

Name: -

Jahrgang: 5YHELI

Gruppe: -

Name der Mitarbeiter: Buchberger Florian, Weiland Mathias

HN5 - TINF
Abteilung Elektronik

an der Höheren technischen Bundeslehranstalt 1
Innsbruck, Anichstraße 16 - 28

Betreuer:
Weiss

Combination Lock

Ausgeführt
am 24. 03. 14

Abgegeben am
01. 04. 14

Inhaltsverzeichnis

1	Aufgabenstellung	2
2	Vorbereitung	2
2.1	PhoneGap	2
2.2	HTML5	2
2.3	CSS 3	3
2.4	Javascript	3
2.5	Sensoren	3
3	Durchführung	3
3.1	Theoretische Überlegungen	3
3.2	Aufbau der App	4
3.3	Implementierung	4
4	Diskussion	5

1 Aufgabenstellung

Es sollte eine App entworfen werden, die mindestens einen Sensor des Mobil-Telefons benutzt. Als Technologie sollte PhoneGap eingesetzt werden.

Die Gruppe entschied sich dafür, eine Kombinations-Schloss-App zu erstellen. Die Kombination sollte durch Drehung des Telefons einstellbar sein.

2 Vorbereitung

2.1 PhoneGap

Adobe PhoneGap ist eine Software zum Erstellen von Apps für Android, Apple iOS und Microsoft Windows Phone. Die eigentliche App wird in HTML5, CSS3 sowie Javascript geschrieben. Der Zugriff auf die Sensoren erfolgt mit Hilfe von kleinen Java-Applikationen, welche von PhoneGap bereitgestellt werden

Nachteile:

- Die Performance ist suboptimal, da der Code des Programmes nicht compiliert, sondern interpretiert wird.
- Quellcode steht mehr oder weniger im Klartext in der App (bei diesem Mini-Projekt kein Problem, da es sowieso open-source ist).
- Zugriff auf die Sensoren, und sonstige Hardware ist sehr eingeschränkt.

Vorteile:

- Nur eine App für die 3 gängigsten Plattformen.
- Dementsprechend müssen nur die Eigenschaften/Einschränkungen einer Programmiersprache beachtet werden.

2.2 HTML5

HTML steht für „Hyper Text Mark-up Language“. Es handelt sich um eine XML-basierte Auszeichnungssprache, welche für den Einsatz auf Websites gedacht ist.

HTML steht in engem Zusammenhang mit HTTP (dem „Hyper Text Transfer Protocoll“), welches im Gegensatz zu früheren Netzwerk-Informationen-Protokollen (wie zum Beispiel „Gopher“) den Vorteil, dass es auch Bilder, Videos, Anwendungen und so weiter übertragen kann.

HTML5 ist, wie der Name schon sagt die fünfte Version von HTML. Die für dieses Projekt wichtigste Änderung ist die Einführung des canvas-Elements.

2.3 CSS 3

CSS steht für „Cascading Style Sheets“ und wird zur zentralen Definition von Stilen für HTML-Dokumente verwendet.

2.4 Javascript

JavaScript ist eine clientseitige Script-Sprache für Websites. Der Name kommt daher, dass die Standard-Objekte in JavaScript den gleichen Namenskonventionen wie die von Java unterliegen. Ansonsten haben diese zwei Sprachen sehr wenig gemeinsam.

JavaScript ist objektorientiert. Klassen sind ebenfalls Objekte, mit denen Prototypen von ihren Instanzen generiert werden. Prinzipiell sind alle Objekte, Eigenschaften und Methoden Variablen. Es gibt daher keine prinzipielle Unterscheidung zwischen Eigenschaften und Methoden. Die Variablentypisierung ist dynamisch. Es gibt keine Unterscheidung zwischen öffentlichen und privaten Eigenschaften und Methoden.

In Javascript kann auch ohne objektorientierung programmiert werden. Für dieses Projekt wird nur diese Art der Programmierung verwendet.

2.5 Sensoren

In diesem Projekt wurden zwei verschiedene Sensoren verwendet: der Kompass und der Beschleunigungssensor.

Der Bewegungssensor funktioniert mittels eines beweglichen Ankers und mindestens 4 feststehenden Kondensatorelektroden. Wenn sich nun der Anker bewegt, ändert sich der Abstand zu den Elektroden. Dies führt zu Änderungen der Kapazitäten, welche gemessen werden können.

Der Kompass misst das ihm umgebende elektrische Feld und errechnet sich daraus die Richtung, in welcher Norden liegt. Durch andere elektrische Geräte oder Magneten in der Nähe kann das Erdmagnetfeld jedoch überlagert werden, wodurch der Kompass nicht mehr nach Norden zeigt.

3 Durchführung

3.1 Theoretische Überlegungen

Es ist durch ein Plugin in PhoneGap auf die Beschleunigungssensoren des mobilen Gerätes zuzugreifen.

Dieses gibt die Beschleunigung in x, y und z-Richtung zurück. Für weitere Überlegungen werden nur x und y-Richtung verwendet.

Wenn das Handy mit dem Display zum Benutzer und der Kopf des Gerätes nach oben gewandt

ist, so zeigt die positive y-Beschleunigung nach unten (normalerweise Erdbeschleunigung: $g \simeq 9.8 \frac{m}{s}$) Für die Umrechnung von den Beschleunigungen in den Neigungswinkel wird Gleichung 1 verwendet.

$$\phi = -\arctan\left(\frac{y}{-x}\right) \cdot \frac{360^\circ}{2 \cdot \pi} - 90^\circ \quad (1)$$

Diese Gleichung ist allerdings nur bei $x < 0$ gültig. Für $x > 0$ wird daher die Gleichung 2 verwendet \Rightarrow Es ist eine Fallunterscheidung notwendig.

$$\phi = 360 - \arctan\left(\frac{y}{-x}\right) \cdot \frac{360^\circ}{2 \cdot \pi} - 90^\circ \quad (2)$$

Zusätzlich muss der Fall $x = 0$ verhindert werden, da sonst eine Division durch 0 vorliegt. Dies wird durch Addieren eines vernachlässigbar kleinem Wert an x , im Falle dessen, dass $x = 0$, verhindert.

3.2 Aufbau der App

Die App zeigt ein rundes Codezahlen-Schloss, dessen Ziffernblatt durch Rotation des Gerätes ebenfalls rotiert. Darunter befindet sich ein Button mit der Aufschrift „I’m feeling lucky“. Durch Klick auf denselben, werden die, seit dem letzten Betätigen des Buttons, aufgenommenen Positionen der Richtungswechsel ausgegeben und mit dem codemäßig eingestellten Code verglichen.

3.3 Implementierung

Mittels der folgenden Codezeile wird darauf gewartet, dass das Gerät bereit ist und die Sensoren verwendet werden können.

```
document.addEventListener("deviceready", startup, false);
```

Wenn dieser Befehl nicht vorhanden ist, gibt die App einen Fehler zurück, weil der angesprochene Sensor aus der Sicht der App nicht vorhanden ist.

Daraufhin wird der folgende Befehl ausgeführt.

```
watchID = navigator.accelerometer.watchAcceleration( onSuccess ,  
onError , options );
```

Dieser sorgt dafür, dass im eingestellten Intervall die Beschleunigungen gemessen werden. In diesem Fall wurde eingestellt, dass die Sensoren alle 100ms ($\hat{=}$ 10 Hz) abgefragt werden.

Wenn diese Abfrage funktioniert, wird die Funktion `onSuccess` verwendet, bei einem Fehlschlag `onError`.

Es wird der Neigungswinkel des Gerätes bestimmt und der Funktion `handleAngle` mitgegeben. Diese rechnet mit Hilfe der Funktion `angleToNum` vom Winkel auf die entsprechenden Zahl der Zahlenscheibe um.

Nun folgt eine Prüfung auf bestimmte Sonderfälle, welche das Ergebnis beeinflussen:

- Ist der vorige Wert minus den aktuellen Wert größer als zwei Drittel der Wählscheibe, so kann davon ausgegangen werden, dass über 0 gedreht wurde.
Resultat: Die Richtung wird auf „links“ gesetzt.
- Ist der aktuelle Wert minus den vorigen Wert größer als zwei Drittel der Wählscheibe, so kann davon ausgegangen werden, dass über 0 gedreht wurde.
Resultat: Die Richtung wird auf „rechts“ gesetzt.
- Ist der aktuelle Wert kleiner als der vorige, so wird die Richtung auf „rechts“ gesetzt.
- Ist der vorige Wert kleiner als der vorige, so wird die Richtung auf „links“ gesetzt.
- Sonst wird die Richtung beibehalten.

Sollte die Richtung nun ungleich der alten Richtung sein (und ungleich der Nullrichtung), so wird der aktuelle Wert zum Array der Eingangswerte hinzugefügt.

Sonst wird, im Falle dessen, dass nach rechts (Richtung ist „rechts“) gedreht wird, die Anzahl der Nullübergänge geprüft. Ist diese größer als 2, so wird das Array der Eingangswerte gelöscht.

Nun wird noch die alte Richtung durch die neue Richtung, sowie der alte Wert durch den neuen Wert, ersetzt.

Zum Schluss wird noch das Ziffernblatt aktualisieren, dies ist jedoch trivial.

Die Funktion, die beim Click auf den Button ausgelöst wird, macht folgendes:

- Ist die Länge des einprogrammierten Codes ungleich der Länge des eingegebenen Codes, so kann davon ausgegangen werden, dass der Code falsch ist.
- Es wird Schritt für Schritt durch das Eingangsarray gelaufen, und bei jedem Eintrag geprüft, ob der absolute Abstand des richtigen Wertes zum eingegebenen größer ist, als eine gewisse Tolleranzgrenze (standardmäßig 2). Ist das der Fall, so wird der Code als falsch angesehen.

Ist das Ergebnis dieser Prüfung, dass der Code falsch ist, so wird die Reihe der eingegebenen Zahlen ausgegeben, und das Eingangsarray zurückgesetzt.

Ist der Code richtig, so wird ein Youtube-Video geöffnet (Link).

Für Details, siehe Source-Code.

4 Diskussion

Ursprünglich wurde versucht, das Programm über den Kompass anzusteuern, da dieser direkt einen Winkel zurückgibt. Da sich allerdings herausstellte, dass der eingebaute Kompass zu ungenau ist, entschied sich das Team für das Auslesen des Beschleunigungssensors.

Das Programm funktioniert prinzipiell, allerdings sind die durch den Sensor verursachten Ungenauigkeiten so stark, dass sie in der `handleAngle`-Funktion einen falsch-interpretierten Richtungswechsel triggern, was das Ergebnis unbrauchbar verfälscht. Um dieses Verhalten zu kompensieren, wurde eine Hysterese eingebaut. Diese funktionierte allerdings nicht wie gewünscht, also wurde diese auf 0 gestellt.