

Laborbuch von



Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG



Vorbemerkungen

Du brauchst einen google-Account für die Anmeldung beim App Inventor ai2.appinventor.mit.edu.

Das Badewannenbild zum Nachbauen der App findest du auf www.lightandschools.de unter 'Versuche Mittelstufe' 'App-Programmierung'.

Faszination Apps

Wie unglaublich praktisch, diese Smartphones. Wie viele Apps hast du auf deinem Gerät? Zu eigentlich jedem nur erdenklichen Zweck können wir uns die Apps runterladen, umsonst oder gegen Geld. Aber was, wenn doch mal eine Anwendung fehlt?

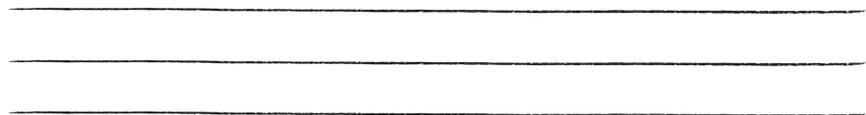


Einfach selber machen

Eine eigene App zu schreiben, kommt den meisten Menschen nicht unbedingt in den Sinn, dabei ist es nicht besonders schwierig. Mit dem App Inventor kann jede und jeder, auch völlig ohne Programmierkenntnisse, eine eigene App schreiben. Das ist ein hervorragender Weg, um deine eigenen Ideen umzusetzen und ganz nebenbei auch grundsätzlich ein Gefühl dafür zu bekommen, wie Computerprogramme funktionieren/denken.

Was für eine App würdest du gerne schreiben? Skizziere deine Ideen hier, damit du sie nicht vergisst.

Grundidee:



Um deine Ideen umzusetzen möchte ich mit dir zusammen zunächst eine ganz einfache Quiz-App entwickeln.

Anhand dieser App stelle ich dir den App Inventor und seine Möglichkeiten vor. Wenn du die Quiz-App geschafft hast, dann könntest du im Prinzip auch jede andere App programmieren. Für Inspirationen und auch Anleitungen schau doch mal auf ai2.appinventor.mit.edu  oder in der weiten Welt von YouTube. 

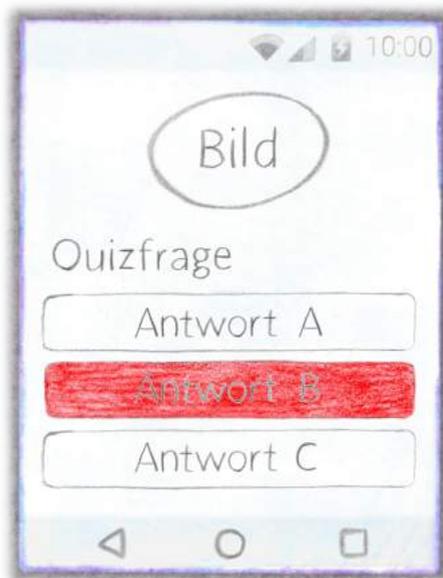
Die Quiz-App

Grundidee:

- Eine Quiz-Frage wird mit verschiedenen Antwortmöglichkeiten gezeigt.
- Beim Klicken auf die Antworten wechselt der Antwortknopf die Farbe, grün für die richtige Antwort, rot für die falsche.
- Extra: Bei der richtigen Antwort wird noch ein kurzer Erklärungstext angezeigt.

Design:

- Ein schönes Bild zur Illustration der Frage (und weil Bilder immer alles schöner machen)
- Darunter die Quizfrage und drei mögliche Antwortknöpfe, die dann entsprechend die Farbe wechseln.

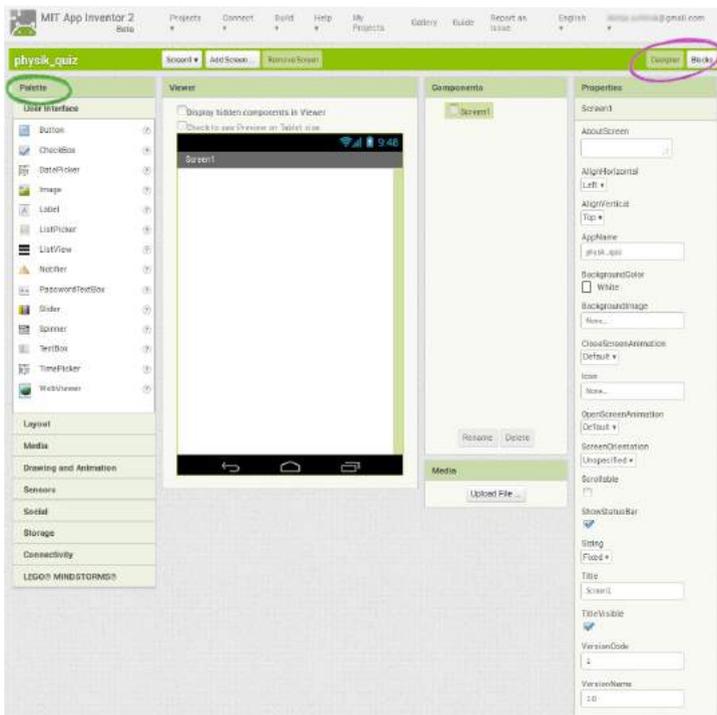


Umsetzung

Um aus diesem Design nun eine App zu machen, nutze ich den App Inventor des MIT (Massechusetts Institute of Technologie) ai2.appinventor.mit.edu.

Ich melde mich mit meinem google-Account (kann auch ein Fake-Account sein) an und eröffne unter 'Projects' ein neues Projekt und nenne es `physik_quiz`. Der Projektname darf keine Leerzeichen enthalten!

Danach lande ich auf einem Schirm, der wie folgt aussieht:



Rundgang über die Seite

Oben rechts (lila Kringel ) kann ich zwischen 'Designer' und 'Blocks' umschalten. Du siehst, dass wir uns hier gerade auf der 'Designer'-Seite befinden. Also da, wo das Design der App festgelegt wird. Unter 'Blocks' werden die Funktionen der App festgelegt.

In der Mitte ist im 'Viewer' der leere Bildschirm (Screen) eines Smartphones dargestellt. Hier kann ich jetzt das Layout meiner App erstellen.

Die benötigten Komponenten wie z.B. die Buttons, Text oder Bilder finde ich auf der 'Palette' am linken Rand (grüner Kringel ). Von dort kann ich sie ganz einfach auf den leeren Bildschirm ziehen.

Rechts neben dem Screen siehst du eine Komponentenliste, die gerade bis auf den Screen selbst noch leer ist. Hier behalte ich den Überblick über alle verwendeten Komponenten, kann sie umbenennen und auch wieder löschen.

Noch weiter rechts lassen sich die Eigenschaften der einzelnen Komponenten ändern. Dem Screen kann ich hier zum Beispiel eine andere Hintergrundfarbe geben.

Unter der Komponentenliste ist noch ein kleines Feld 'Media' zu sehen. Hier werden alle Dateien hochgeladen, die für die App benötigt werden.

Quizfrage

Jetzt aber los. Eine Quiz-App. Dafür brauche ich erstmal eine spannende Quizfrage. Zum Beispiel:

Wann läuft das Wasser aus der Badewanne schneller ab?



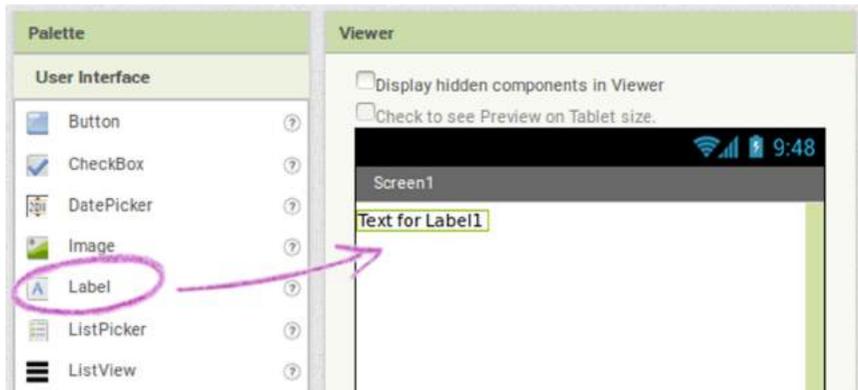
- A: Es geht besonders schnell, wenn du noch selber in der Wanne sitzt.
- B: Es geht schneller, wenn sich nur das Wasser aber nicht du in der Wanne befindet.
- C: Völlig egal, es geht immer gleich schnell.

Und nein, ich will jetzt nicht von dir die Antwort haben.
Stell dir lieber vor, wie diese Quizfrage in dem von mir vorgeschlagenen App-Design aussieht.

Design nachbauen

Textfeld - Label

Um das Design von Seite 7 im App Inventor nachzubauen beginne ich mit der Quizfrage, also einem einfachen Textfeld. Dafür ziehe ich mir von der 'Palette' aus dem Unterpunkt 'User Interface' das 'Label' auf den 'Screen'.

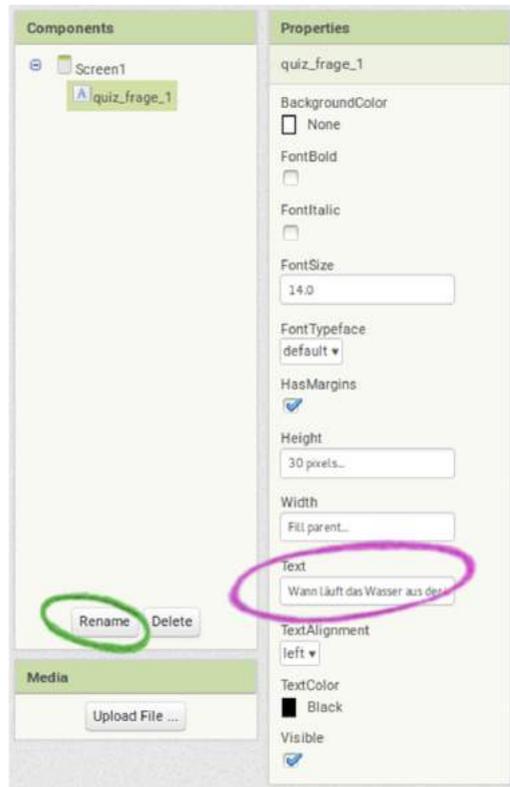


Ein 'Label' kann einen Text auf dem Bildschirm anzeigen.

Eigenschaften des Labels, wie Textfarbe (TextColor), Hintergrundfarbe (BackgroundColor), Schriftgröße (FontSize), etc. kann ich rechts unter 'Properties' einstellen.

Den Text selbst, also die Quizfrage, kann ich ebenfalls unter 'Properties' 'Text' (lila Kringel auf der nächsten Seite) eingeben. Er erscheint dann direkt in dem 'Viewer'.

Ich habe das 'Label' unter 'Components' 'Rename' (grün) umbenannt, damit ich später, wenn ich z.B. fünf verschiedene Textfelder habe, immer genau weiß um welches es sich jeweils handelt. Überlegt euch also einen guten, eindeutigen Namen! Hier kann man überflüssige Komponenten übrigens auch wieder löschen. Das sollte aber wohl durchdacht sein, denn eine 'Zurück'-Taste gibt es nicht.



Auch die Größe des Textfensters wird unter 'Properties' festgelegt. Für die Höhe habe ich 30 Pixel angegeben, so passen genau zwei Textzeilen rein. Für die Breite habe ich 'Fill parent...' gewählt. Da heißt, das Textfenster ist immer genauso lang wie der Bildschirm. Wenn das Handy also z.B. von der Vertikalen in die Horizontale gekippt wird, dann würde sich die Breite des Textfensters entsprechend anpassen. Dieses Verhalten einer App nennt man 'responsive', also auf unterschiedliche Bildschirmgrößen reagierend.



Schaltfläche – Button

Um die Antwortknöpfe zu generieren, habe ich ebenfalls den 'Button' auf den Bildschirm gezogen und die Eigenschaften unter 'Properties' eingestellt. Versucht das am besten selber einmal! Im Bild auf Seite 14 könnt ihr meine Einstellungen nachschauen, wenn ihr Hilfe braucht.

Components

- Screen1
 - quiz_frage_1
 - antwort_a_richtig**
 - antwort_b_falsch
 - antwort_c_falsch

Rename Delete

Media

Upload File ...

Properties

antwort_a_richtig

BackgroundColor
 None

Enabled

FontBold

FontItalic

FontSize
14.0

FontTypeface
default ▾

Height
50 pixels...

Width
Fill parent...

Image
None...

Shape
rounded ▾

ShowFeedback

Text
Es geht besonders schnell, wenn

TextAlignment
left ▾

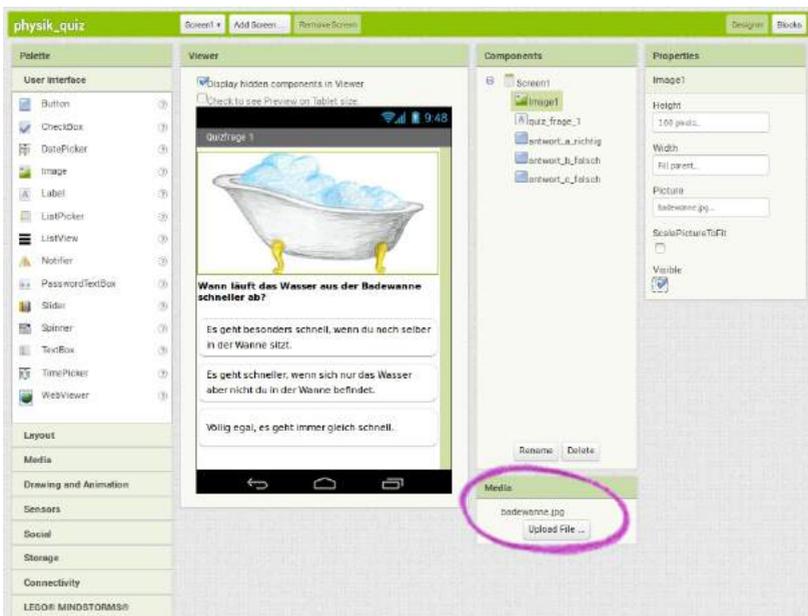
TextColor
Default

Visible

Bild – Image

Das sieht doch schon nach einem brauchbaren Layout aus, nicht wahr? Fehlt eigentlich nur noch ein hübsches Bild, das die Quizfrage illustriert.

Alle Mediendateien, also z.B. Bilder, Musik, Filme, etc., die für eine App gebraucht werden, müssen vorher hochgeladen werden. Um mein Badewannenbild also hochzuladen, klicke ich unter 'Media' (direkt unter der Komponentenliste 'Components') auf 'Upload File...', suche über 'Browse...' das Badewannenbild in den Ordnern auf meinem Computer und klicke 'OK'. Anschließend wird das Bild unter 'Media' aufgelistet.



Nun suche ich in der Palette 'Image' heraus und ziehe es auf den Screen an die gewünschte Stelle. In den 'Properties' kann ich nun unter 'Picture' aus der Liste der hochgeladenen Mediendateien ein passendes Bild aussuchen. Bisher ist da nur mein Badewannenbild zu finden.

Damit ist mein Layout nun vollständig! Aber es passiert natürlich noch nichts, wenn ich auf die Buttons drücke. Und ohne Funktion hätte ich bis jetzt auch einfach ein beliebiges Zeichenprogramm benutzen können.

Funktionen festlegen

Um nun die Schaltflächen auch mit Funktionen zu belegen, wechsle ich oben rechts im ApplInventor von 'Designer' zu 'Blocks'.

Es ist eine große, weiße Fläche zu sehen und links davon eine Liste mit den zu Verfügung stehenden Blocks. Bei den Blocks seht ihr unter den 'Built-in' wieder die Komponenten, die wir soeben im 'Designer' angelegt haben. Auch die Mediendateien könnt ihr hier wieder finden.

Jetzt wisst ihr auch, warum es so wichtig war, die Komponenten sinnvoll zu benennen. Sonst würde ich jetzt nicht mehr wissen, welche der drei Antworten eigentlich die richtige war.

Als Funktion meiner App möchte ich nun gerne, dass sich beim Klicken auf den Button 'antwort_a_richtig' die Hintergrundfarbe des Buttons von weiß zu grün ändert, da ja die richtige Antwort ausgewählt wurde.

when-Anweisungen

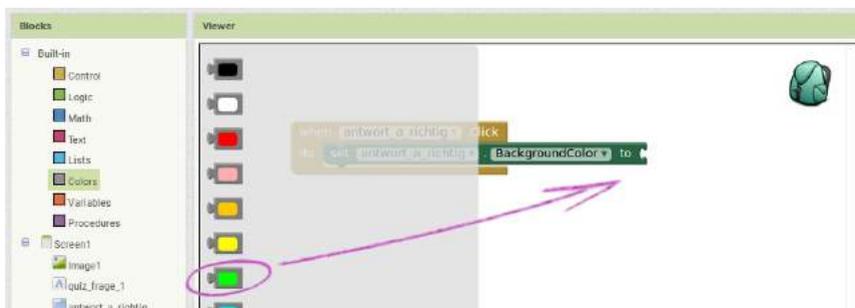
Ich wähle also unter 'Blocks' den Button 'antwort_a_richtig' aus. Es erscheint eine Reihe von Blocks, die alle zu dem Button 'antwort_a_richtig' gehören. Den Block 'when antwort_a_richtig.Click' ziehe ich auf die große weiße Fläche. Mit diesem Block kann ich definieren, was passieren soll, wenn 'antwort_a_richtig' geklickt wird. Diese Anweisungen kommen in die Lücke in dem Block. Hier passen nur Teile rein, die auch sinnvolle Anweisungen enthalten.



Die Anweisung, die ich geben möchte (Änderung der Hintergrundfarbe der Schaltfläche) hat wieder etwas mit der Schaltfläche selbst zu tun. Daher klicke ich wieder unter 'Blocks' auf 'antwort_a_richtig' und wähle nun den Block 'set antwort_a_richtig.BackgroundColor to'. Diesen ziehe ich in die Lücke des 'when-Blocks'.

Weitere mögliche Anweisungen findet ihr, wenn ihr die Blocks runter scrollt.

Zuletzt muss ich jetzt noch die Hintergrundfarbe auswählen, die der Button annehmen soll. Dafür wähle ich bei den 'Built-in' Blocks 'Colors' aus und ziehe die grüne Farbe auf die weiße Fläche. Dort kann ich das 'grün' nun an die richtige Stelle einklicken.



Die entstandene when-Anweisung ändert nun also immer dann, wenn die Schaltfläche 'antwort_a_richtig' geklickt wurde, die Hintergrundfarbe zu grün.

Testen

So weit, so gut. Für meine fertige App müsste ich jetzt nur noch die anderen beiden Antwort-Button in gleicher Weise programmieren.

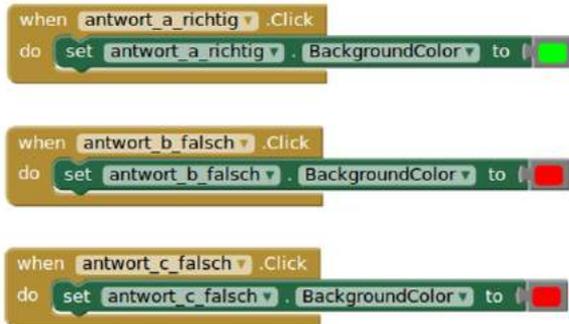
Bevor ich das mache, möchte ich aber erst einmal testen, ob meine App bisher alles so macht, wie ich mir das vorstelle.

Das Testen ist beim Programmieren und somit auch bei der App-Entwicklung eine der wichtigsten Aufgaben. Am Besten wird nach jedem kleinen Schritt getestet. Ist das Programm erst einmal fertig geschrieben, ist es wirklich schwer einen Fehler zu finden, den man vielleicht schon ganz zu Anfang gemacht hat. Und Fehler macht man IMMER. Selbst bei den Profis nimmt die Fehlersuche in der Regel einen Großteil der Zeit ein.

Das Testen ist am einfachsten, wenn du dein Smartphone oder Tablet direkt mit dem Computer, an dem du arbeitest, verbindest. Wenn der Computer das gleiche W-Lan nutzt wie dein mobiles Gerät, dann ist dafür der AI Companion die beste Wahl. Lade aus dem Play-Store die AI Companion App herunter. Anschließend wählst du im App Inventor unter 'Connect' 'AI Companion' aus und folgst den Anweisungen.

Es gibt auch die Möglichkeit, dein mobiles Gerät per USB-Kabel mit dem Computer zu verbinden. Oder wenn du kein mobiles Gerät zur Verfügung hast, kannst du auch ein kleines Programm herunterladen, das dir einen Smartphonebildschirm simuliert (Emulator). Diese Optionen liest du am besten auf der App Inventor Seite nach. Dort gibt es einfache Anleitungen.

Ist dein Smartphone verbunden, so kannst du die App, an der du arbeitest, immer live testen. Funktioniert der Antwortknopf A korrekt? Falls ja, kannst du jetzt auch die anderen entsprechend programmieren.



Erweiterung

Jetzt fände ich es noch schön bei der richtigen Antwort nicht nur die Hintergrundfarbe zu grün zu wechseln, sondern auch eine kurze Erklärung zu der richtigen Antwort zu geben.

Ich möchte also, dass sich auch der Text ändert zu 'Richtig! Durch den höheren Wasserspiegel ist auch der Druck am größten.' Kannst du das programmieren?

Fertige App

Nun müssen wir die App noch auf unser Smartphone/Tablet bekommen. Dafür klickst du oben in der grauen Zeile des App Inventors auf 'Build' und wählst 'App (provide QR code for .apk)' aus. Jetzt dauert es eine Weile bis dir ein QR code angezeigt wird. Diesen musst du nun mit deinem mobilen Gerät scannen und die App auf dein Gerät laden und installieren. Eventuell musst du vorher noch die Einstellungen deines Geräts so ändern, dass Apps aus fremden Quellen auf deinem Gerät zugelassen sind.

Ausblick

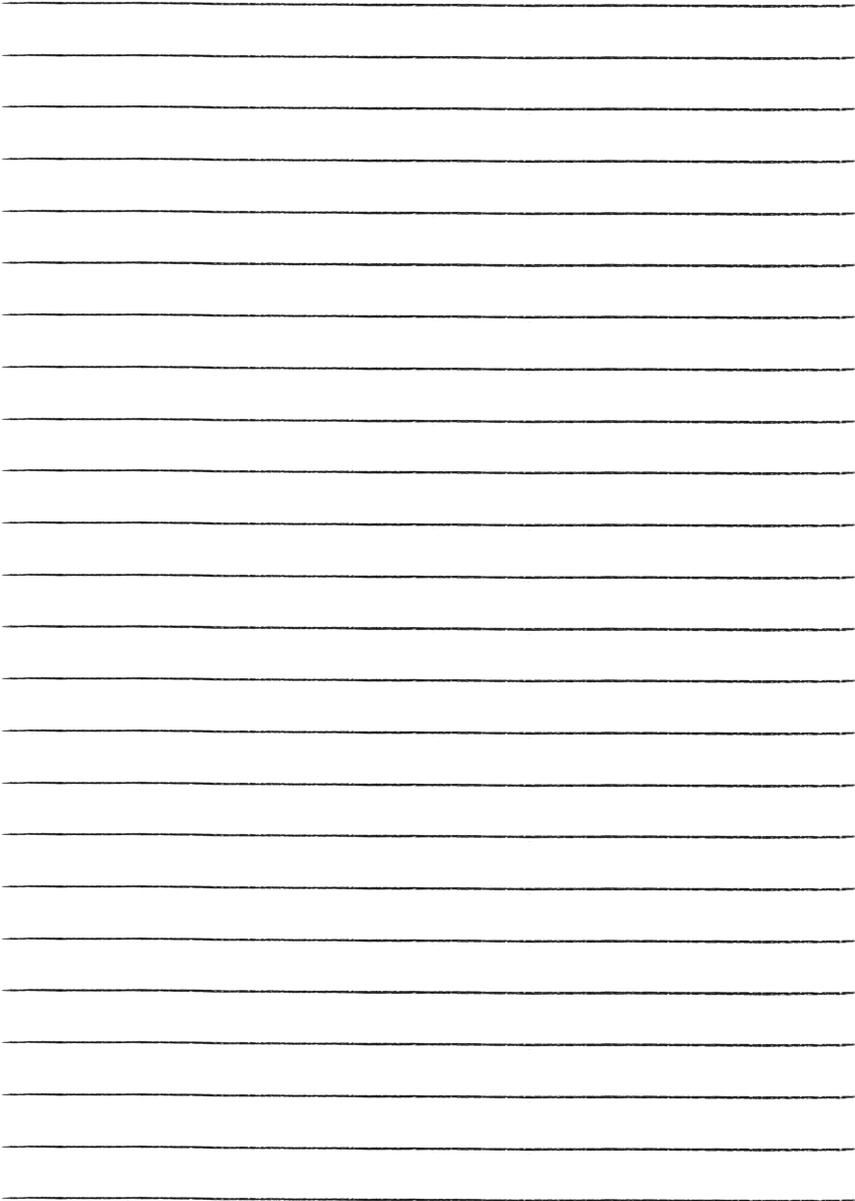
Jetzt haben wir eine wunderbare App mit einer Quizfrage. Schön wäre jetzt natürlich, noch mehr Fragen zu haben. Jede auf einem eigenen Screen. Und am besten werden auch noch Punkte für richtige Antworten gezählt. Mit einer kleinen google-Recherche oder durch Ausprobieren wirst du sicher herausfinden, wie das geht.

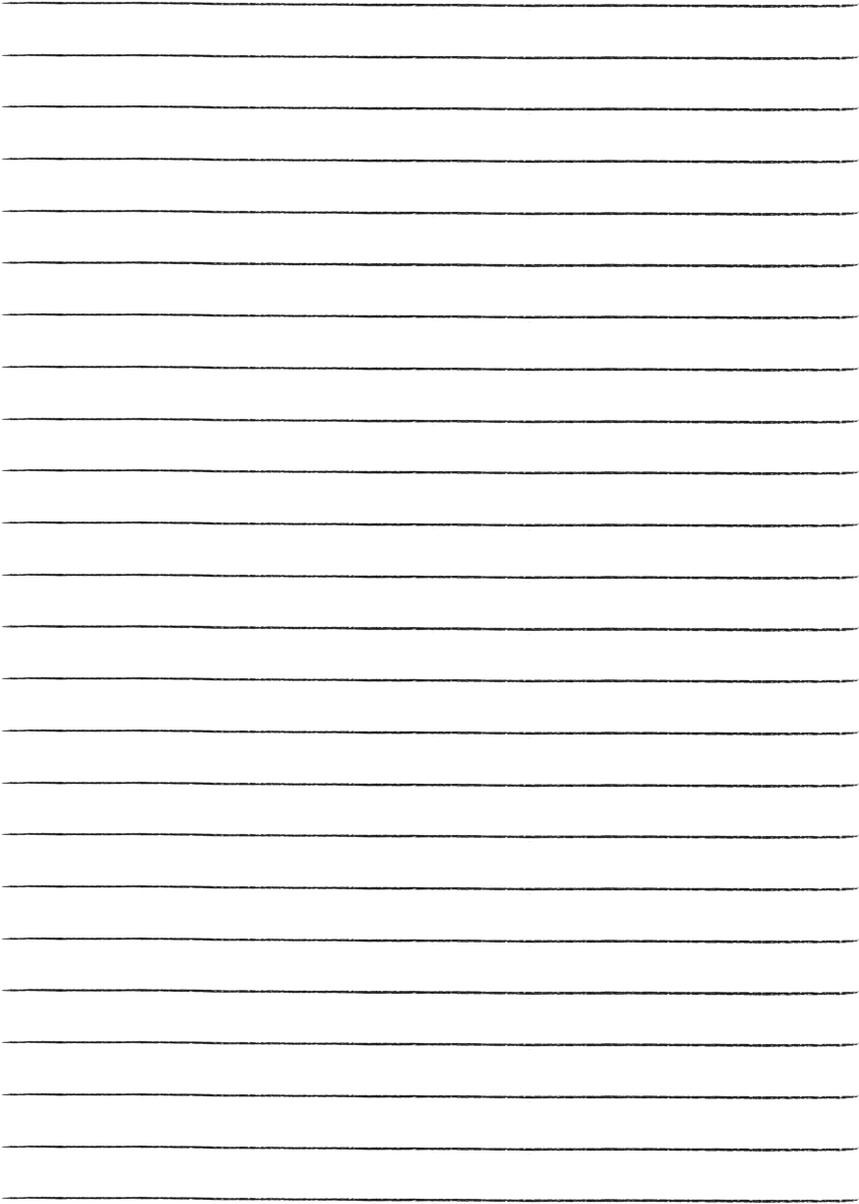
Wenn nicht, schreib mir und frag mich!
dortje.schirok@uni-hamburg.de

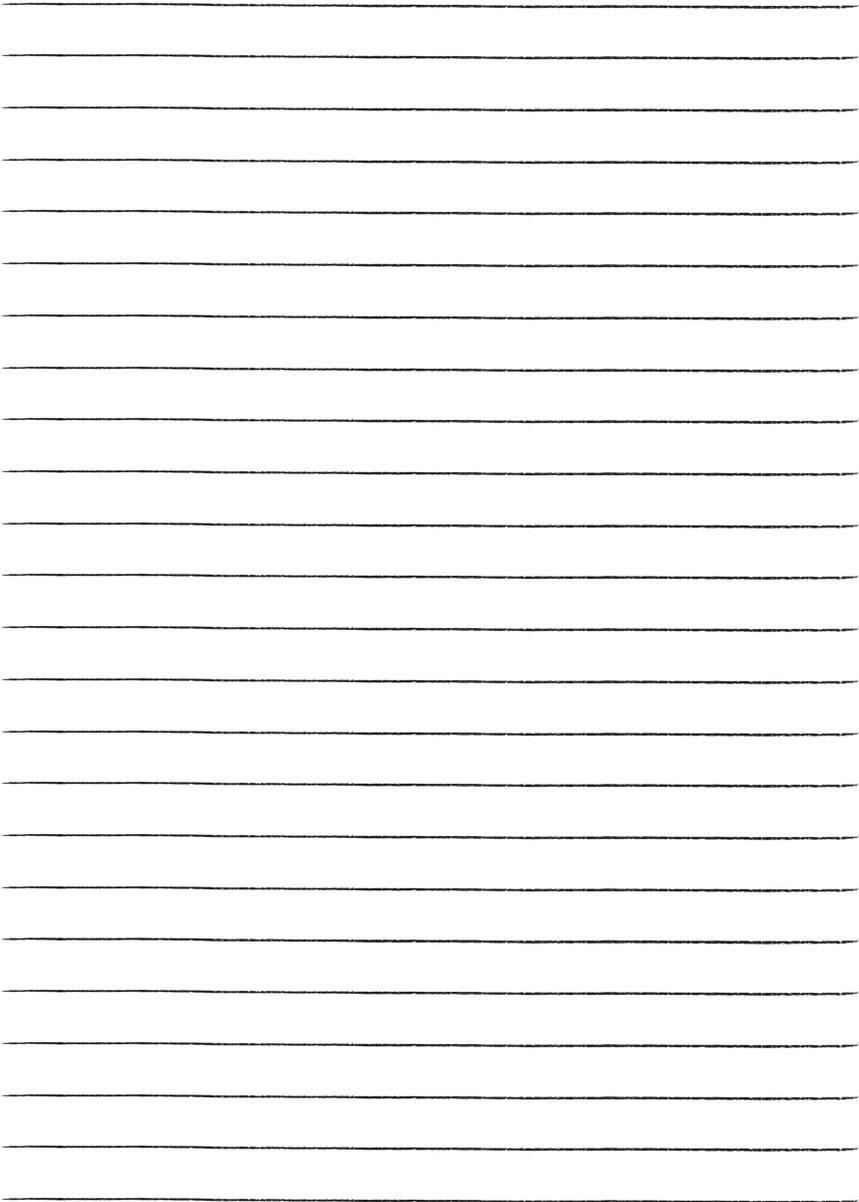
Die Physik dahinter

Und was hat das Ganze nun mit Physik zu tun? Nun ja, abgesehen davon, dass wir, natürlich, eine Physik-Quizfrage ausgesucht haben, spielt das Thema Programmieren in der Physik eine bedeutende Rolle. Egal, ob es um Simulationen geht oder Steuerungssoftware für die Versuche im Labor, die benötigten Programme sind in der Physik meist so speziell, dass sie von den Physikerinnen und Physikern selber geschrieben werden. Ohne Programmieren also heutzutage keine Physik mehr.

Steuerungssoftware ist jetzt aber nicht die spannendste Aufgabe, daher starten wir erst einmal mit einer App!







IMPRESSUM:

Universität Hamburg, Light & Schools (www.lightandschools.de)

Text, Bilder, Layout: Dortje Schirok (dortje.schirok@uni-hamburg.de)

Inspiriert durch App Camps (www.appcamps.de)