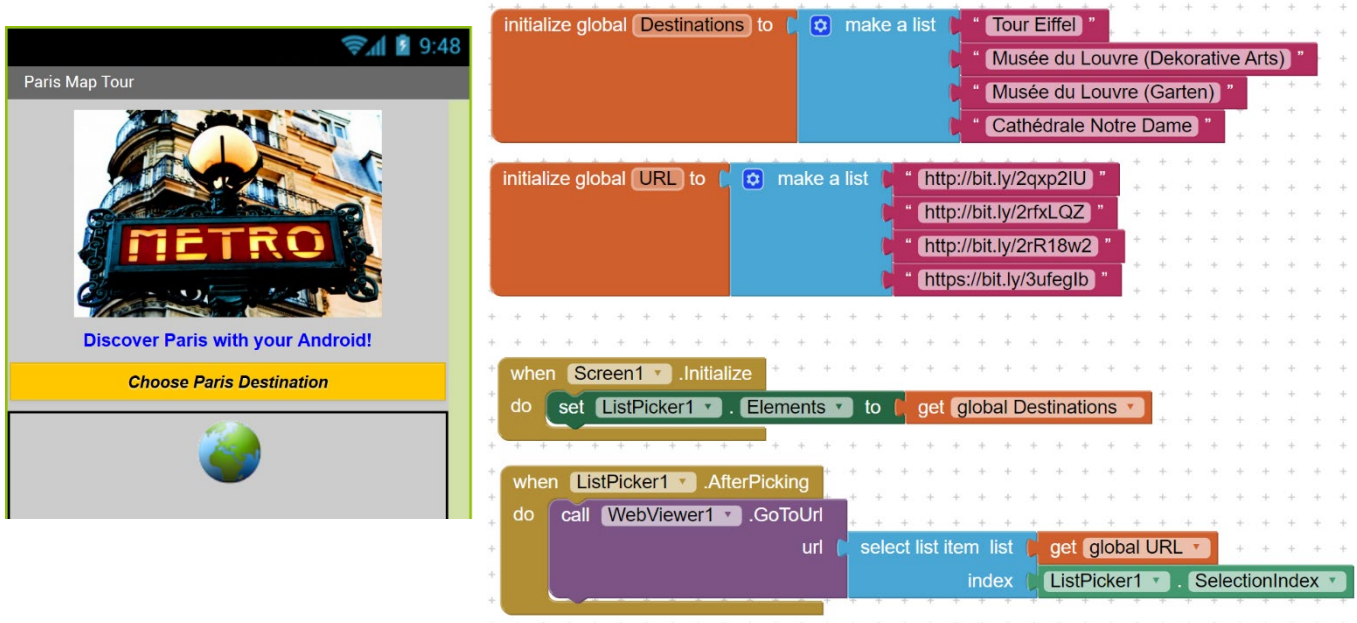


(4) Paris Map Tour

Als nächstes probieren wir die folgende Aufgabe: eine fertige App importieren, verstehen und erweitern!



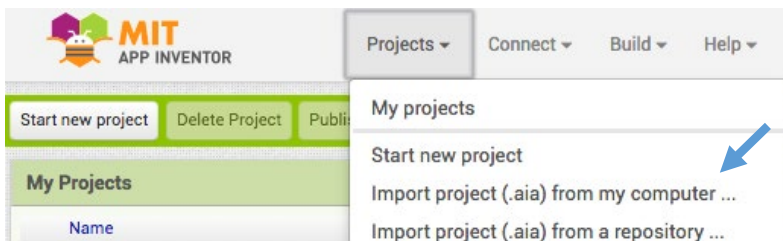
The image shows the app interface on the left and the corresponding App Inventor code blocks on the right. The app interface includes a title bar 'Paris Map Tour', a background image of a Paris street scene with a 'METRO' sign, the text 'Discover Paris with your Android!', a yellow button 'Choose Paris Destination', and a globe icon.

The code blocks are as follows:

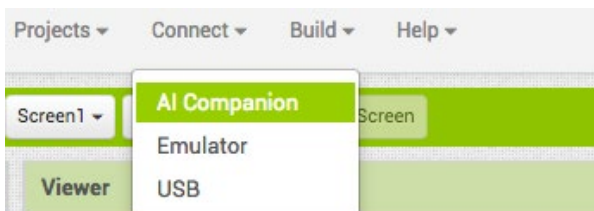
- initialize global Destinations to make a list** with items: "Tour Eiffel", "Musée du Louvre (Dekorative Arts)", "Musée du Louvre (Garten)", "Cathédrale Notre Dame".
- initialize global URL to make a list** with items: "http://bit.ly/2qxp2IU", "http://bit.ly/2rfxLQZ", "http://bit.ly/2rR18w2", "https://bit.ly/3ufeglb".
- when Screen1.Initialize do** set ListPicker1.Elements to get global Destinations.
- when ListPicker1.AfterPicking do** call WebViewer1.GoToUri with url selected list item list and index ListPicker1.SelectionIndex.

Schritt 1) Paris-Map Tour App importieren, ausprobieren und Code verstehen.

Logge Dich in AppInventor2 ein und wähle aus dem Menü *Projects > Import project (.aia)*



Offene die App und teste sie auf deinem Handy/Tablet mit dem AI Companion



Zwei Components in Code sind neu:
List und WebViewer.

Diese werden jetzt von Workshop
Helpers erklärt.

Kontrolliere, ob Du diese zwei neue Komponente gut verstanden haben:

- Wie kriert man eine Liste? Wie kann man eine Elemente aus der Liste wählen und weiter benutzen?
- Was macht der Webviewer Component?

Schritt 2) App für die eigene Schule (oder Reise/Anlass) umprogrammieren

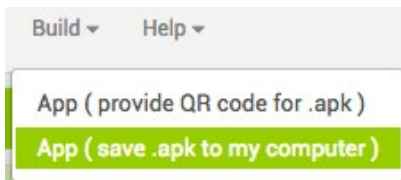
Programmiere die App/Map so um, dass sie z.B. 4 Standorte deiner Schule anzeigt:

1. Kindergarten
2. Schulhaus
3. Turnhalle
4. Schulverwaltung

Ändere den Text auf Label1 und das Image1 Bild, damit es zum Inhalt passt. Wenn Du genug Zeit hast, versuche weitere Orte hinzuzufügen und/oder das App-Icon zu ändern.

Schritt 3) Installation auf Handy

Speichere das Projekt als *.apk Datei auf deinem Desktop. Sende die Datei per E-Mail an dein Telefon/Tablet. Öffne die Mail und klicke auf den Anhang. Danach wirst du gefragt, ob du es installieren möchtest. Mit der Installation wird das Icon auf dem Handy hinzugefügt.



Schritt 4) Präsentationen

Erkläre einem anderen Teilnehmenden des Workshops die Neuprogrammierung deiner App. Für welche Aufgabe konntest du diese App nutzen? Wo hast du Hilfe von den Workshop-HelferInnen gebraucht?

Die originelle detaillierte Anweisung für das «Paris Map App» Tutorial findet man unten:

<http://www.appinventor.org/bookChapters/chapter6.pdf>

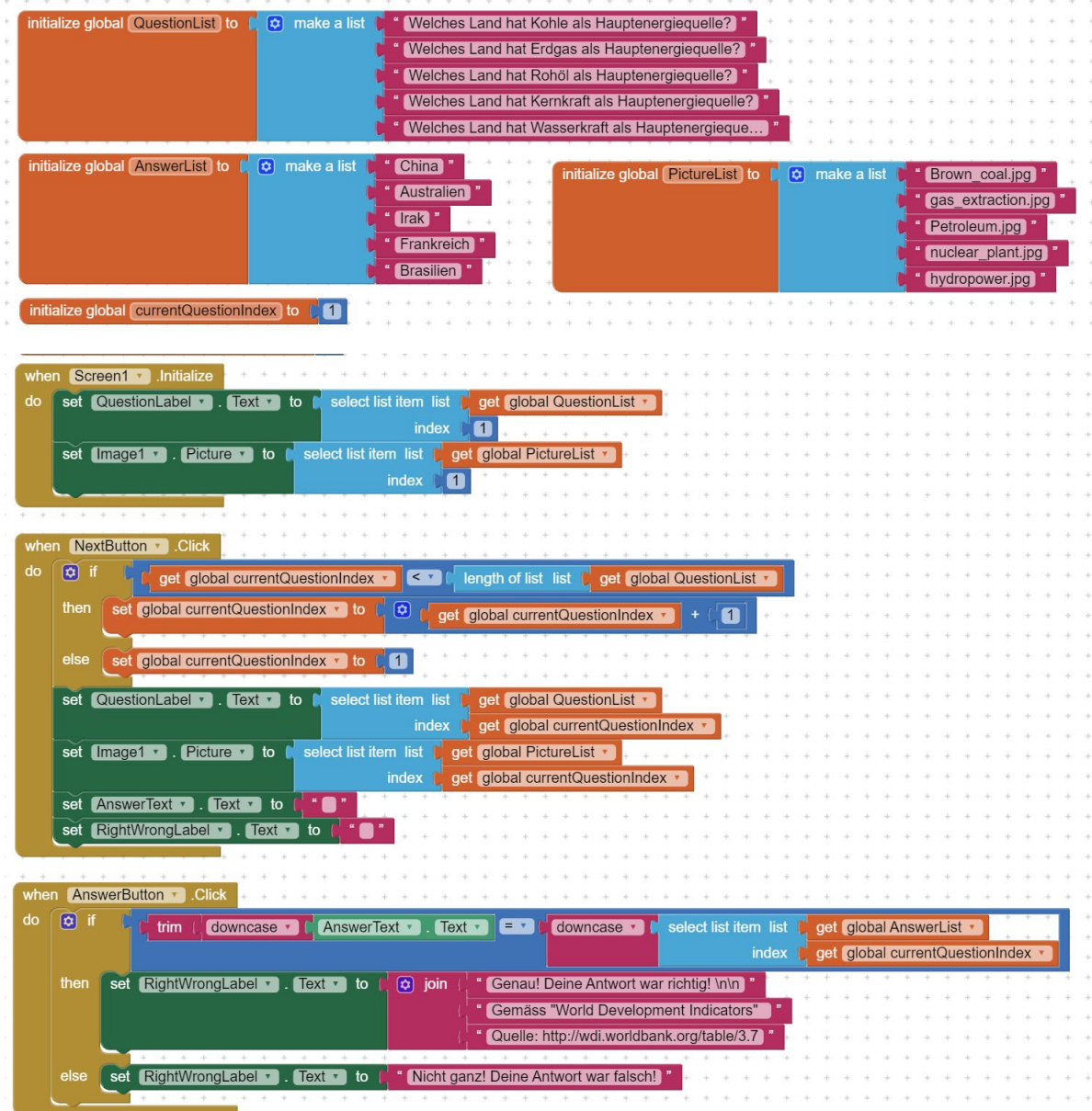
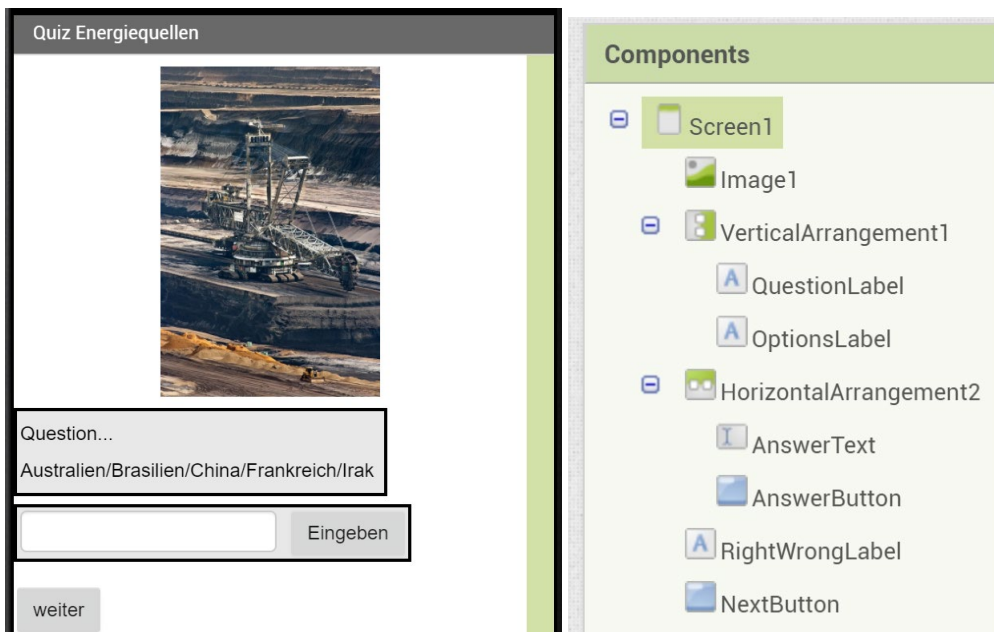
Wenn Du noch Zeit hast, schau Dir im nächsten Abschnitt eine coole Quiz-App an, die auch mit Listen funktioniert.

Schritt 5) Energy Quizz App importieren, ausprobieren und Code verstehen.

Importiere die Quelldatei (*.aia) der EnergyQuizzApp, verbinde dich mit deinem mobilen Gerät und probiere die App aus. Kannst du dir vorstellen, welche Ressourcen für die Programmierung dieser App benötigt werden? Können z.B. Listen hilfreich sein?

Schau dir nun den Code an und beantworte die folgenden Fragen, um die Implementierung Schritt für Schritt zu verstehen:

- a) Welche Variablen sind in diesem Code definiert?
- b) Welche Variablen sind vom Typ Liste? Und welche sind es nicht?
- c) Wie lang sind die definierten Listen? Sind sie alle gleich lang?
- d) Wofür wird die Variable `currentQuestionIndex` verwendet?
- e) Was passiert, wenn die App gestartet wird? Welche Elemente werden initialisiert?
- f) Was passiert, wenn die BenutzerIn den `NextButton` drückt?
- g) Was passiert, wenn die BenutzerIn alle Fragen gesehen hat und `NextButton` erneut drückt?
- h) Was passiert, wenn die BenutzerIn auf den `AnswerButton` drückt?



Die originale detaillierte Anweisung für das «Quiz App» Tutorial findet man unten:

<http://www.appinventor.org/bookChapters/chapter8.pdf>