1 Anforderungen

2 Vorbereitung

2.1 Versionsverwaltung

2.2 Android-Tutorium

3 Alternativen

3.1 Alternativen ohne Infrarot

von Anja Hafner

Für den Fall, dass die Ansteuerung des AirSwimmers mit Infrarot nicht realisiert werden kann, wird nach alternativen Geräten gesucht.

Diese Geräte sollten mittels Bluetooth ansteuerbar sein, nicht mehr als 100 € kosten und fliegen können.

Warum Bluetooth?

Die Steuerung per Bluetooth ist gewünscht, da generell alle aktuellen Tablets und Smartphones Bluetooth besitzen und keine zusätzliche Hardware (wie bei der Steuerung mit Infrarot) nötig ist. Außerdem ist die Umsetztung dieser Kommunikation mit dem Fisch deutlich einfacher als mit Infrarot, da die Kommunikation mit Bluetooth besser dokumentiert ist.

Warum soll es fliegen?

Da es sich um ein Avionik-Projekt handelt, sollte der Gedanke, etwas flugfähiges zu steuern, nicht verloren gehen.

Suche nach Alternativen

Zu Beginn wird nach einem AirSwimmer gesucht, der mit Bluetooth gesteuert werden kann. Jedoch gibt es keinen alternativen AirSwimmer, der mit Bluetooth gesteuert wird, lediglich ferngesteuerte AirSwimmer werden angeboten.

Aus diesem Grund wird die Suche auf alle flugfähigen, mit Bluetooth und Android gesteuerten Geräte, erweitert.

Auch hier ist die Auswahl nicht groß, lediglich eine flugfähige Alternative der Firma BeeWi wird angeboten. Dieser Hubschrauber ist per Bluetooth 3.0 ansteuerbar und bietet bis zu 8 Minuten Flugzeit. Die Ladezeit hingegen beläuft sich auf ca. 40 Minuten und erfolgt per USB. Ein weiterer Nachteil ist die schwere Steuerbarkeit des Hubschraubers. Die Kosten des BeeWi BB/301-A0 Bluetooth Controlled Hubschraubers in schwarz belaufen sich auf 50,37 €.

Fahrfähige Alternativen

Da nur eine flugfähige Alternative mit Bluetooth gefunden wurde, wird ebenfalls nach Bluetooth-gesteuerten Fahrzeugen gesucht, auch wenn dies nicht ganz den Anforderungen entspricht. Der Vorteil von Fahrzeugen ist die einfachere Steuerbarkeit.

Das erste Modell stammt ebenfalls von der Firma BeeWi und ist ein Mini Cooper. Dieses Modellauto wird mit Bluetooth 2.0 gesteuert und hat eine Reichweite von bis zu 10 Metern. Für den Betrieb sind 3 AA Baterien erforderlich und die Kosten betragen 30,54 € bei Amazon.

Eine weitere Alternative sind die sogenannten Tankbots. Die kleinen Fahrzeuge werden per USB aufgeladen und besitzen 3 bereits vorhandene Fahrmodi: sie können Hindernisse selbständig ausweichen, in freien Bewegungen "tanzen" oder per Smartphone gesteuert werden. Die Kosten belaufen sich auf 24,99 € bei Amazon.

Für den Fall, dass die Ansteuerung per Infrarot nicht realisiert werden kann, wurde der Hubschrauber als mögliche Alternative ausgewählt.

3.2 Simulation/Emulation

4 Projektleitung

5 Senden

5.1 Irdroid-App

5.2 Senden

6 Empfangen

6.1 Signalanalyse

6.2 Lirc-File

7 Gehäuse

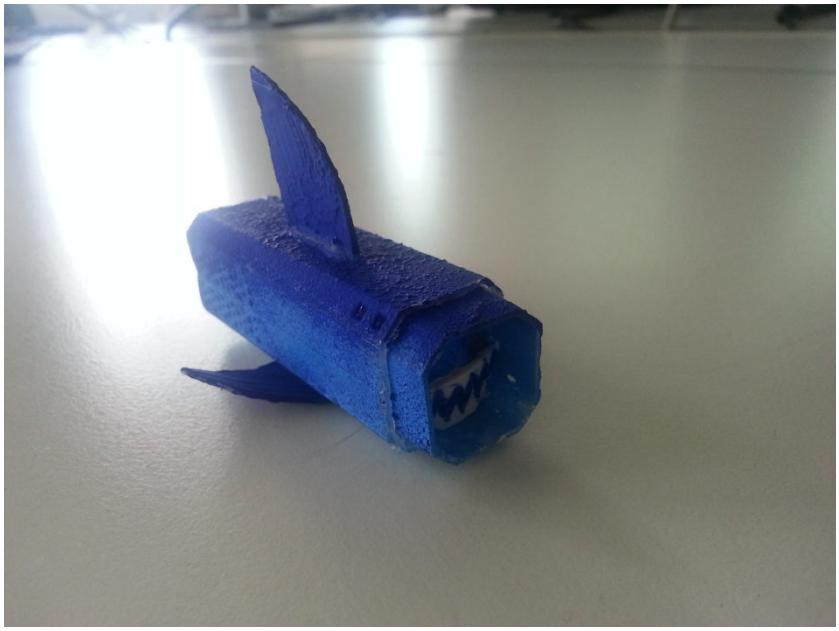
7.1 Recherche Gehäusemöglichkeiten

7.2 Gehäuse bauen

7.3 Gehäuse bemalen und verschönern

Von Belgüzar Kocak

Nachdem ein passendes Gehäuse gefunden wurde und auch der Anfang zur Gestaltung von Andreas Gerken gegeben wurde, musste dieses noch farblich gestaltet werden. Als erstes wurden die Plastikflossen gefeilt anschließend mit Tesa Sekundenkleber angebracht. Die farbliche Gestaltung wurde mit Acrylfarbe und Naturschwamm erarbeitet.



8 Oberfläche

8.1 Konzept

8.2 Startseite

Von Caroline Pilot

8.2.1 Zielsetzung

Jede App benötigt eine Startseite, die es dem Benutzer ermöglicht, eine Auswahl zu treffen, zwischen verschiedenen Seiten und Funktionen zu wechseln und somit den Benutzer empfängt und leitet.

Da die Startseite einen Knotenpunkt für alle Teilanwendungen (Tasten, Kippen, Wischen) darstellt, müssen diese hiermit verknüpft werden.

8.2.2 Vorüberlegung

Aussehen

Da die Startseite in erster Linie die Navigation zu den verschiedenen Benutzermodi übernehmen soll, werden drei Buttons gebraucht.

Außerdem soll die erste Seite der App natürlich optisch ansprechend wirken, deshalb wäre eine Eröffnungsanimation oder eine einladende Graphik des AirSwimmers von Vorteil.

Programmcode

Mit Hilfe der Java-Klasse „Activity“ können die Buttons mit Aktionen belegt werden, die zu den einzelnen Teilfunktionen führen.

Der Programmcode soll in Englischer Sprache erstellt werden.

Außerdem soll für ein Tablet der Größe 10,1 Zoll mit einer Auflösung von 1280 x 800 Pixeln programmiert werden.

8.2.3 Vorgehensweise

Erstellen der Startseite

a) Konfiguration

Um die Einbindung der Teilfunktionen einfacher zu gestalten, wird ein neues Android Appication Project aufgesetzt („AirSwimmer“) und mit folgenden Einstellungen konfiguriert:

- Minimum Required SDK → API 7: Android 2.1 (Eclair)
- Target SDK → API 7: Android 2.1 (Eclair)
- Compile with → API 7: Android 2.1 (Eclair)

Um Probleme bei der Abwärtskompatibilität zu vermeiden, werden niedrige SDK-Versionen verwendet, damit die App auch auf Endgeräten mit älteren Android-Versionen lauffähig ist.

Die Launcher-Konfiguration wird über ein „Image“ der Form „Circle“ mit der erstellten Graphik gespeist.

Der „Activity-Name“ ist passend dazu „FrontPage“ sowie der zugehörige „Layout Name“ „front_page“ lautet.

b) Layout Erstellung

Im Unterordner „Layout“ kann nun per Drag&Drop das Erscheinungsbild der Startseite, wie in der Vorüberlegung angesprochen, erstellt werden. Diese setzt sich aus folgenden Komponenten zusammen:

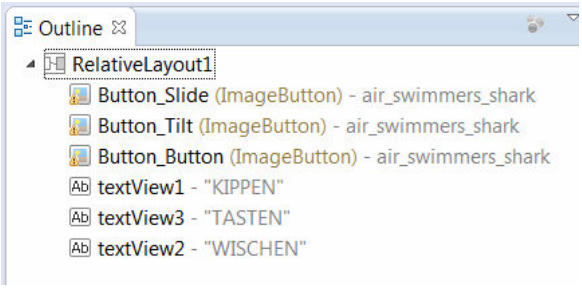


Abbildung Fehler! Kein Text mit angegebener Formatvorlage im Dokument.**1** Komponenten des Layouts der Startseite

Die Buttons werden je nach Funktion mit einem String beschriftet.

Nach Erstellen der Buttons befindet sich für jeden Button ein Codeblock in der xml-Datei. Dieser wird jeweils um folgende Codezeile erweitert, damit der Benutzer diesen Button tatsächlich anklicken kann.

```
android:onClick="onButtonClick"
```

Außerdem wird ein passender Hintergrund gewählt, der das „AirSwimmer-Thema“ aufgreift
und die Buttons werden in ihrer Größe und Ausrichtung auf die Maßen eines Tablets ausgerichtet.

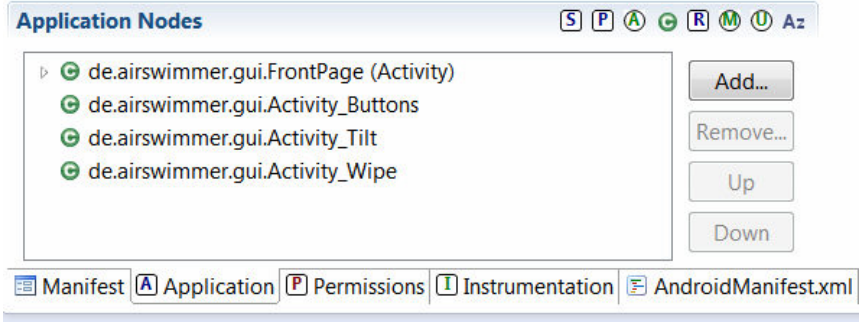
Einbinden der Teilfunktionen

a) Layout und Source-Code einbinden

Durch Erzeugen neuer xml-Dateien („activity_layout_buttons“, „activity_layout_wipe“ und „activity_layout_tilt“) im Unterordner „Layout“ und Einfügen des bereits vorhandenen Codes für die Modi, sind die Layouts der Unterfunktionen nun Bestandteil des Hauptprogramms.

Dasselbe Vorgehen findet auch für den Source-Code im Unterordner „src“ statt, mit neuen Java-Dateien („Activity_Buttons“, „Activity_Wipe“ und „Activity_Tilt“).

Diese Funktionen müssen nun im AndroidManifest unter „Application“ als Knoten hinzugefügt werden:



b) Funktion programmieren

Da durch den Buttonklick zu der jeweiligen Funktion gewechselt werden soll, wird der Source-Code der Startseite um switch-case Anweisungen erweitert:

```
// What will happen when you click on a button
public void onButtonClick(View view){
	switch(view.getId()){
		case R.id.Button_Buttons:
			startActivity(new Intent(this, Activity_Buttons.class));
			break;
		case R.id.Button_Tilt:
			startActivity(new Intent(this, Activity_Tilt.class));
			break;
		case R.id.Button_Wipe:
			startActivity(new Intent(this, Activity_Wipe.class));
		default:;
}
```

Insgesamt muss beachtet werden, dass alle Namenskonflikte aufgelöst werden, alle zusätzlich genutzten Pakete ebenfalls implementiert werden und alle eingefügten Graphiken in das neue Projekt eingepflegt werden.

8.2.4 Ergebnis

Der Benutzer gelangt mit Hilfe dieser App nun über die Startseite hin zu den Teilfunktionen (Tasten, Wischen, Kippen) und kann somit den AirSwimmer steuern.

8.3 Steuerung mittels Buttons

8.4 Steuerung durch Wischen

8.5 Steuerung durch Kippen

8.6 Menü

8.6.1 Steuerungsoberflächen

von Sabine Kressierer

Zielsetzung

In den verschiedenen Oberflächen zum Steuern des AirSwimmers soll ein Menü eingefügt werden, dass das Umschalten zwischen den einzelnen Steuerungsmodi sowie eine Auswahl des Hintergrund ermöglicht.

Vorgehensweise

Ich Ordner „res“ wurde ein Ordner „menu“ angelegt, der die XML-Dateien zum darstellen der Menüs enthält. Da die Menüpunkte der verschiedenen Oberflächen zum Steuern des AirSwimmers gleich sind, wurde für sie nur eine XML-Datei „control.xml“ erstellt. Diese enthält die Menüpunkte mit allen Unterpunkten wie hier für den Menüpunkt „Hintergrund ändern“:

```
        <item android:title="@string/change_background"
android:id="@+id/change_background">
 <!-- Submenu with different background pictures -->
            <menu>
                <group android:id="@+id/submenu_changeBackground" >
                <item
                    android:id="@+id/water"
                    android:title="@string/water_picture"/>

                <item
                    android:id="@+id/sky"
                    android:title="@string/sky_picture"/>
                <item android:id="@+id/th_picture"
android:title="@string/th_picture"/>
                </group>
            </menu>
        </item>
```

In einer Basisklasse „BaseActivity.java“, von der alle Activities zum Steuern des Airswimmers ableiten, wird die Logik für das Menü implementiert. Das Erzeugen des richtigen Menüs erfolgt durch überschreiben der onCreateOptionsMenu-Methode, in der den Activities mit Hilfe des Befehls `getMenuInflater().inflate(R.menu.control, menu)` das Menü der Datei „control.xml“ zugewiesen wird Die Reaktion auf die Auswahl der Menüpunkte erfolgt in der Funktion `onOptionsItemSelected`, die ebenfalls in der Basisklasse überschrieben wird. Auf diese Weise werden die nachfolgenden Menüpunkte realisiert.

a) Hintergrund ändern

von Sabine Kressierer

Zweck

Dem Benutzer soll die Möglichkeit zur Verfügung stehen zwischen verschiedenen Hintergründen für die Steuerungsoberflächen zu wählen.

Umsetzung

Der Menüpunkt „change_background“ enthält ein Untermenü mit den verschiedenen Hintergrundbildern. Wird einer dieser Punkte ausgewählt wird mittels

```
background.setBackgroundDrawable(source.getDrawable(R.drawable.ic_sky));
```

das entsprechende Bild (in diesem Fall ein Himmel) als Hintergrund verwendet. „background“ ist in diesem Fall das Layout der Activity. Zur einheitlichen Verwendung hat das Layout bei allen Activities die Id „layout“.

Besonderheit der Oberfläche Kippen: Da beim Kippen der Hintergrund anstatt einem einfachen Drawable eine BitMap ist wird hier das ändern des Hintergrunds überschrieben. Hier wird der Hintergrund geändert, indem das gewünschte Bild in einer Variablen „background“ des Threads zum Zeichnen der Oberfläche gespeichert wird. Das dort gespeicherte Bild wird in der run()-Methode des Threads gezeichnet.

b) Modus auswählen

von Sabine Kressierer

Zweck

Um die Bedienung zu vereinfachen soll man nicht nur über die Startseite sondern auch über das Menü zwischen den einzelnen Steuerungsmodi (Kippen, Wischen, Tasten) wechseln können

Umsetzung

Der Menüpunkt „change_mode“ enthält ein Untermenü mit den Namen der einzelnen Steuerungsarten. Wird ein Punkt ausgewählt, wird mittels `startActivity(new Intent(this, Activity_Buttons.class));` die entsprechende Activity gestartet. Um ein erneutes Auswähler der bereits laufenden Activity zu verhindern, wird bei Erzeugung des Menüs, der entsprechende Menüpukt ermittelt und unsichtbar gemacht.

Besonderheit der Oberfläche Kippen: Da in dieser Oberfläche ein Thread verwendet wird, muss dieser bei Verlassen der Oberfläche gestoppt werden, da sonst auf einigen Geräten eine Fehlermeldung auftritt. Dies geschieht durch überschreiben der Methode `onOptionsItemSelected` in der entsprechenden Activities. Damit bei Rückkehr zur Oberfläche durch Betätigen des Zurück-Knopfes die im Thread realisierte Animation der Oberfläche wieder stattfindet, wird in der Methode `onResume()` dieser Thread wieder gestartet.

c) Steuerungsart ändern

von Caroline Pilot

Zielsetzung

Der Benutzer soll die Möglichkeit haben das Versenden von Signalen an den AirSwimmer über verschiedene Steuerungsarten zu realisieren.

Zum einen soll genau ein Signal (ein einziger kompletter Bewegungsbefehl) je Tastendruck ausgeführt werden (→ Einfach), zum anderen soll eine permanente Bewegung mit nur einem Klick ebenfalls möglich sein(→Permanent).

Diese Steuerungsart soll sowohl im Tastenmodus als auch im Wisch- und Kippmodus möglich sein.
Im Folgenden wird das Einfügen des Menüpunkts für die Tastensteuerung beschrieben, da die anderen Steuerungsmodi analog ablaufen.

Vorüberlegung

Da es sich zwar bei der Optik jeweils um die gleichen Klassen handelt (Einfache Tastensteuerung und Permanente Tastensteuerung), jedoch die Logik dahinter anders ist, muss die vorhandene Klasse geklont werden, damit ein Menüpunkt später dorthin verweisen kann.

Dieser Menüpunkt („Steuerungsart ändern“) muss als ein weiterer in die Liste eingetragen werden, wobei ein Untermenü zum Wechseln zwischen „Einfach“ und „Permanent“ ebenfalls implementiert werden muss.

Damit der Benutzer auch weiß, in welchem Modus er sich befindet, muss der Android Action Bar Text (Wird am oberen Bildschirmrand während jeder Aktivität angezeigt) angepasst werden.

Vorgehensweise

Erstellen des Menüpunkts „Steuerungsart ändern“

In der bereits vorhandenen *control.xml* Datei werden im Layout folgende Punkte hinzugefügt:

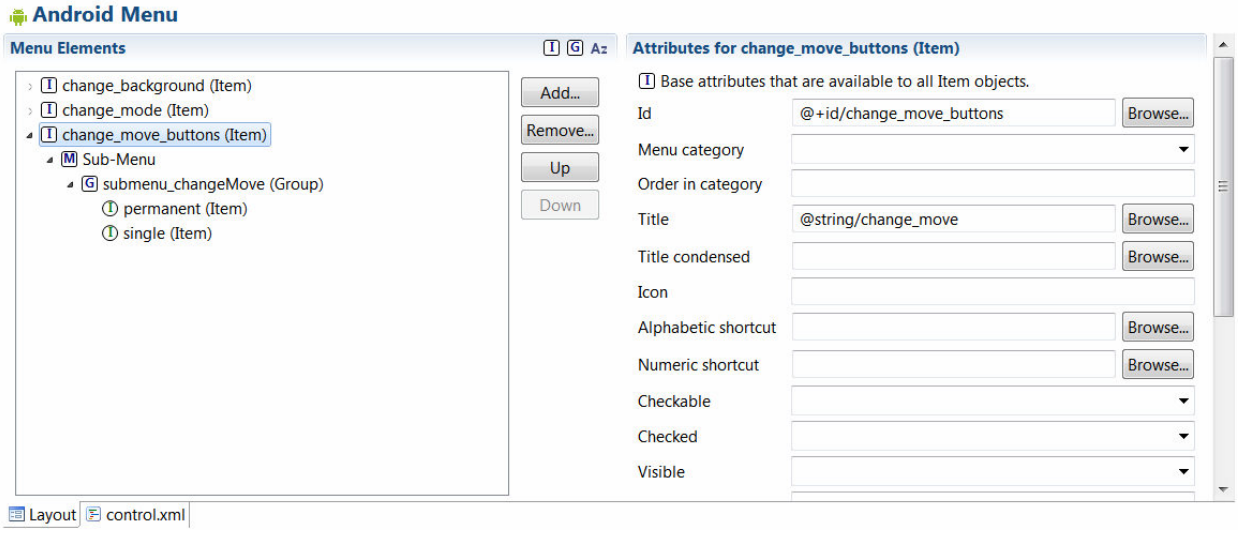


Abbildung 2
Menüpunkt für Steuerungsmodus hinzugefügt

Passende „id‘s“ und „strings“ werden eingetragen. So findet sich unter *@string/change_move* der Text „Steuerungsart ändern“.
Hiermit hat man nun im Menü diesen Punkt, sollte dieser durch Klicken aufgerufen werden, so öffnet sich ein Untermenü mit den Unterpunkten „Einfach“ und „Permanent“.

Um auch die Logik dahinter zu realisieren wird in der BaseActivity Klasse strukturiert abgefragt, um welche Klasse es sich aktuell handelt:

```
if (currentActivityName.equals(getResources().getString(
    R.string.title_activity_activity__buttons))){…}
```

und ob der „Steuerungsart ändern“ Menüpunkt aufgerufen wurde:

```
if (item.getGroupId() == R.id.submenu_changeMove){…}
```

Sollte dies der Fall sein, so wird die Nutzereingabe über `switch (item.getItemId())` eingelesen und zur jeweils zugehörigen Klasse gesprungen:

```
case R.id.permanent:
     startActivity(new Intent(this,Activity_Buttons_Permanent.class));
      return true;
case R.id.single:
      startActivity(new Intent(this, Activity_Buttons.class));
      return true;
default:
      return false;
```

Ändern des Android Action Bar Texts

Der Android Action Bar Text kann im *AndroidManifest.xml* verändert werden.

Da hier ohnehin schon zu jeder Activity ein Codeblock mit einem Label (default: Namen der Klasse) generiert wurde kann dieser mit gebräuchlicheren und verständlicheren Namen ersetzt werden.

```
<activity
            android:name="de.airswimmer.gui.Activity_Buttons"
            android:label="Einfache Tastensteuerung"
            android:screenOrientation="landscape" >
</activity>
```

Hier wird unabhängig von definierten Strings gearbeitet und macht ein Modifizieren und Erweitern einfach.

Ergebnis

Der Benutzer gelangt nun mit Hilfe eines „Steuerungsart ändern“ Menüpunkts zu der Auswahl, ob der aktuelle Modus in einer Permanenten oder Einfachen Steuerungsart bedient werden soll.

Optisch unterscheiden sie sich kaum, jedoch ist die Logik dahinter jeweils eine andere.

Für eine bessere Orientierung sieht der Benutzer am oberen Bildschirmrand nun in welchem Modus und welcher Steuerungsart er sich befindet.

8.6.2 Startseite

Abfrage Aux-Anschluss

Von Sabine Kressierer

Zielsetzung

Die Ausrichtung der Oberfläche soll so erfolgen, dass sich der Aux-Anschluss immer an der oberen Seite des Tablets befindet. Somit kann der IrDroid Sender direkt auf den AirSwimmer gerichtet werden.

Umsetzung

Bei erstmaligem Öffnen der App wird zuerst ein Dialog geöffnet, in dem angegeben werden muss wo sich der Aux-Anschluss befindet. Dies geschieht mit Hilfe eines AlertDialog, der in der onCreate()-Methode der FrontPage-Activity erzeugt und angezeigt wird. Die Reaktion auf die Auswahl eines Punktes erfolgt mittels onClick-Listener, in dessen onClick-Methode der Index des gewählten Punktes bekannt ist. Entsprechend dieser Auswahl wird die ActivityInfo-Konstante für das entsprechende Layout gespeichert.

```
if (item == 0) { //get value for requested screen orientation
      layout = ActivityInfo.SCREEN_ORIENTATION_PORTRAIT;
} else if (item == 1) {
      layout = ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE;
}
```
[1]

Da sich die Position des Aux-Anschlusses nicht verändert, wird dieser Wert dauerhaft gespeichert. Dazu werden shared-preferences verwendet, die es ermöglichen innerhalb der gesamten App auf den Wert zuzugreifen. Mit `Preferences = getSharedPreferences("AirSwimmerPrefs",Context.MODE_WORLD_READABLE);` werden die preferences in der Datei „AirSwimmerPrefs“ ermittelt. In diesen Preferences wird dann mit dem `ShardPreferences.Editor` der int-Wert zum Schlüssel „layout“ auf die oben ermittelte Konstante gesetzt. Am Ende wird in der onCreate-Methode der Startseite mittels `setRequestedOrientation(layout);` die Orientierung entsprechend verändert. Mit der selben Funktion wird in der onCreate-Methode der „BaseActivity“ die Orientierung gesetzt. Um dort den in den Preferences gespeicherten Wert zu erhalten, werden wie oben die Preferences ermittelt und anschließend `preferences.getInt("layout", -1)` aufgerufen.

Bei erneutem Starten der App wird der Dialog nur noch gestartet, wenn in den Preferences kein Wert gespeichert ist.

Menü der Startseite

Von Sabine Kressierer

Zweck

Die Ausrichtung der App soll nachträglich veränderbar sein.

---

[1] 0=Aux-Anschluss an kurzer Seite→Portrait / 1= Aux-Anschluss an langer Seite → Landscape

Umsetzung
In der Datei „start.xml“ wurde ein Menüpunkt „change_layout_orientation“ eingefügt, mit einem Untermenü für die beiden Orientierungsmöglichkeiten. Das Erzeugen des Menüs erfolgt in der FrontPage-Aktivity analog zum control-Menü. Bei Auswahl einer Orientierung im Untermenü, wird dieser Wert wie beim AlertDialog in den Preferences gespeichert und die Ausrichtung der FrontPage angepasst.

9 Graphische Gestaltung der Oberfläche

9.1 Konzept

Von Caroline Pilot

9.1.1 Zielsetzung

Ziel eines Prototyps für das Design der Oberfläche[2] ist es, der Gruppe einen ersten visuellen Eindruck für die App zu geben. So bekommt das gemeinsame Ziel des Teams ein Gesicht, auf das motivierter hingearbeitet werden kann.

Außerdem bieten Modelle eine bessere Diskussionsgrundlage für Wünsche, Ideen und Verbesserungsvorschläge. So kann die Oberflächenentwicklung mit konkreten Designs ausgeführt werden.

9.1.2 Vorüberlegung

Zunächst muss ein Graphikprogramm gefunden werden, das den Anforderungen entspricht.

Hierbei ist folgendes zu beachten:

- Vielerlei Möglichkeiten Bilder zu bearbeiten (Kontraste, Layer, Transparenz, …)
- Erstellen von Graphiken in gängigen Bildformaten
- Einfache Bedienung, damit keine lange Einarbeitungszeit nötig ist
- Kostengünstig, am besten ein Freeware Download Programm

Des Weiteren muss der Kreativität bei folgenden Punkten freien Lauf gelassen werden:

- Wie soll der Hintergrund aussehen?
- Wie soll das Objekt dargestellt werden?
- Welche Besonderheiten benötigen die einzelnen Benutzermodi?

[2] Hiermit sind im Kapitel 5.1 „Konzeptentwicklung für das Oberflächendesign“ die unterschiedlichen Benutzermodi zum Steuern des AirSwimmers gemeint, nicht die Startseite der App

9.1.3 Vorgehensweise

Unter http://www.chip.de/downloads/PhotoFiltre_13012070.html ist das Freeware Graphikprogramm als Download verfügbar. Es entspricht unseren Anforderungen und enthält zudem sehr gute Nutzerrezensionen.

Aus dem Brainstorming zum Thema Oberflächendesign ergibt sich folgendes:

- Hintergrund:
  - Himmel
  - Unterwasserwelt
  - Stadt/Skyline
- Objekt
  - Bild des AirSwimmers
  - Bild eines Fisches
  - Bild eines Köders, dem der echte AirSwimmer „hinterher schwimmt“
- Besonderheiten einzelner Modi
  - Tastensteuerung
    - Tasten in gewöhnlicher Buttonform
    - Tasten in Form eines Fisches
    - Kontrastreiche Farben <–> Gedeckte Farben
  - Kippsteuerung
    - Angabe von Himmelsrichtungen (north, south, west, east)
    - Angabe von Bewegungsrichtung (up, down, left, right)
    - Keine Angabe
  - Wischsteuerung
    - Angabe von Himmelsrichtungen (north, south, west, east)
    - Angabe von Bewegungsrichtung (up, down, left, right)
    - Keine Angabe

9.1.4 Ergebnis

Bilder zum Verwirklichen dieser Ideen und zum Erstellen von Designprototypen lassen sich im Internet finden oder mit dem Graphikprogramm erstellen.

In verschiedenen Kombinationen, umgesetzt mit dem Photofiltre7 ergeben sich folgende Entwürfe:

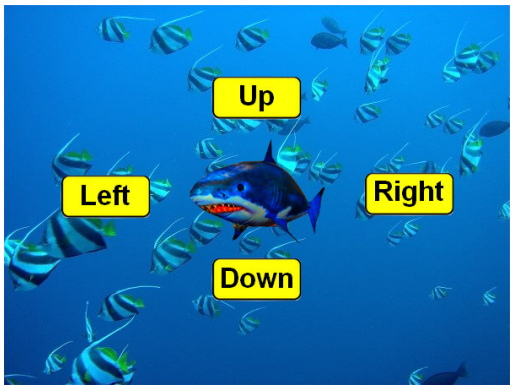


Abbildung 1
Tastensteuerung;
Unterwasserwelt, Kontrastreiche Buttons

Abbildung 2
Tastensteuerung;
Himmel, Gedeckte Buttons

Abbildung 3
Kippsteuerung;
Unterwasserwelt, Angabe von Himmelsrichtungen

Abbildung 4
Kippsteuerung;
Skyline, ohne Angaben

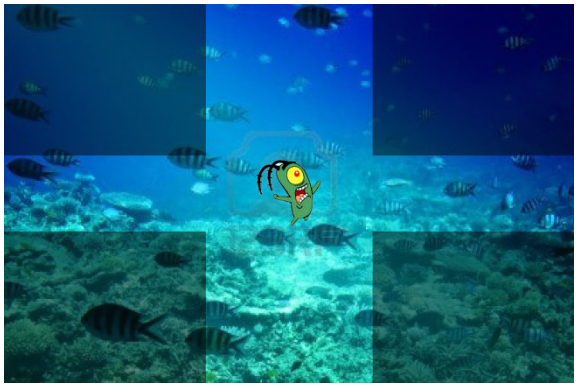


Abbildung 5
Wischsteuerung
Benutzergrenzen, für definierte Bewegungen, Köder als Objekt

Abbildung 6
Wischsteuerung
Benutzergrenzen, für definierte Bewegungen

Die Gruppendiskussion bringt hervor, dass es in der App einen Punkt für einen Wechsel des Hintergrunds zwischen Himmel und Unterwasserwelt geben wird, je nach Wunsch des Benutzers.

Des Weiteren werden nicht die Himmelsrichtungen im Kippmodus angegeben, denn dies könnte zu Verwirrungen führen, stattdessen werden keine Angaben platziert, ebenso im Wischmodus.

Das Objekt in der Mitte soll ein Fisch sein.

9.1.5 Ausblick

Da es sich bei den verwendeten Bildern um Eigentum Dritter handelt, könnte man sich durch dessen Verwendung rechtlich Probleme einhandeln. Aus diesem Grund ist es wünschenswert, eigene Graphiken, vielleicht auch dynamische, zu erstellen oder die Urheber schriftlich um eine Genehmigung zu bitten.

9.2 Launcher-Icon

von Caroline Pilot

9.2.1 Zielsetzung

Um einen einheitlichen Auftritt der App zu ermöglichen, wird ein Launcher-Icon benötigt. Dieser repräsentiert die App auf dem Bildschirm und ist auch bei der Ausführung am Rand zu sehen.

9.2.2 Vorüberlegung

Da bereits das Graphikprogramm PhotoFiltre vorhanden ist, kann dieses verwendet werden, um den Launcher-Icon zu erstellen.

Des Weiteren muss der Kreativität bei folgenden Punkten freien Lauf gelassen werden:

- Welche Form soll der Icon bekommen?

- Welches Objekt soll darauf zu sehen sein?
- Welche Farben könnten passen?

9.2.3 Vorgehensweise

Aus dem Brainstorming ergeben sich folgende Punkte:

- Form
    - Rund, da es einer Wasserblase entspricht, in dem sich der AirSwimmer fortbewegt
- Objekt
    - Da es eine App für den AirSwimmer sein soll, liegt es nahe, dass auch auf dem Icon der Fisch abgebildet ist. Somit wäre es eindeutig auf dem Desktop zu identifizieren
- Farben
    - Die Farben müssen herausstechen, sodass die App nicht auf dem Desktop untergeht und man sie nicht sehen kann (Beispiel schwarz), dennoch müsste die Farbe zum Thema Wasser/Himmel passen

9.2.4 Ergebnis

Das Bild des AirSwimmers lässt sich im Internet finden.

In verschiedenen Kombinationen mit Farben und Hintergründen, umgesetzt mit dem Photofiltre7 ergeben sich folgende Entwürfe:

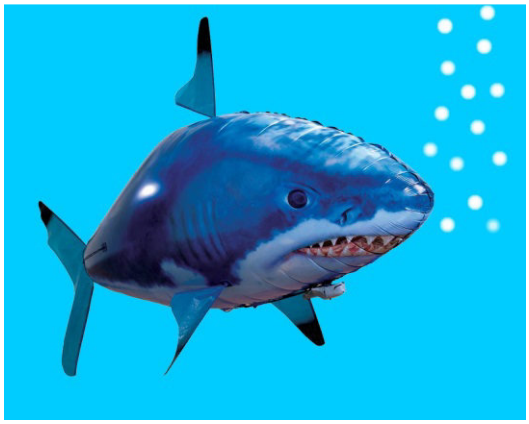


Abbildung 1

Launcher-Icon mit hellem Hintergrund

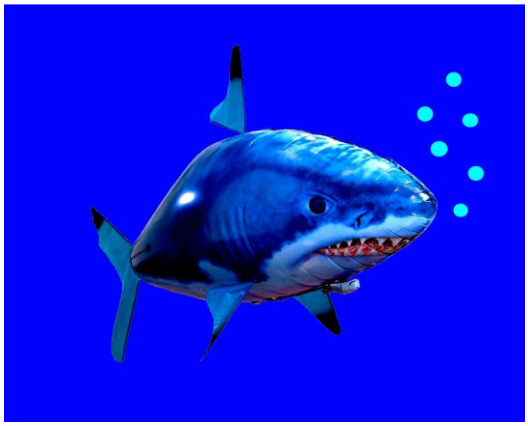


Abbildung 2

Launcher-Icon mit hellem Hintergrund

Nach dem Testen auf dem Tablet wird festgestellt, dass der dunkle Hintergrund bei einem schwarzen Desktop untergeht, daher fiel die

Entscheidung auf einen runden Launcher-Icon mit hellem Hintergrund (Abbildung 2).

9.3 Animierter Hintergrund

9.4 Hintergrundbilder

9.5 Graphiken für Buttons und Hai

Von Belgüzar Kocak

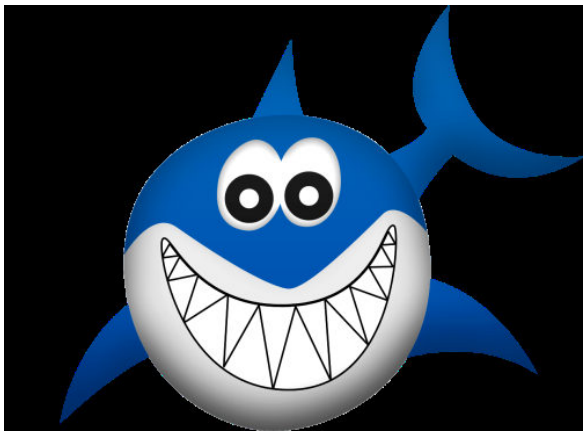
9.5.1 Zielsetzung

Ziel war es für die Oberfläche Kippen und Wischen eine Grafik von einem Hai zu erstellen und auch Buttons für die Buttons Oberfläche zu zeichnen. Genutzt wurden hier folgende Tools: Microsoft Word, Paint.Net und Photoshop.

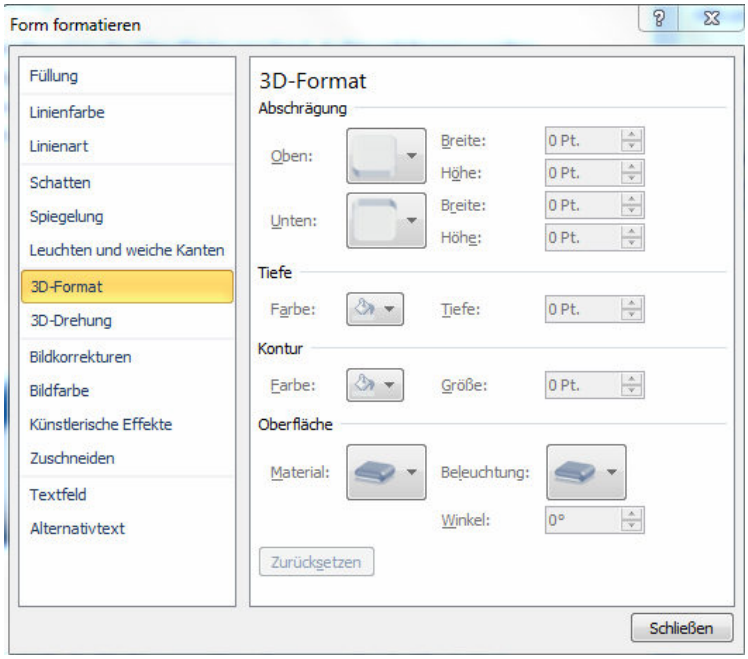
9.5.2 Vorgehen und Erstellen

Verlangt wurde, dass der Hai direkt den Benutzer der Oberfläche anschaut. Aufgrund dessen wurden vorerst Zeichnungen per Hand erstellt genutzt wurden dabei ein einfaches Papier so wie Bleistift. Somit war der erste Schritt der Zeichnung getan. Jetzt war es wichtig die Zeichnung vom Papier auf den Bildschirm als Grafik zu verwirklichen. Dies gelang mit dem Photoshop Programm.

Somit entstand folgende Grafik:



Die Pfeile wurden in Microsoft Word erstellt, die Pfeilform wurde in die passenden Größe skaliert. Nach der Skalierung wurde unter Form formatieren in 3D-Format passende Werte eingegeben sowie Farben geändert. Anschließend in Paint.Net der Hintergrund anhand der Zauberstabfunktion entfernt.



Nach dem Erstellen der Grafiken wurden diese in die Oberfläche eingebaut. Wie zum Beispiel in die Buttonoberfläche.