielzeuge können in der Schule oder zu Hause für die ersten digitalen Erfahrungen genutzt werden.

##### App - Swift Playgrounds

In dieser iPad-App von Apple lösen EinsteigerInnen in die Programmierung Rätsel und steuern Charaktere mit echtem Swift-Code. Im Buch "Intro to App Development with Swift" erfahren sie, wie man eine erste App von Anfang bis Ende erstellt - durch das Erlernen der Grundlagen von Xcode. Darüber hinaus bietet Apple ein 'Everyone Can Code Curriculum', das LehrerInnen dabei unterstützt, Codierung von der Grundschule bis zur Universität zu unterrichten.

#### Programmiersprachen für Kinder

Es gibt spezielle Programmiersprachen für Kinder, die meist auf visuellen Codes basieren.

##### Scratch

Scratch ist eine visuelle Programmiersprache, die am MIT Media Lab speziell für Kinder zwischen 8 und 16 Jahren entwickelt wurde, um interaktive Geschichten, Spiele und Animationen zu programmieren, die mit



anderen geteilt werden können. Darüber hinaus lernen die SchülerInnen von der Grundschule bis zur weiterführenden Schule fächerübergreifend mit Scratch Mathematik, Informatik, Sprachen und Sozialwissenschaften. Im [ScratchEd Online Community Archive](#) können PädagogInnen Ressourcen, Diskussionsbeiträge und Geschichten vieler anderer PädagogInnen finden.

### [NEPO - Open Roberta Lab](#)

Die Initiative "Open Roberta" (vom Fraunhofer-Institut für Intelligente Analyse- und Informationssysteme IAIS) führt in die Programmierung von Robotern ein. Kinder können die visuelle Programmiersprache NEPO erlernen und ihre Fähigkeiten mit Hilfe eines virtuellen Roboters testen. Die Roberta-Initiative bietet Schulungen, Materialien, Lehrpläne und Bauanleitungen für LehrerInnen an.

### **Workshops**

Workshops sind für Kinder eine großartige Möglichkeit, zusammenzuarbeiten und in und mit einer Gruppe anderer Kinder zu lernen.

### [Maker Spaces](#)

In Maker Spaces können Kinder mit Materialien und Werkzeugen etwas Neues erschaffen. Maker Spaces verwenden oft Werkzeuge wie Arduino, Raspberry Pi, Calliope Mini, BBC micro:bit und Makey Makey oder Werkzeuge zum Bau und zur Programmierung von Robotern. Smartphones, Apps, Tablets und 3D-Drucker können genutzt und ausprobiert werden. Es geht dabei nicht nur um den Aufbau von Programmierkenntnissen, sondern auch um einen Ort, an dem Kinder kreativ sein können.

### **Online Ressourcen**

**Video 'What most schools don't teach'** - <https://youtu.be/nKlu9yen5nc>: Anregendes Video über den Aufbau von Computer-/Programmierfähigkeiten

**Scoyo - Programmieren lernen für Kinder: Mit Spaß fit für die Zukunft** - <https://www-de.scoyo.com/eltern/kinder-und-medien/programmieren-lernen-kinder-fit-fuer-die-zukunft>

Online-Magazin für Eltern rund um Lernen, Schule, Familienleben & Medienkompetenz

**Code.org** - <https://code.org/>: Bietet einstündige Tutorials für jedes Alter in über 45 Sprachen.

**CodeAcademy** - <https://www.codecademy.com/>: Lernen von Programmiersprachen wie Python, PHP, jQuery, JavaScript,.....

**Lightbot** - <http://lightbot.com/index.html>: App für Kinder zum Erlernen der Programmierung

**Dash** - <https://uk.makewonder.com/dash/>: Dash ist ein Roboter, der über Apps für verschiedene Altersgruppen gesteuert werden kann.

**CoderDojo** - <https://zen.coderdojo.com/find>: Ein globales Netzwerk von kostenlosen, ehrenamtlich geführten, kommunalen Programmierclubs für junge Menschen von 7 bis 17 Jahren.



## **Making - Kreatives digitales Gestalten und Experimentieren mit Kindern. -**

[https://de.slideshare.net/sandra\\_slideshare/making-kreatives-digitales-gestalten-und-experimentieren-mit-kindern-einfhrung-und-ausgewhlte-werkzeuge](https://de.slideshare.net/sandra_slideshare/making-kreatives-digitales-gestalten-und-experimentieren-mit-kindern-einfhrung-und-ausgewhlte-werkzeuge)

Einführung und ausgewählte Werkzeuge

**Arduino** - <https://www.arduino.cc/>

**Raspberry Pi** - <https://www.raspberrypi.org/>

**Makey Makey** - <https://makeymakey.com/>

**Calliope Mini** - <https://calliope.cc/en>

**BBC micro:bit** - <https://microbit.org/>

## **Kapitel 2: Erstellen von Apps mit „App Inventor“**

### **2.1 – Erste Schritte mit “App Inventor”**

“App Inventor” ist eine Webanwendung, mit der Sie Apps für Android-Geräte (Smartphones oder Tablets) mit Hilfe einer visuellen Programmiersprache entwickeln können. Es handelt sich um eine kostenlose Open-Source-Software.

Um mit dem “App Inventor” arbeiten zu können, müssen Sie Folgendes besitzen:

- ein Google-Konto,
- einen Computer mit Internetzugang,
- einen kompatiblen Browser (Mozilla Firefox, Apple Safari, Google Chrome - Microsoft Internet Explorer wird nicht unterstützt).

Wenn Sie Probleme mit der Internetverbindung oder der Anmeldung mit einem Google-Konto haben, können Sie App Inventor 2 Ultimate verwenden.

App Inventor speichert die Projekte im Cloud-Speicher unter dem Namen Ihres Google-Kontos. Sie müssen keine Projekte auf Ihrer lokalen Festplatte speichern.

Eine App Inventor-Anwendung besteht aus dem Component Designer und dem Blockeditor. Im App Inventor Designer können Sie die Bildschirme der App erstellen, indem Sie die Elemente (Komponenten) der Anwendung anordnen. Im App Inventor Blocks Editor können Sie das Verhalten der Komponente programmieren, indem Sie Codeblöcke verschieben. Blöcke stellen Befehle dar, die eine auszuführende Aktion angeben.



Es gibt zwei Arten von Komponenten: sichtbare und unsichtbare. Sichtbare Komponenten (wie Buttons, Labels, Textboxen) sind diejenigen, die Sie beim Start der Anwendung sehen können und die Teil der Anwendungsoberfläche sind. Unsichtbare Komponenten (wie Accelerometer, Sound, OrientationSensor) werden nicht angezeigt und ermöglichen den Zugriff auf die eingebaute Funktionalität des Gerätes. Sie sind nicht Bestandteil der Benutzeroberfläche.

Sie können mehr als einen Bildschirm für eine App erstellen, aber Sie müssen vorsichtig sein, da jeder Bildschirm Computerressourcen verbraucht. Jeder Bildschirm hat seinen Namen und seine Eigenschaften. Als Faustregel gilt, dass Sie nicht mehr als 10 Bildschirme pro Projekt haben sollten.

Wenn Sie kein Android-Handy oder Tablet haben, um sich mit Ihrem Computer zu verbinden, können Sie mit App Inventor immer noch Apps entwickeln und testen. App Inventor bietet einen Android-Emulator, der ein virtuelles Android-Gerät auf Ihrem Computerbildschirm installiert und genau wie ein Android-Gerät funktioniert.

### Online Ressourcen

Ein Einsteiger-Tutorial für App Inventor: <http://appinventor.mit.edu/explore/ai2/beginner-videos.html>

Erste Schritte mit MIT App Inventor 2: <https://appinventor.mit.edu/explore/get-started.html>

App Inventor Video von Google: <https://www.youtube.com/watch?v=sGiaXOKqeKg>

Ein kurzer Überblick über den Designer und den Blockeditor:

<https://appinventor.mit.edu/explore/designer-blocks.html>

## 2.2 – Die Designumgebung

Der **Designer** hilft Ihnen, das Erscheinungsbild Ihrer App anzupassen. Er besteht aus fünf Spaltenbereichen:

### ● Palette

Aus der Palette können Sie die Komponenten sowohl On- als auch Off-Screen auswählen und in den Viewer-Bereich ziehen. Die Spalte Palette ist in Gruppen von Komponenten organisiert, die als Schubladen bezeichnet werden (User Interface, Layout, Media, Drawing and Animation, Maps, Sensoren, Social, Storage, Connectivity, Experimental, Extension usw.).

**User Interface** Komponenten sind: Schaltflächen, Kontrollkästchen, Datums-Picker, Bilder, Labels, Listen-Picker, Listenansichten, Benachrichtiger, PasswortTextBoxen, Schieberegler, Spinner, Zeitauswahl und Web Viewer.

**Layout** besteht aus Formatierungselementen. Durch die Verwendung dieser Komponenten können Sie das Aussehen jedes Bildschirms bestimmen (HorizontalArrangement, HorizontalScrollArrangement, TableArrangement, VerticalArrangement, VerticalScrollArrangement).

**Media** Schublade besteht aus Elementen wie:



- Camcorder (eine sichtbare Komponente zur Aufnahme eines Videos mit dem Camcorder des Geräts)
- Kamera (eine nicht sichtbare Komponente, um ein Bild mit der Kamera des Geräts aufzunehmen)
- Player (eine Komponente, die Audio wiedergibt und die Telefonvibration steuert)

**Drawing and Animation** Drawer ermöglicht es Ihnen, Zeichnungen und Animationen zu erstellen. Es besteht aus Elementen wie Ball, Canvas und ImageSprite.

- Canvas ist ein berührungsempfindliches rechteckiges Untermenü in Ihrer App, in dem Sie zeichnen und Sprites verschieben können.
- Sie können Animationen programmieren, indem Sie Ball und ImageSprite Komponenten innerhalb einer Leinwand platzieren. Diese Komponenten können auf Tippen und Ziehen reagieren, mit anderen Sprites und dem Rand der Leinwand interagieren und sich entsprechend ihren Eigenschaftswerten bewegen und transformieren.

**Sensorkomponenten** ermöglichen den Zugriff auf die Gerätesensoren (Beschleunigungssensoren, Positionssensoren, Orientierungssensoren, Gyroskope, Barcodescanner, Näherungssensoren usw.).

**Storage** ist eine nicht sichtbare Komponente zum Schreiben und Lesen von Dateien in das mit Ihrer App verbundene private Datenverzeichnis.

Die **Konnektivitätsschublade** besteht aus ActivityStarter, BluetoothClient, BluetoothServer und Webkomponenten.

- **Viewer**

Im **Viewer**-Bereich sehen Sie den Inhalt der einzelnen Bildschirme Ihrer Anwendung.

- **Components**

In der Spalte **Components** sehen Sie in einer baumartigen Struktur alle Komponenten (sichtbar und unsichtbar) eines Bildschirms. Sie können eine Komponente umbenennen oder löschen.

- **Media**

In der Spalte **Media** können Sie die Mediendateien (Audio-, Bild- und Videodateien) organisieren.

- **Properties**

In der Spalte **Properties** können Sie das Aussehen und andere Eigenschaften der Komponenten festlegen.

## Online Ressourcen

Eine Kurzanleitung für den Designer



<https://appinventor.mit.edu/explore/sites/all/files/Teach/media/MITAppInventorQuickReference.pdf>

Tutorial zur Verwendung der Komponenten der Benutzeroberfläche

<http://www.appinventor.org/content/howDoYou/UIModules>

Eine Liste der Grundkomponenten von App Inventor

<https://appinventor.mit.edu/explore/content/basic.html>

## 2.3 - Programmieren mit Blöcken

Der **Blockeditor** verwaltet Codeblöcke und programmiert das Verhalten Ihrer App. Blöcke sind Puzzleteile, die zur Erstellung eines Programms verwendet werden.

Von App Inventor erstellte Anwendungen sind ereignisgesteuert. Anweisungen werden nicht in einer vordefinierten Reihenfolge ausgeführt, sondern reagieren auf Ereignisse. Ein Ereignis ist eine Aktion wie das Anklicken einer Schaltfläche oder das Tippen auf dem Bildschirm. Das Schütteln des Telefons ist ebenfalls ein Ereignis. Der Programmierer beschreibt, wie das Telefon auf Ereignisse reagieren soll, indem es **Ereignisbehandlungsblöcke** verwendet.

### Built-in Blöcke

Built-in Blöcke (Control, Logic, Math, Text, Lists, Colors, Variables and Procedures blocks) sind unabhängig von den Komponenten des Projekts verfügbar.

**Control blocks** werden für Entscheidungen verwendet.

**Logic blocks.**

**Math blocks** sind die Blöcke, die mathematische Funktionen ausführen.

**Text** sind die Blöcke, die die Zeichenkettenbehandlung ausführen.

Mit den **Lists** blocks können Sie die Elemente einer Liste verwalten..

Mit den **Variable** blocks können Sie globale und lokale Variablen verwalten. Es gibt fünf Arten von Variablenblöcken (Initialisieren des globalen Namens auf, Erhalten, Setzen, Initialisieren des lokalen Namens auf In-Do und Initialisieren des lokalen Namens auf In-Return).

### Komponentenspezifische Blöcke

Viele der Komponenten Ihrer App können **Methodenaufrufblöcke** in ihrer komponentenspezifischen Schublade haben. Ein **Methodenaufrufblock** ist ein vordefinierter Satz von Anweisungen, mit denen Sie die Funktionalität der Komponente nutzen können.

### Verfahrensblöcke

Mit den Verfahrensblöcken können Sie Verfahren verwalten. Ein Verfahren ist eine Folge von Anweisungen, die unter einem Namen zusammengefasst sind und eine bestimmte Aufgabe ausführen.





Anstatt komplexe Programme zu haben, können Sie Verfahren erstellen und diese aufrufen, wann immer Sie sie verwenden möchten.

Die Verfahrens- und viele andere Blöcke, einschließlich "if-else" und "make list", sind mit einer blauen **Mutatorentaste** versehen. Wenn Sie darauf klicken, können Sie den Block transformieren, z.B. Argumente zu Verfahren hinzufügen oder 'else'-Verzweigungen zu einem if.

### Online Ressourcen

Auf dieser Seite wird erklärt, wie Basisblöcke funktionieren.

<https://appinventor.mit.edu/explore/understanding-blocks.html>

Ein Überblick über die Architektur einer App

<http://www.appinventor.org/Architecture2>

Dieses Tutorial zeigt Ihnen, wie Sie Verfahren in App Inventor erstellen können.

<http://appinventor.mit.edu/explore/ai2/support/concepts/procedures.html>

Dieses Tutorial zeigt Ihnen, wie Sie Verfahren in App Inventor aufrufen können.

<http://ai2.appinventor.mit.edu/reference/blocks/procedures.html#do>

Ein kurzer Überblick über die Verwendung von Mutatoren in App Inventor

<https://appinventor.mit.edu/explore/ai2/support/concepts/mutators.html>

## 2.4 - Verbindung zu einem Telefon oder Tablet herstellen

### Einrichten des App Inventors

Sie können die Apps auf zwei verschiedene Arten auf Ihrem Android-Gerät ausführen: über eine Wi-Fi-Verbindung (AI Companion) oder über ein USB-Kabel. Im ersten Fall müssen sowohl Ihr Computer als auch das Android-Gerät mit demselben Wi-Fi-Netzwerk verbunden sein. Wenn Sie kein Android-Handy oder Tablet haben, können Sie mit App Inventor immer noch Apps erstellen. App Inventor bietet einen Android-Emulator, der genau wie ein Android-Gerät funktioniert, aber alles läuft auf Ihrem Computerbildschirm ab. Sie können Ihre Apps auf dem Emulator testen und die App trotzdem an andere weitergeben, auch über den Play Store. In diesem Fall müssen Sie die App Inventor Companion App auf Ihrem Gerät installiert haben.



### **Installation und Betrieb des Emulators in AI2**

Die Schritte zur Installation und Ausführung von App Inventor sind wie folgt:

Schritt 1. Installieren der App Inventor Setup Software

Schritt 2. aiStarter starten (nur Windows & GNU/Linux)

Schritt 3. Öffnen Sie ein App Inventor-Projekt und verbinden Sie es mit dem Emulator.

### **Verbinden mit einem Telefon oder Tablet über ein USB-Kabel**

Verbinden mit einem Telefon oder Tablet über ein USB-Kabel

Die Schritte für den Beginn der Verwendung von App Inventor mit dem USB-Kabel sind wie folgt:

Schritt 1: Installieren der App Inventor Setup Software

Schritt 2: Laden Sie die MIT AI2 Companion App herunter und installieren Sie sie auf Ihrem Handy.

Schritt 3. aiStarter starten (nur Windows & GNU/Linux)

Schritt 4: Richten Sie Ihr Gerät für USB ein (Schalten Sie USB Debugging EIN).

Schritt 5: Verbinden Sie Ihren Computer und Ihr Gerät und authentifizieren Sie sich, falls erforderlich.

Schritt 6: Testen der Verbindung

### **Verbinden Sie Ihr Telefon oder Tablet über WiFi.**

Die folgenden Schritte führen Sie durch den Prozess:

Schritt 1: Laden Sie die MIT AI2 Companion App herunter und installieren Sie sie auf Ihrem Handy.

Schritt 2: Verbinden Sie sowohl Ihren Computer als auch Ihr Gerät mit dem gleichen WiFi-Netzwerk.

Schritt 3: Öffnen Sie ein App Inventor-Projekt und verbinden Sie es mit Ihrem Gerät.

### **Online Ressourcen**

Ein Überblick über die Einrichtung von App Inventor

<http://appinventor.mit.edu/explore/ai2/setup.html>

Ein Überblick über die Installation und den Betrieb des Emulators

<http://appinventor.mit.edu/explore/ai2/setup-emulator.html>

Eine Übersicht über den Anschluss an ein Telefon oder Tablet mit einem USB-Kabel.

<http://appinventor.mit.edu/explore/ai2/setup-device-usb.html>

Eine Übersicht über die Verbindung zu einem Telefon oder Tablet über WiFi

<http://appinventor.mit.edu/explore/ai2/setup-device-wifi.html>



## Kapitel 3: Erstellen von Spielen mit App Inventor

### 3.1 - Wie man ein einfaches Spiel erstellt

Nachfolgend sind exemplarische Schritte aufgeführt, die befolgt werden können, um ein einfaches Spiel mit dem App Inventor zu erstellen.

#### Gestalten Sie die grafische Benutzeroberfläche des Spiels.

Im App Inventor Designer:

- Wählen Sie Komponenten (Button, Label usw.) aus der **Palette** aus.
- Ziehen Sie die ausgewählten Komponenten in den **Viewer**-Bereich und ordnen Sie sie so an, dass sie das Aussehen Ihres Spiels bestimmen.
- Aus der **Layoutpalette** können Sie mit Hilfe von Bildschirmanordnungen **HorizontalArrangement** (von links nach rechts), **VerticalArrangement** (von oben nach unten links ausgerichtet) oder **TableArrangement** (tabellarisch) der Komponenten erstellen.
- Die ausgewählten Komponenten werden im **Komponentenbereich** platziert und vom App Inventor benannt. Sie können den Namen dieser Komponenten ändern oder bei Bedarf löschen.
- Füge Mediendateien (Bilder, Sounds) zu deinem Spiel hinzu, indem du sie von deinem Computer hochlädst. Mediendateien werden im **Medienbereich** unter der Spalte **Komponente** aufgelistet.
- Durch Anklicken einer Komponente aus dem Viewer-Bereich können Sie deren Details (Titel, Hintergrundfarbe usw.) ändern, die im Properties-Bereich aufgelistet sind.
- Wählen und konfigurieren Sie ggf. aus der Zeichen- und Animationspalette die Leinwand. Auf dem zweidimensionalen, berührungsempfindlichen rechteckigen Panel der Leinwand können Sie Bälle oder ImageSprites zeichnen und bewegen.
- Fügen Sie bei Bedarf eine Uhrenkomponente (Clock) hinzu. Clock funktioniert wie ein Timer, der in regelmäßig eingestellten Intervallen feuert. Es ist eine nicht sichtbare Komponente, die Manipulationen und Berechnungen zum Thema Zeit durch die Verwendung der internen Uhr am Telefon oder am Tablett ermöglicht.

#### Definieren Sie das Verhalten der Komponenten - Der Blockeditor

- Erstellen und initialisieren Sie globale und lokale Variablen.
- Wählen Sie eine eingebaute Schublade (Control, Logic, Math, Text usw.) oder eine komponentenspezifische Schublade (eine Schublade für jede Komponente, die Sie im Designer

erstellt haben), um eine Gruppe von Blöcken für diese Komponente zu erhalten. Jede spezifische Komponente hat Blöcke für Ereignisbehandler, Methodenaufrufe, das Setzen von Eigenschaften und das Abrufen von Eigenschaften.

- Wählen Sie die richtigen Blöcke aus und ziehen Sie sie in den Anzeigebereich.
- Konfigurieren Sie diese Blöcke und ziehen Sie sie entsprechend, um das Spielverhalten einzustellen.
- Definieren Sie Verfahren, um neue Befehlsblöcke zu erstellen, die wiederholt verwendet werden können.
- Verwenden Sie Verfahren und Methodenaufrufe.

### 3.2 – Erstellen eines Pong Spiels mit App Inventor

[Pong](#) ist ein Spiel, mit dem die Schritte zur Entwicklung von Spielen, wie in Abschnitt 3.1 beschrieben, demonstriert werden. Die folgenden beiden Dokumente beschreiben, wie man ein **Pong**-Spiel entwickeln kann.

[Design der grafischen Benutzeroberfläche des Pong-Spiels](#)

[Definition der Funktionalität der Komponenten des Pong-Spiels - Der Blockeditor](#)

#### Zusätzliche Ressourcen

[https://appinventor.mit.edu/explore/sites/.../Pong%205a\\_13.pdf](https://appinventor.mit.edu/explore/sites/.../Pong%205a_13.pdf)

<http://www.appinventor.org/content/ai2apps/intermediateApps/pong>

### 3.3 – Erstellen eines MoleMash-Spiels mit App Inventor

[MoleMash](#) ist ein weiteres Spiel, das die in Abschnitt 3.1 beschriebenen Schritte zur Entwicklung von Spielen demonstriert. Die folgenden beiden Dokumente beschreiben, wie das **MoleMash**-Spiel entwickelt werden kann.

[Design der grafischen Benutzeroberfläche des MoleMash-Spiels](#)

[Definition der Funktionalität der Komponenten des MoleMash-Spiels - Der Blockeditor](#)

#### Zusätzliche Ressourcen

Mole Mash - App Inventor Tutorial Part 1 <https://youtu.be/Ya1ejdRwKvw>

Mole Mash - App Inventor Tutorial Part 1 <https://youtu.be/IPSG7bYXN2M>



## Kapitel 4 - Metaverse Augmented Reality

### 4.1 – What is Metaverse

**Metaverse** ist eine kostenlose und einfach zu erlernende Webplattform, die es LehrerInnen und SchülerInnen ermöglicht, ihre eigenen interaktiven Augmented Reality (AR)-Erlebnisse ohne Programmieraufwand zu erstellen. Die AR-Erlebnisse werden im Web im [Metaverse Studio](#) entwickelt und dann in der Metaverse Mobile App ([iOS](#) and [Android](#)) betrachtet.

Was können LehrerInnen und SchülerInnen mit Metaverse gestalten? Die Möglichkeiten sind endlos, aber hier sind einige exemplarische Beispiele:

- AR Schatzsuche,
- AR-Spiele,
- AR interaktive Geschichten,
- AR Interaktive Rätsel und Quizfragen,
- AR Virtuelle Exkursionen
- Medienwände

Was ist der Nutzen für die SchülerInnen? Die Schaffung interaktiver AR-Erlebnisse kann die digitale Kompetenz und die Fähigkeiten der SchülerInnen, die Kreativität und das logische Denken verbessern. Darüber hinaus engagieren sich die SchülerInnen stärker im Lernprozess, wenden ihr Wissen an und beginnen, je nachdem, auf welchem Fachgebiet ihr AR-Projekt basiert, als ForscherInnen zu denken und zu handeln. LehrerInnen können Metaverse in jedem Fach mit SchülerInnen aller Klassen verwenden und so projektbezogenes Lernen, Learning by Doing sowie kollaboratives Lernen fördern.

#### Online Ressourcen

[What Educators Have To Say About Metaverse:](#) In diesem Video teilen PädagogInnen, die Metaverse mit ihren SchülerInnen nutzen, ihre eigenen Erfahrungen und die Rückmeldungen und das Feedback ihrer SchülerInnen.

[Metaverse Preview:](#) Eine kurze Vorschau auf die Metaverse Studio und Metaverse Mobile App.

[Live AR Portals:](#) Dieses Video zeigt AR-Portale, die die NutzerInnen an verschiedene Orte auf der ganzen Welt "teleportieren". Die Portale wurden mit einer 360°-Kamera und Metaverse erstellt.

[Joe Merrill's Lake Park Elementary Class:](#) AR-Vokabularspiel für GrundschülerInnen, erstellt mit Metaverse.



## 4.2 - Beginnen Sie mit der Verwendung des Metaverse Studios

Um Metaverse Studio nutzen zu können, müssen Sie zunächst Ihr Konto auf der Plattform erstellen und sich dann mit der App-Umgebung vertraut machen. Das folgende Dokument enthält die Schritte, die Sie ausführen müssen, um Metaverse zu verwenden, sowie eine kurze Präsentation der Metaverse Studio-Umgebung.

Verwenden Sie dieses Dokument, [um sich mit Metaverse Studio vertraut zu machen](#).

### Online Ressourcen

[Metaverse Studio Overview](#): Demonstration der grundlegenden Möglichkeiten, das Metaverse Studio zur Erstellung von Augmented Reality Erfahrungen zu nutzen.

[Metaverse Dashboard Overview](#): Ein kurzer Überblick über die Metaverse Dashboard-Seite.

## 4.3 - Erstellen eines AR-Erlebnisses

Metaverse Studio bietet viele Werkzeuge, um ein einfaches oder auch komplizierteres AR-Erlebnis zu erstellen. Das folgende Dokument stellt die grundlegenden Schritte dar, die Sie ausführen müssen, um mit der Erstellung von AR-Erlebnissen zu beginnen.

Verwenden Sie dieses Dokument, [um ein AR-Erlebnis in Metaverse zu erstellen](#).

### Online Ressourcen

[Metaverse Storyboard Overview](#): Ein kurzer Überblick über das Metaverse Experience Storyboard.

[Character Scenes](#): Veranschaulichung der Einsatzmöglichkeiten von Charakterszenen.

[Text Input Scenes](#): Veranschaulichung, wie Texteingabeszenen verwendet werden können.

[Publishing Experiences](#): Ein kurzer Überblick darüber, wie Sie Ihre Metaverse-Erfahrungen veröffentlichen können.

[Add a GPS Location: T](#) Tutorial zum Hinzufügen einer GPS-Position zu einem Metaverse-Erlebnis.

[Build a Digital Breakout](#): Erfahren Sie, wie Sie in weniger als 15 Minuten eine digitale Breakout-Aktivität für Ihre Klasse erstellen können!

Schauen Sie sich diese Erlebniswelt an: <https://studio.gometa.io/discover/me/...>



[Build a Simple Trivia Game](#): Erfahren Sie, wie Sie ein einfaches 3-Fragen-Quizspiel erstellen, das Szenen-Timer für Fragen verwendet und Punkte mit Hilfe von Property Blocks auswertet.

