

Schere, Stein, Papier

Entwicklung einer eigenen App mit dem MIT App Inventor

Lara Penz & Stefan Hickl



Übersicht

1	Über den MIT App Inventor	1
2	Vorstellung der Projektidee	1
3	Motivation	1
4	Zeitlicher Umfang	2
5	Benötigte Materialien	2
6	Voraussetzungen & Lernziele	2
7	Grober Zeitplan	4
8	Hinweise	5
9	Beschreibungen der Apps.....	6



1 Über den MIT App Inventor

Auf der Seite <http://appinventor.mit.edu/> können Apps für Android-Geräte in einer blockbasierten Sprache programmiert werden. Dafür wird ein Google-Account benötigt. Ohne Google-Account lässt sich der App Inventor auch auf der Seite <http://code.appinventor.mit.edu/> nutzen. Auf die Projekte kann in diesem Fall jedoch nur mit einem individuellen Code zugegriffen werden. Sicherheitshalber sollten die SchülerInnen ihre Projekte daher bei der Nutzung dieser Seite immer auf dem Rechner speichern. Ob der App Inventor mit den vorhandenen Android-Geräten funktioniert, sollte vorher getestet werden. Die Nutzung des Emulators auf dem PC ist zu zeitaufwendig und frustrierend für die SchülerInnen.

2 Vorstellung der Projektidee

In diesem Projekt lernen die SchülerInnen den grundlegenden Umgang mit dem *MIT App Inventor* kennen, indem sie ein eigenes Spiel programmieren. Dafür wird im ersten Abschnitt mit einer einfachen Variante des Spiels *Schere, Stein, Papier* begonnen. Diese wird in den folgenden Einheiten modular erweitert. Im zweiten Abschnitt können die SchülerInnen das Projekt beispielsweise um eine selbstgewählte Umsetzung eines Multiplayer-Modus erweitern.

3 Motivation

Die Einheit motiviert insbesondere durch ihre Alltagsrelevanz. Apps werden von den SchülerInnen in ihrer Freizeit auf dem Smartphone genutzt. In dieser Einheit können die SchülerInnen selbstständig eine App entwickeln, wobei durch den App Inventor der Rahmen ausreichend beschränkt ist, um auch SchülerInnen mit wenig Programmiererfahrung nicht zu frustrieren.

Das Spiel *Schere, Stein, Papier* besitzt einfache und leicht zu implementierende Spielregeln. Dadurch können die SchülerInnen mit der ersten App direkt ein Erfolgserlebnis erzielen, das sie für weitere Programmier-Projekte motiviert.

Außerdem wird Motivation durch das anschließende Spielturnier geschaffen, in dem die selbst entwickelten Apps genutzt werden. Durch die modulare Erweiterbarkeit des Spiels werden auch sehr gute SchülerInnen gefordert und motiviert.

4 Zeitlicher Umfang

~ 4 Doppelstunden (ein möglicher grober Zeitplan wird in Kapitel 7 vorgestellt)

5 Benötigte Materialien

Hardware

- Mindestens ein Computer je 2 SchülerInnen
- Mindestens ein Android-Gerät je 2 SchülerInnen

Software

- Browser auf dem Computer
- MIT AI2 Companion App auf den Android-Geräten

6 Voraussetzungen & Lernziele

6.1 Voraussetzungen

Vorausgesetzt werden grundlegende Programmierkenntnisse in einer blockbasierten Programmiersprache (z.B. Scratch, Snap, Programmierung des Calliopes mit dem makecode-Editor). Variablen, Schleifen und Verzweigungen sollten bereits bekannt und in der Programmier-Praxis verwendet worden sein.

6.2 Beinhaltete Grundkonzepte der Informatik

- Variablen
- Schleifen und Verzweigungen
- Unterprogramme
- Ereignisorientierte Programmierung

6.3 Zentrale Lernziele

Die SchülerInnen können nach Abschließen der Einheit...

- eine einfache App eigenständig mit dem MIT App Inventor programmieren (L01)
- Apps auf Android-Geräten installieren (L02)

- Variablen/Schleifen/Verzweigungen/Unterprogramme zielführend in einem größeren Projekt verwenden (L03)
- verwendete informatische Konzepte in existierenden Apps erkennen (L04)
- eine nutzbare App designen (L05)
- eigene Ideen verständlich und strukturiert präsentieren (L06)
- die Funktionalitäten von Apps verständlich beschreiben (L07)
- Funktionalitäten von Apps überprüfen (L08)
- Apps bewerten (L09)
- Feedback in Software-Projekte einarbeiten (L10)

6.4 Mögliche Verortungen im Lehrplan

Im Bildungsplan zu IMP Klasse 8 (Baden-Württemberg) werden zum Thema Algorithmen folgende Lernziele formuliert, die das Projekt abdeckt. Die SchülerInnen können ...

- (2) Zufallszahlen in eigenen Programmen verwenden (zum Beispiel, um Würfelergebnisse zu simulieren oder einen Spielverlauf abwechslungsreicher zu gestalten)
- (5) *Unterprogramme* verwenden, um Programmcode zu strukturieren und redundanten Code zu vermeiden
- (6) Anforderungen an Programme oder Programmteile beschreiben, dazu geeignete Testfälle entwerfen und die Implementierungen dagegen testen
- (7) ein kleines Softwareprojekt (zum Beispiel Spiel, Smartphone-App, Robotik, Simulation) unter Anleitung durchführen

7 Grober Zeitplan

Doppelstunde	Lernziele	Arbeitsschritte	Zeit (min)	Materialien
1	L01-L05	Einführung in den MIT App Inventor durch die Lehrkraft, Vorführen der App als Motivation	10	Hardware Computer Android-Geräte Software
		SchülerInnen entwickeln eine einfache Schere, Stein, Papier - App	70	MIT AI2 Companion
		SchülerInnen testen die App in einem gemeinsamen Turnier	10	Apps: SSP-Random-Single.apk
2	L01-L05 L06	Lehrkraft zeigt mögliche Umsetzungen zu Multiplayer-Varianten (Zufällig auf einem Gerät, wie bei der SSP-Random-Multi-App; manuelle Eingabe und Weitergeben, wie bei der SSP-Manual-Multi-App)	10	Hardware Computer Android-Geräte Software MIT AI2 Companion
		SchülerInnen beginnen, die App um einen Multiplayer-Modus zu erweitern	70	Apps: SSP-Random-Multi.apk
		Design-Zwischenpräsentation	10	SSP-Manual-Multi.apk

3	L01–L05 L07	Lehrkraft stellt mögliche Erweiterungen der App vor (z.B. Spielen gegen den Computer)	10	Hardware Computer Android-Geräte
		Fortsetzung des Arbeitsauftrags der Vorstunde, schnelle SchülerInnen können die vorgestellten Erweiterungen oder eigene Ideen implementieren	60	Software MIT AI2 Companion Apps: SSP-Trainer.apk
		SchülerInnen erstellen Spielanleitung für ihr Spiel	20	
4	L01–L05 L08–L10	SchülerInnen testen Apps der anderen Gruppen auf den Android-Geräten an Stationen, füllen Feedback-Bogen aus	45	Hardware Computer Android-Geräte Software
		SchülerInnen arbeiten Verbesserungs- und Erweiterungsvorschläge ein	45	MIT AI2 Companion

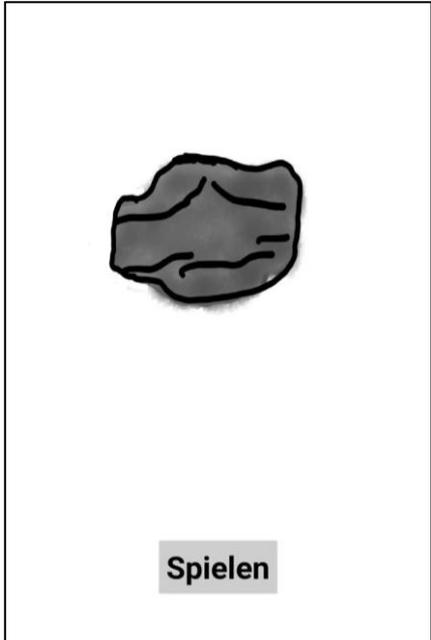
8 Hinweise

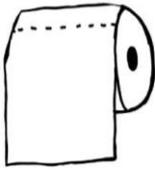
- Während der Entwicklung kann die App *MIT AI2 Companion* genutzt werden. Fertige Apps können als apk-Datei gespeichert werden und auf dem Android-Gerät installiert werden.
- Zu den Apps werden jeweils eine apk- und eine aia-Datei bereitgestellt. Die apk-Datei kann direkt auf einem Android-Gerät installiert werden. Die aia-

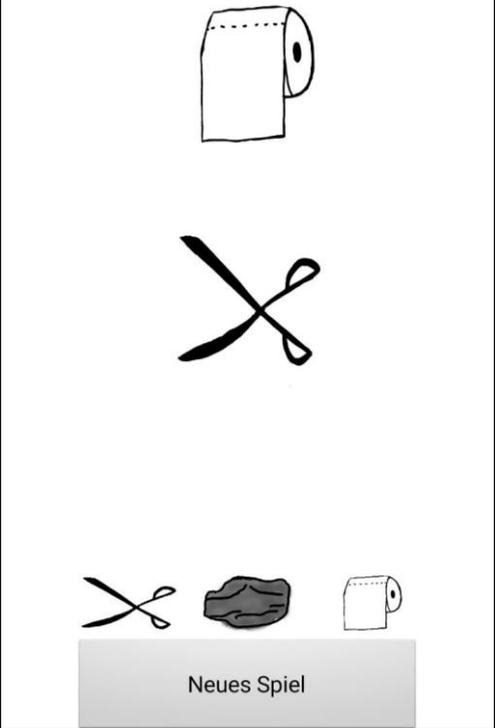
Datei kann zur Betrachtung des Quellcodes der App auf der App-Inventor-Seite importiert werden.

- Der **Name der App** muss im App Inventor festgelegt werden. Eine Umbenennung der apk-Datei beeinflusst diesen nicht.
- Bilder müssen beim Befehl ‚set Picture to‘ inklusive Dateieindung angegeben werden (‚set Picture to „bild1.png““ oder ‚set Picture to „bild1.jpg““ anstelle von ‚set Picture to „bild1““).
- Wird ein farbiger Hintergrund verwendet, können **Bilder** im png-Format **mit transparentem Hintergrund** verwendet werden, um einen weißen Rahmen um die Bilder zu vermeiden. Solche Bilder lassen sich beispielsweise mit dem Programm *Paint 3D* erstellen. Dazu muss ein transparenter Zeichenhintergrund gewählt werden.
- Schnelle SchülerInnen können ihre Apps mit weiteren Symbolen ausschmücken oder zusätzliche Spielelemente wie eine Liste für die Symbole oder Highscores einbauen. Optional kann bei der Trainer-App (siehe Seite 8) gespeichert werden, welche Symbole vom Spieler oft genutzt werden und dementsprechend die Spielstrategie des Computers verbessert werden.

9 Beschreibungen der Apps

Name	Screenshot	Beschreibung
SSP Random Single		In dieser einfachen Variante wird beim Drücken des Spielen-Buttons zufällig ein Gegenstand gewählt und angezeigt.

Name	Screenshot	Beschreibung
SSP Random Multi	<p style="text-align: center;">Spieler 1: Fertig</p> <div style="text-align: center;">  </div> <p style="text-align: center;">Spieler 2 gewinnt Nächstes Spiel</p> <div style="text-align: center;">  </div> <p style="text-align: center;">Spieler 2: Fertig</p>	<p>Spieler 1 und Spieler 2 drücken den Spielen-Button.</p> <p>Dadurch wird ein zufälliger Gegenstand gespielt.</p> <p>Haben beide Spieler gespielt, wird das Ergebnis ausgewertet (siehe Screenshot).</p>
SSP Manual Multi	<p>Spiel: 10 Punkte Spieler 1: 2 Punkte Spieler 2: 8</p> <div style="text-align: center;">  </div> <div style="text-align: center;">  </div> <div style="display: flex; justify-content: center; gap: 10px;">    </div> <div style="text-align: center; margin-top: 10px;"> Neues Spiel </div> <p style="text-align: center; font-weight: bold; margin-top: 5px;">Der Gewinner ist Spieler 2</p>	<p>Spieler 1 trifft seine Wahl verdeckt und übergibt dann das Gerät an Spieler 2.</p> <p>Nachdem dieser seine Wahl getroffen hat, wird das Spielergebnis angezeigt (siehe Screenshot).</p>

Name	Screenshot	Beschreibung
SSP Trainer	<p data-bbox="448 297 544 324">Spiel: 17</p> <p data-bbox="842 297 938 324">Punkte: 2</p> 	<p data-bbox="975 297 1345 383">Der Spieler trifft hier selbstständig seine Wahl.</p> <p data-bbox="975 405 1345 539">Der Gegenstand des Gegenspielers/Computers wird zufällig gewählt.</p>