

Erste App programmieren (online)

Einstieg in die App-Programmierung mit
dem MIT App Inventor 2



Verfasser:

Christina Schramm

Kurz-Info:

Informatischer Inhalt: Einstieg in die grafische Programmierung mit Android

Jahrgangsstufe: 6 bis EF

Vorwissen: Keins

KURZINFORMATION FÜR DIE LEHRKRAFT

Titel: Erste App programmieren (online) – Einstieg in die App-Programmierung mit dem MIT App Inventor 2

Schulstufe: Unter-, Mittel- und Oberstufe

optimale Jahrgangsstufe: 7 bis EF

geeignete Kursart: Grundkurs



Themenbereich: Grafische Programmierung, Layoutgestaltung, Variablen, Kontrollstrukturen

EINORDNUNG IN GESETZLICHE RAHMENBEDINGUNGEN

Lehrplan NRW: Kompetenzbereiche: Modellieren, Implementieren, Kommunizieren und Kooperieren; Inhaltsbereiche: Algorithmen, Informatiksysteme

Bildungsstandards der GI: Prozessbereiche: Modellieren und Implementieren, Begründen und Bewerten, Strukturieren und Vernetzen, Kommunizieren und Kooperieren; Inhaltsbereiche: Algorithmen, Informatiksysteme, Informatik, Mensch und Gesellschaft

Einbindung in den Unterricht: Das Modul eignet sich zum Einsatz im Fernunterricht. Inhaltlich kann es einen ersten, motivierenden Einblick in die (App-)Programmierung bieten. Hierzu wird die grafische Programmierumgebung des MIT App Inventor 2 genutzt, sodass für die Programmierung vorgefertigte Blöcke zur Verfügung stehen, die wie ein Puzzle zusammengesetzt werden können.

Technische Voraussetzung: Alle Teilnehmenden müssen über einen PC mit Internetzugang verfügen. Ein Android-basiertes mobiles Endgerät (Tablet, Smartphone) ist hilfreich, jedoch nicht zwingend notwendig, da die Programme auch über einen Emulator ausgeführt werden können. Die zur Verfügung gestellten Arbeitsmaterialien müssen im Vorfeld heruntergeladen werden. Auch die Installation des AI Starters (siehe Blatt 0) sollte bereits vor dem Kurs erfolgen. Bei Problemen kann per Mail Kontakt zum InfoSphere-Team aufgenommen werden: infosphere-support@informatik.rwth-aachen.de

Vorkenntnisse: keine

Dauer: ca. 4,5 Stunden

Inhaltsbeschreibung: Zu Beginn erfolgt eine Einführung in die Grundelemente und die Handhabung der Videokonferenzsoftware Zoom, über die die Lernenden miteinander und mit den Betreuenden kommunizieren. Außerdem wird die Entwicklungsumgebung des MIT App Inventor 2 eingeführt. Nachdem die Schülerinnen und Schüler in Teams eingeteilt wurden, arbeiten sie in Breakout Sessions, angeleitet durch detaillierte Arbeitsmaterialien, an ihren ersten Programmierprojekten und entwickeln auf diese Weise eine erste Zeichenapp und kleine Spiele. Das Vorgehen ist dabei immer identisch strukturiert: Zunächst widmen sich die Schülerinnen und Schüler der Gestaltung des Layouts. Anschließend wird die Programmierung fokussiert. Während des Programmierens können die Teilnehmenden ihre Fortschritte stets unmittelbar nachvollziehen – entweder direkt auf einem Android-basierten mobilen Endgerät oder mit Hilfe eines Emulators.

Das Modul ermöglicht binnendifferenziertes Arbeiten.

INHALT

Kurzinformation für die Lehrkraft.....	2
Lernziele	5
Fachliche Analyse.....	5
Einordnung in gesetzliche Rahmenbedingungen	6
Didaktische/Methodische Schwerpunktsetzung.....	7
Beschreibung des Modulablaufs mit methodischer/didaktischer Begründung.....	7
Didaktische Prinzipien nach Baumann und Hubwieser.....	8
Einbettung in den Unterricht.....	9
Verlaufsplan des Moduls.....	10
Quellenverzeichnis	12
Abbildungsverzeichnis.....	12
Modulcheckliste.....	12
Troubleshooting/Problembehebung.....	12
Probleme bei der Verwendung des Emulators	12
Vorgehen beim Auftreten eines unbekannten Bugs.....	13
Modulmaterialien.....	13
Arbeitsblätter.....	13
Download des AI Starters	13
MIT App Inventor 2.....	13

LERNZIELE

Die Schülerinnen und Schüler...

- lösen angeleitet durch Arbeitsmaterialien in projektorientierter Zusammenarbeit informatische Problemstellungen (Programmierung kleiner, aber funktionierender Apps für Android-basierte Endgeräte).
- erproben die Handhabung eines blockbasierten Editors (MIT App Inventor 2) und wenden diesen zur zielgerichteten Programmierung kleiner mobiler Apps an.
- gestalten die Benutzeroberfläche zweckmäßig, indem sie sich mit der Platzierung von Steuerelementen auseinandersetzen.
- können den Unterschied zwischen Design und Funktion im Bereich der App-Programmierung erklären.
- zerlegen – angeleitet durch Arbeitsmaterialien – das Ausgangsproblem (Programmierung einer Zeichenapp, Programmierung kleiner Spieleapps) in mehrere Teilergebnisse, die schließlich in ihrem Zusammenwirken funktionierende Apps ergeben.
- entwerfen einfache Algorithmen unter Verwendung von Variablen und Kontrollstrukturen.
- überprüfen die Wirkung ihrer Programmierung mittels eines Android-basierten Endgerätes oder des Emulators, erkennen auf diese Weise unmittelbar Fehler und korrigieren diese.
- nutzen die Videokonferenzsoftware Zoom als geeignetes Werkzeug zur Zusammenarbeit trotz physischer Distanz.
- verwenden einfache Fachsprache (Design, Button, Screen Arrangement, Variable, If-Anweisung etc.), um innerhalb ihres Teams verschiedene Möglichkeiten und Lösungsansätze zur Umsetzung der Apps zu diskutieren.

FACHLICHE ANALYSE

Essentiell für dieses Online-Modul sind zwei Werkzeuge: Zoom und der MIT App Inventor 2.

Zoom als Software für Videokonferenzen ermöglicht eine Betreuung der Kursteilnehmenden ohne gegenseitigen physischen Kontakt. Außerdem bietet die Software das Feature der sogenannten Breakout Sessions. Hierbei handelt es sich um virtuelle Nebenräume, die die Teilnehmenden betreten können, um in kleineren Teams an ihren individuellen Projekten zusammenzuarbeiten. Falls während dieser Teamarbeit Fragen aufkommen oder die Teilnehmenden Hilfestellungen benötigen, können sie mittels eines Buttons einen Betreuenden zur Unterstützung in ihre Breakout Session rufen. Über das Screensharing können die Schülerinnen und Schüler ihre Fortschritte miteinander teilen.

Für die App-Programmierung nutzen die Teilnehmenden den MIT App Inventor 2, eine integrierte Entwicklungsumgebung, die ursprünglich von Google zur Programmierung von mobilen Android-basierten Endgeräten entwickelt wurde. Es handelt sich um einen Block-Editor, der eine Open-Blocks-Java-Bibliothek zur Erstellung der grafisch-basierten Programmiersprache verwendet, die durch das renommierte Massachusetts Institute of Technology (MIT) bereitgestellt wird. Die grafische Oberfläche des MIT App Inventors 2 ist geeignet, um Schülerinnen und Schülern, die bisher keine Kenntnisse im Bereich höherer Programmiersprachen haben, einen Einstieg in die App-Programmierung zu ermöglichen. Um den Computer, an dem mit dem MIT App Inventor 2 programmiert wird, mit dem mobilen Endgerät zu verbinden oder den Emulator zu verwenden, muss der sogenannte MIT AI2 Companion installiert werden – ein Vorgang, der nicht immer ganz selbsterklärend und fehlerfrei verläuft. Deshalb stellt das InfoSphere den Teilnehmenden bereits im Vorfeld des Kurses eine Sammlung aller Arbeitsmaterialien zum Download zur Verfügung. Diese Sammlung enthält unter anderem auch ein Informationsblatt zur Installation des MIT AI2 Companion. Sollten

dennoch Probleme bei der Installation auftreten, leistet das InfoSphere-Team einen individuellen Support (infosphere-support@informatik.rwth-aachen.de).

Im Erarbeitungsprozess kommen die Teilnehmenden mit grundlegenden Konzepten der Informatik in Berührung, z. B. mit Kontrollstrukturen (if-then-else-Abfragen, Schleifen).

Weiterführende Links zur Einarbeitung in die Thematik:

- Lang, M. (2018). *Einstieg in die Android IDE „App Inventor“*. Abgerufen von: <https://www.dev-insider.de/einstieg-in-die-android-ide-app-inventor-a-678505/>.
- Lehrerinnenfortbildungsserver Baden-Württemberg (2016). *Programmieren mit dem MIT App Inventor 2*. Abgerufen von: https://lehrerfortbildung-bw.de/u_matnatech/imp/gym/bp2016/fb1/2_i2_alg/2_kopiervorlagen/08_ai2/1_einfuehrung/.
- MIT App Inventor 2. Abgerufen von: <http://ai2.appinventor.mit.edu/>.
- MIT App Inventor for Educators. Abgerufen von: <https://community.appinventor.mit.edu/top/weekly>.
- Teaching with the App Inventor. Abgerufen von: <https://appinventor.mit.edu/explore/teach>.

EINORDNUNG IN GESETZLICHE RAHMENBEDINGUNGEN

Das Online-Modul ist für den Einsatz in der Mittelstufe konzipiert und weist in den folgenden Bereichen eine Nähe zum **Kernlehrplan für die Sekundarstufe I Gymnasium in Nordrhein-Westfalen für Informatik als Wahlpflichtfach**¹ auf.

Die Kernkompetenz des „angestrebten selbstständigen informatischen Problemlösens“² wird gefördert durch ...

- den eigenverantwortlichen Umgang der Lernenden mit der Entwicklungsumgebung des MIT App Inventor 2.
- das Durchlaufen eines kompletten Softwarekonstruktionsprozesses (Design und Funktion), an dessen Ende eine fertige und funktionsfähige App steht, die die Teilnehmenden nach ihren eigenen Vorstellungen gestaltet haben.

Die Kompetenzbereiche

Argumentieren: Innerhalb der Teams diskutieren die Teilnehmenden ihre Entwürfe und Lösungsansätze und verständigen sich auf eine gemeinsame Vorgehensweise, um ihre ersten eigenen Apps mit Hilfe des MIT App Inventor 2 zu programmieren.

Modellieren und Implementieren: Die Teilnehmenden lernen eine informatische Denkweise einzunehmen, indem sie einfache Algorithmen entwerfen und mit Hilfe des MIT App Inventor 2 zu funktionierenden Programmen zusammenpuzzeln. Das Ergebnis wird auf dem Android-basierten Endgerät oder dem Emulator erlebt und überprüfbar.

Kommunizieren und Kooperieren: Die Schülerinnen und Schüler nutzen Zoom als geeignetes Werkzeug zur Zusammenarbeit trotz physischer Distanz. Sie kooperieren, indem sie in Zweierteams gemeinsam an ihren App-Projekten arbeiten.

¹ Ministerium für Schule und Bildung des Landes Nordrhein-Westfalen (2019): *Kernlehrplan für die Sekundarstufe I Gymnasium in Nordrhein-Westfalen – Wahlpflichtfach Informatik*. Abgerufen von: https://www.schulentwicklung.nrw.de/lehrplaene/lehrplan/211/g9_wpif_klp_%2034241_2019_06_23.pdf.

² Ebd. S. 13.

Die Inhaltsfelder

Inhaltsfeld 2 – Algorithmen: Durch das Experimentieren mit den vordefinierten Programmierblöcken des MIT App Inventor 2 erkennen die Schülerinnen und Schüler, dass Computer Anweisungen in anderer Form als Menschen benötigen und sämtliche Informatiksysteme auf von Menschen entworfenen Handlungsschritten, also Algorithmen, basieren. „Sie erfahren, dass ein Algorithmus eine genaue Beschreibung von Handlungsschritten zur Lösung eines Problems ist, die von einem ‚Prozessor‘ [...] ausgeführt werden kann.“³ Die Lernenden entwerfen und realisieren solche Algorithmen mit Hilfe der algorithmischen Grundbausteine und stellen diese geeignet in der Programmierung dar.

Inhaltsbereich 4 – Informatiksysteme: Die Schülerinnen und Schüler wenden den MIT App Inventor 2 und ggf. ein Android-basiertes Endgerät im Sinne eines Informatiksystems zielgerichtet an.

DIDAKTISCHE/METHODISCHE SCHWERPUNKTSETZUNG

BESCHREIBUNG DES MODULABLAUFS MIT METHODISCHER/DIDAKTISCHER BEGRÜNDUNG

Das Modul setzt sich aus vier größeren Phasen zusammen. In die erste Phase fällt die Begrüßung der Schülerinnen und Schüler, eine kurze Vorstellungsrunde sowie eine knappe Präsentation des Schülerlabors. Denn auch wenn die Teilnehmenden sich während der Moduldurchführung nicht unmittelbar in den Räumlichkeiten des Schülerlabors aufhalten, soll ihnen Sinn und Zweck des InfoSpheres und dessen Einbettung innerhalb der RWTH Aachen University kurz vermittelt werden. In diesen Kontext fällt auch die Vorstellung der Betreuenden, die die Teilnehmenden eben nicht als Lehrkräfte, sondern als Tutorinnen und Tutoren verstehen sollen. Die Vorstellungsrunde der Teilnehmenden untereinander wird kombiniert mit einer Einführung in relevante Funktionen von Zoom: Die Teilnehmenden werden in zufälligen Zweierteams – den Zweierteams, die später auch an gemeinsamen App-Projekten arbeiten werden – unterschiedlichen Breakout Sessions zugeordnet und erhalten ein Zeitlimit von fünf Minuten, um einander gegenseitig vorzustellen. Nach Ablauf dieser fünf Minuten kehren sie automatisch wieder in den Hauptraum zurück.

Dort erfolgt in der zweiten Phase mittels Screensharing eine interaktive Einführung in das zweite Tool, den MIT App Inventor 2. Ziel ist es, den Teilnehmenden den Einstieg in die Arbeit mit dem Editor zu erleichtern und ihre Motivation zu steigern. Darüber hinaus steht während der Programmierung stets ein Infoblatt zu den beiden großen Bereichen des MIT App Inventor 2 (Designer und Blocks-Editor) als Zusammenfassung zur Verfügung.

Nach dieser zentralen Einführung beginnen die Schülerinnen und Schüler in der dritten Phase mit der eigentlichen Programmierung. Sie kehren in die Breakout Sessions vom Anfang zurück und erarbeiten dort zu zweit eine erste, sehr einfache Hallo-InfoSphere-App, um den Umgang mit dem Editor zu üben. Nach diesem Austesten des Tools folgt die Beschäftigung mit dem Hauptprojekt des Moduls: der Programmierung einer Zeichenapp. Diese App soll ihren Usern ermöglichen, über die Kamera ein Foto aufzunehmen und dieses dann zu bearbeiten. Dazu sollen unterschiedliche Farben und Stiftbreiten sowie ein Wischer zur Verfügung stehen. Um dieses Ziel zu erreichen, das den Teilnehmenden von Beginn an transparent gemacht wird, erhalten die Jugendlichen kleinschrittige Arbeitsblätter, die sie anleiten und ihre Arbeit gleichzeitig strukturieren. Außerdem gibt es bei den Arbeitsblättern zur Erstellung der Zeichenapp eine Differenzierung: Zum einen gibt es Arbeitsblätter für diejenigen, die mit einem mobilen Android-basierten Endgerät ihre App testen. Zum anderen gibt es jedoch auch Arbeitsblätter, die die Arbeit mit dem Emulator anleiten. Auf diese Weise soll sichergestellt werden, dass auch diejenigen ihre App erfolgreich testen können, die über kein eigenes mobiles Android-basiertes Endgerät verfügen. Das Vorgehen ist bei beiden Wegen stets zweigeteilt: Zunächst wird das Layout im Designer gestaltet; an dieser Stelle ist die Kreativität der

³ Ebd., S. 15.

Teilnehmenden gefragt. Anschließend werden im Blocks-Editor die Funktionen der App programmiert, wobei die Jugendlichen immer wieder ganz automatisch auf informatische Konzepte stoßen. Auf diese Weise erarbeiten die Teilnehmenden ihr theoretisches Grundwissen stets in direkter Verbindung mit der praktischen Anwendung. Durch die Vernetzung von theoretischem Wissen und seiner praktischen Anwendung wird neben einer besseren Behaltensleistung auch angestrebt, dass die Jugendlichen ihre neuerworbenen Kompetenzen später auch auf andere Programmierumgebungen oder -sprachen transferieren können. Außerdem können die Teilnehmenden während des Programmierens die Fortschritte ihrer App unmittelbar auf ihrem eigenen Android-Smartphone oder Tablet bzw. in dem detailgetreuen Emulator beobachten. Dies steigert nicht nur ihre Motivation, sondern ermöglicht zugleich eine direkt erfahrbare und durch die Teilnehmenden selbstgesteuerte Rückmeldung über Fortschritte, aber auch über mögliche Fehler, an deren Behebung es noch zu arbeiten gilt. Auch diese Art der Fehlerbehebung ist prägender Bestandteil der Arbeit innerhalb der Informatik. Für schnelle Entwicklerteams (Binnendifferenzierung) stehen Bonusaufgaben zur Verfügung: Die Teilnehmenden haben die Wahl zwischen der Programmierung zweier kleiner Spiele. Bei MoleMash entwickeln sie ein Spiel, in dem ein kleiner Maulwurf an zufälligen Positionen des Bildschirms erscheint und von dem Spielenden angetippt werden muss, damit er/sie Punkte erhält. Bei FlappyBird gilt es, einen kleinen Vogel an Gewitterwolken vorbeizusteuern.

In der abschließenden vierten Phase haben die Entwicklerteams die Gelegenheit, über das Screensharing ihre Apps den anderen Teilnehmenden vorzustellen und auf diese Weise Feedback und Anerkennung zu erhalten. Den Abschluss bildet eine Gruppendiskussion über Erfahrungen mit der neuen Form des Arbeitens, Schwierigkeiten und Verbesserungsvorschlägen.

DIDAKTISCHE PRINZIPIEN NACH BAUMANN UND HUBWIESER

Die Prinzipien, die bei der Gestaltung des Moduls besonders berücksichtigt wurden, sind die Prinzipien der Lebensnähe/Aktualität, des aktiven Lernens, der Zielvorstellung, der Strukturierung, der Kreativitätsförderung, des individuellen Lerntempos sowie das Spiralprinzip.

Durch den Stellenwert von Smartphones und Tablets im Alltag der Schülerinnen und Schüler ist die **Lebensnähe und Aktualität** bereits auf thematischer Ebene gegeben. Auch kann davon ausgegangen werden, dass die Schülerinnen und Schüler in ihrer Freizeit (und wahrscheinlich auch in der Schule) ganz selbstverständlich die verschiedensten Apps verwenden, um miteinander zu kommunizieren, Fotos zu bearbeiten, zu lernen oder zu spielen. Im Rahmen dieses Moduls wird ihnen ermöglicht, hinter die Fassade dieser kleinen, beliebten Anwendungen zu blicken. Dabei setzen sie sich nicht nur mit den Funktionen der App auseinander, sondern auch mit deren Design, sodass ihre **Kreativität** gefördert wird. Zudem wird die Kreativität der Schülerinnen und Schüler gefördert, weil sie bei der Programmierung den Freiraum haben, eigene Bearbeitungsfunktionen hinzuzufügen oder bereits vorhandene nach ihren eigenen Vorstellungen zu erweitern.

Um dem **Prinzip des aktiven Lernens** gerecht zu werden, wird von Anfang an spielerisches, entdeckendes Lernen ermöglicht, da es sich bei dem MIT App Inventor 2 um einen Online-Editor handelt und die Teilnehmenden somit an ihrem heimischen PC aktiv programmieren können. Zwar erhalten die Teilnehmenden eine grundlegende Einführung in den Aufbau des Editors, den eigentlichen Umgang erlernen sie jedoch hauptsächlich durch das Bearbeiten der Aufgaben. Ihre Lösungen können sie sofort testen, sodass sie unmittelbar Rückmeldung über ihren Fortschritt und auch über mögliche Fehler erhalten. Dadurch können die Teilnehmenden von Beginn an **selbstständig** und **eigenverantwortlich** arbeiten.

Strukturiert wird das Vorgehen der Teilnehmenden durch die detaillierten Arbeitsmaterialien, die immer nach dem gleichen Muster (erst Design, dann Funktion) aufgebaut sind. Auf diese Weise soll eine Überforderung vermieden werden. Auch wird zu Beginn eines neuen Arbeitsschrittes stets das **Ziel transparent** gemacht, sodass die Teilnehmenden wissen, was von ihnen verlangt wird.

Außerdem wurde bei der Erstellung der Arbeitsmaterialien größter Wert auf das **individuelle Lern-tempo** der Schülerinnen und Schüler gelegt. Die Teams können sich entweder alleine auf die Konstruktion der Zeichenapp konzentrieren, oder aber sie arbeiten im Anschluss weiter an einem von zwei kleinen Spielen. Ihre Programmierung fordert die Teilnehmenden noch einmal mehr, da sie weniger präzise angeleitet ist. Auf diese Weise werden lernstärkere Schülerinnen und Schüler gefördert und zugleich die Frustration derer vermieden, die möglicherweise mehr Zeit benötigen, da auch sie am Ende eine fertige, funktionierende App (Zeichenapp) vorweisen können.

Bei dem MIT App Inventor 2 handelt es sich um eine blockbasierte Entwicklungsumgebung, die einen einfachen, frustfreien Einstieg in die App-Programmierung ermöglicht. Allerdings durchlaufen die Lernenden durchaus einen kompletten Softwareentwicklungsprozess, wenn sie sich mit Design und Funktionen der App beschäftigen. Auf diese Wissensbasis lässt sich zu einem späteren Zeitpunkt im Sinne des **Spiralprinzips** aufbauen, denn wer einmal den groben Aufbau eines Programms in der Block-Programmierung verinnerlicht hat, versteht auch leichter den Aufbau textueller Programmiersprachen. Dabei folgt bereits das Modul selbst dem Spiralprinzip, da die Schülerinnen und Schüler zunächst die Handhabung des MIT App Inventors erlernen, sie anschließend das Prinzip der blockbasierten Programmierung nachvollziehen und schließlich die darin enthaltenen informatischen Konzepte (wie z. B. If-Anweisungen) unter Verwendung der Blöcke selbstständig anwenden.

EINBETTUNG IN DEN UNTERRICHT

Das Modul kann als Einstieg in die Programmierung genutzt werden – auch dann, wenn Präsenzunterricht gerade nicht möglich ist.

Im weiteren Verlauf bietet der MIT App Inventor 2 diverse Vertiefungsmöglichkeiten, beispielsweise zur Nutzung von Schleifen oder Listen.

Ebenso bietet sich die Bearbeitung des Moduls „Spielend Programmieren lernen mit Scratch an“, um den Lernenden eine zweite blockbasierte Entwicklungsumgebung vorzustellen.

VERLAUFSPLAN DES MODULS

Zeit	Phase	Inhalt	Medien	Sozialform	Teillernziel
0:00 - 0:15	Begrüßung	Vorstellung InfoSphere und Betreuerteam und ggf. der Teilnehmenden untereinander, Transparentmachung der Modulinhalte, Vorstellung Zoom	PC, Internetzugang, Zoom, Zoom Breakout Sessions	UG	Kennenlernen; grundlegende Kompetenz im Umgang mit einer Videokonferenz-Software
0:15 - 0:30	Einstieg	Vorstellung des MIT App Inventor 2 sowie der grundlegenden Funktionen dieses Tools	PC, Internetzugang, Zoom, Zoom Screen-sharing, MIT App Inventor 2	LV → UG	Grundlegende Kompetenz im Umgang mit einer blockbasierten Entwicklungsumgebung
0:30 - 1:15	Erarbeitung I	Erstellen der ersten App <i>HalloInfoSphere</i> , Ansteuern der Kamera über die App, Einbinden eines Hintergrundes, Programmierung eines Buttons	PC, Internetzugang, MIT App Inventor 2, Zoom, Zoom Breakout Sessions, Blatt_1, ggf. Tablet	PA	Kompetenter Umgang mit den beiden Bereichen (Designer und Block-Editor) des MIT App Inventor 2 zur Erstellung einer ersten App
1:15 - 1:35	Pause				
1:35 - 3:15	Erarbeitung II	Erstellen der ersten Zeichenapp, Ansteuern der Kamera, Einbinden eines Hintergrundes, Programmierung mehrerer Buttons zur Farbänderung, Festlegung der Stiftbreite etc.	PC, Internetzugang, MIT App Inventor 2, Zoom, Zoom Breakout Sessions, Blatt_2 / Blatt_2_Emulator, Blatt_3 / Blatt_3_Emulator, ggf. Tablets	PA	Anwendung grundlegender informatischer Konzepte (Variablen, if-Anweisungen, Screen Arrangements etc.) zur Programmierung der ersten Zeichenapp
3:15 - 3:30	Pause				
3:30 - 4:00	Eventualphase	Erstellung eines Android-Spiels	PC, Internetzugang, MIT App Inventor 2, Zoom, Zoom Breakout Sessions, Blatt_4a_MoleMash / Blatt_4b_FlappyBird, ggf. Tablets	PA	Bündelung des bisher Gelernten zur Erstellung komplexerer Programme

4:00 – 4:30	Sicherung	Präsentation der Apps der Schülerinnen und Schüler, abschließende Diskussion der gemachten Erfahrungen, Verabschiedung	PC, Internetzugang, MIT App Inventor, Zoom, Zoom Screensharing	Sicherung der neugelernten Kompetenzen
------------------------	-----------	--	--	--

Legende: UG – Unterrichtsgespräch, LV – Lehrervortrag, PA – Partnerarbeit

QUELLENVERZEICHNIS

Ministerium für Schule und Bildung des Landes Nordrhein-Westfalen (2019): Kernlehrplan für die Sekundarstufe I Gymnasium in Nordrhein-Westfalen – Wahlpflichtfach Informatik. Abgerufen von: https://www.schulentwicklung.nrw.de/lehrplaene/lehrplan/211/g9_wpif_klp_%2034241_2019_06_23.pdf.

ABBILDUNGSVERZEICHNIS

Modul- und Titelbild – Quelle: Screenshot des Emulators des MIT App Inventors 2, <http://ai2.appinventor.mit.edu/> (Creative Commons Attribution-ShareAlike 3.0)

MODULCHECKLISTE

Im Vorfeld erhalten die Teilnehmenden alle Modulmaterialien (s. u.) und werden darauf hingewiesen, dass sie den AI Starter ebenfalls vor der Moduldurchführung installieren sollen. Bei Problemen bei der Installation können sie sich an das InfoSphere-Team wenden: infosphere-support@informatik.rwth-aachen.de

Auch erhalten sie einen Link zu einem Zoom-Raum. Zoom können sie entweder über eine App verwenden oder über den Browser.

TROUBLESHOOTING/PROBLEMBEHEBUNG

PROBLEME BEI DER VERWENDUNG DES EMULATORS

- Vereinzelt kann es zu **Verbindungsabbrüchen** kommen, ohne dass diese angezeigt werden. Dabei übernimmt der Emulator keine Änderungen mehr, die im MIT App Inventor 2 durchgeführt werden. Hier bietet es sich an, die Verbindung einfach zu resettten. Gegebenenfalls muss der AI Starter neugestartet werden.
- Manchmal möchte der MIT App Inventor 2 ein **Update** durchführen. Führt man dieses dann durch, gibt er die folgende Meldung aus: „The update has not been installed.“ Dann fragt er erneut nach einem Update. In diesem Fall ist es möglich, dass kein Update existiert; das Update kann abgelehnt werden, ohne dass es zu Komplikationen kommt.
- Der Emulator ist deutlich langsamer als ein Android-basiertes Endgerät (unabhängig von der Leistung des Computers, auf dem der Emulator läuft). In diesem Fall kann es sinnvoll sein, die Spiele MoleMash und FlappyBird langsamer zu designen. Dies muss allerdings für jedes Gerät individuell bestimmt werden. Dazu müssen die Zeiten der Clock verändert werden.
- Damit die **Bilder** im Emulator genutzt werden können, ist es nötig, die Bilder in den Emulator zu übertragen. Sollte ein Bild hochgeladen werden und nicht im ImagePicker erscheinen, kann es sein, dass der Emulator neugestartet werden muss. Abhängig vom Dateiformat des Bildes steht beim Start im AI Starter eine Meldung, die in etwa den Inhalt „Datei erfolgreich übertragen“ hat. Erst jetzt kann das Bild gewählt werden. (Manchmal sind **zwei** Neustarts nötig; mehr bis jetzt jedoch noch nie.)
- Der Emulator kann nur **einen Touchinput** gleichzeitig verarbeiten; das heißt, dass selbst bei einem Gerät mit Touchscreen nur ein Finger verwendet werden kann.
- Sollte sich der **AI Starter** mehrfach **aufhängen**, lässt sich das Problem nicht unbedingt beheben, indem man den AI Starter einfach neustartet. Häufig hilft es, den AI Starter über den Taskmanager zu beenden, den Browser, über den der MIT App Inventor 2 läuft, zu schließen und dann **zuerst den AI Starter** und dann den Browser zu starten.
- Bei leistungsschwächeren Geräten kann es nötig sein, alle anderen Tabs im Browser zu

schließen. **Es sollten nie zwei App-Inventor-Tabs offen sein** (dabei können Session Errors entstehen).

- Sollte der Emulator **nicht starten**, kann dies an den Einstellungen des Browsers liegen. Googlechrome zeigt keinerlei Probleme und empfiehlt sich daher. Bei Firefox traten vermehrt Probleme auf.
- Nutzer mit **macOS-Betriebssystem** müssen besonders darauf achten, das korrekte Setup zu verwenden.

VORGEHEN BEIM AUFTRETEN EINES UNBEKANNTEN BUGS

Sollte ein unbekannter Bug auftreten, empfiehlt sich folgendes Vorgehen. Nach jedem Punkt gilt es, zu überprüfen, ob das Problem behoben ist.

- Neustart des AI Starters
- Schließen des Browsers und AI Starters, Öffnen des AI Starters, **dann** Öffnen des Browsers und Schließen aller anderen Tabs
- Neustart des Computers
- Neuinstallation des AI Starters (mit geschlossenem Browser!), **dann Neustart vor erstem Start** des AI Starters

MODULMATERIALIEN

ARBEITSBLÄTTER

- Blatt_0_Einrichten_MIT_AppInventor
- Blatt_0_Infoblatt_Designer_Blocks-Editor
- Blatt_1
- Blatt_2
- Blatt_2_Emulator
- Blatt_3
- Blatt_3_Emulator
- Blatt_4aMoleMash
- Blatt_4bFlappyBird

DOWNLOAD DES AI STARTERS

- Downloadseite des [AI Starters](#)
- Achtung: Es muss die Version ausgewählt werden, die zum Betriebssystem passt.

MIT APP INVENTOR 2

- [MIT App Inventor 2](#)
- Achtung: Man braucht ein Gmail-Konto, um sich anmelden zu können.