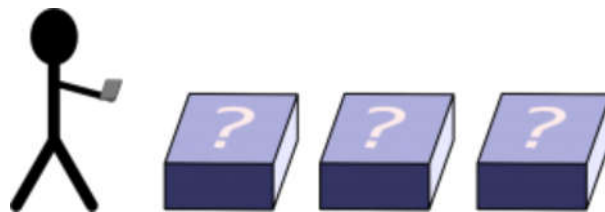


App Quest 2019

Metall Detektor Android

Unsere erste Aufgabe ist es, einen einfachen Metalldetektor zu programmieren. Das gehört zur Grundausrüstung eines jeden Schatzsuchers und bietet uns einen einfachen Einstieg in das Thema der App Programmierung. Die App soll es uns ermöglichen, einen versteckten Metallgegenstand zu finden. Dazu verwenden wir den magnetischen Sensor des Smartphones, welcher sonst für den Kompass gebraucht wird. Befindet sich Metall in der Nähe des Sensors, dann liefert der Sensor stärkere Werte, und das nutzen wir in unserer App aus.



In welcher Kiste befindet sich wohl der Metallgegenstand?

Als Beispiel liesse sich die Stärke des magnetischen Feldes mit einem Fortschrittsbalken ([Android](#)) anzeigen, aber auch andere Darstellungen sind denkbar (vielleicht als Graph?).

Projekt Erstellen

Als erstes solltest du ein neues Projekt erstellen. Sprich dich gleich mit deinem Coach ab, welches minimum SDK ihr verwenden müsst, damit die App auf all euren Geräten läuft. Generell gilt: je höher die Version, desto mehr Features von Android könnt ihr einsetzen. Da wir die App nur für uns entwickeln, verwenden wir einfach die höchstmögliche Einstellung. Ihr könnt vom Wizard auch direkt eine Main Activity erstellen lassen.

Jetzt habt ihr bereits eine Activity Klasse und ein einfaches Layout. Genug, um die Applikation ein erstes Mal auf euren Geräten zu deployen. Wenn bei eurem Gerät der Debug-Modus aktiviert ist (das ist leider von Gerät zu Gerät unterschiedlich, Google hilft weiter!) und per USB verbunden ist, dann könnt ihr die Applikation einfach starten und sie wird automatisch installiert. Alternativ gibt es die Möglichkeit, den Simulator zu verwenden. Da

wir an der App Quest aber viel mit Sensoren und der Kamera arbeiten, ist ein Simulator nicht unbedingt die beste Wahl. Euer Coach kann euch zeigen, wie man den Simulator installiert.

Sensordaten Auslesen

Als nächstes können wir uns um die Sensordaten kümmern. Der `TYPE_MAGNETIC_FIELD` Sensor liefert uns 3 Werte (wg. 3 Achsen), indem wir den [Betrag dieses Vektors berechnen](#) erhalten wir einen Wert der aussagt, wie stark das Feld ist. Oder anders ausgedrückt: wie nah wir an einem Magneten sind.

```
float[] mag = event.values;
double betrag = Math.sqrt(mag[0] * mag[0] + mag[1] * mag[1] + mag[2] * mag[2]);
```

Füge deinem Layout einen Fortschrittsbalken hinzu und zeige den aktuellen Betrag als Balken dar. Um dem Fortschrittsbalken mitzuteilen, was der höchste Wert ist den wir anzeigen können (sozusagen die 100%), können wir die `setMax` Methode der Klasse `ProgressBar` verwenden. Im Prinzip sind wir nun schon mit unserer ersten Applikation fertig. Für den Treasure Hunt müssen wir aber noch die Logbuch-App einbinden. Wie das funktioniert, wird in [dieser Anleitung](#) erklärt.

Beispiellösung

Falls Ihr am Bootcamp nicht fertig wurdet mit der Umsetzung der App, könnt ihr einen Blick in unsere Beispiellösung werfen: [coming soon].