



Dokumentation iPad Spot The Difference

Objective-C/Cocoa Vorlesung

Bericht von

Andreas Knöpfle (281187)

Mathias Hodler(281159)

Konstanz, 18. Februar 2011

Inhaltsverzeichnis

1	Abkürzungsverzeichniss	3
2	Idee und Spielkonzept	4
3	Spezifikation	5
3.1	Benutzerschnittstellen	5
3.2	Architektur	5
4	Umsetzung	8
5	Lösung	9
	Literatur und Quellenverzeichnis	10

1 Abkürzungsverzeichniss

JPEG Joint Photographic Experts Group, ein im Web weit verbreitetes Grafikformat für verlustbehaftete oder verlustfreie Kompression von digitalen Fotografien.

IOS ein Betriebssystem der Firma Apple für mobile Geräte. Es basiert auf Mac OS X und ist das Standard-Betriebssystem der Apple-Produkte iPhone, iPod touch, iPad und der zweiten Generation des Apple TV

2 Idee und Spielkonzept

Im Rahmen des Wahlpflichtfaches "Objective-C/Cocoa" musste eine Anwendung für das iPhone bzw. iPad erstellt werden. Dabei sollte auch wenn möglich die Besonderheiten von IOS wie z.B. der Lagesensor, GPS und Touch-Funktionen mitverwendet werden.

Wir haben uns Entschlossen eine neue Art Fehlersuchspiel zu entwickeln. Dieses lehnt sich an die bekannten Fehlersuchspiele in Zeitschriften an, bei denen zwei auf den ersten Blick identische Bilder nebeneinander zu sehen sind, wobei eines davon mehrere Veränderungen enthält, welche zu finden sind (siehe Abbildung 1). In "iPad Spot The Difference" liegt jedoch die Besonderheit, dass das Fehlerbild ein Foto aus der Realität ist und um die Fehler zu finden, dass auf dem Display dargestellten Foto auch mit der realen Umgebung verglichen werden muss.



Abbildung 1: Typisches Fehlersuchbild in Zeitschriften

Der Spieler erhält zu Beginn eine Weltkarte angezeigt, auf der alle Punkte markiert sind, für diese ein Fehlerbild existiert. Befindet sich der Spieler in der örtlichen Nähe einer solchen Markierung kann er das Fehlerbild mit der realen Umgebung vergleichen und die Fehler durch einen Klick auf das Bild aufdecken.

Das Spiel wurde speziell für das iPad entwickelt, da eine möglichst großes Display nötig ist, um auch die Fehler zu erkennen.

3 Spezifikation

Im folgenden werden die verschiedenen Benutzeroberflächen gezeigt und die grundlegende Architektur festgelegt.

3.1 Benutzerschnittstellen

3.2 Architektur

Das Klassendiagramm (siehe Abbildung 2) zeigt den Aufbau der Anwendung. Die einzelnen Klassen und deren Funktionen sind im folgenden erklärt.

ImageManager

Die Klasse ImageManager hat die Aufgabe aus einer Datenquelle alle SpotImages zu erzeugen und diese zu verwalten.

SpotImage

Ein SpotImage definiert ein spezielles Fehlerbild mitsamt seinen geographischen Koordinaten, Titel, Beschreibungstext und den eigentlichen Fehlern (Klasse Difference). Sie bietet über die Funktion "doesHitWith" die Möglichkeit anhand der Angabe einer xy-Koordinate festzustellen ob dort ein Fehler im Bild existiert. Diese Funktion wird benötigt wenn der Benutzer auf das Bild, um zu überprüfen ob dort auch wirklich ein Fehler versteckt ist.

Difference

Die Klasse Difference stellt einen einzelnen Fehler in einem SpotImage dar. Dieser wird durch seine Position im Bild (xy-Koordinate) und dessen Größe (Breite und Höhe) definiert.

MapViewController

Der MapViewController ist für den View zuständig, der die Weltkarte und die SpotImages anzeigt.

ImageViewController

Das Anzeigen des im MapViewController ausgewählten SpotImage geschieht im View des ImageViewController. Der ImageViewController besitzt daher eine Referenz auf das gerade aktive SpotImage. Alle aufgedeckten Fehler werden temporär gemerkt um dem Spieler zu zeigen wie viele Fehler noch zu finden sind.

AboutViewController

Für die Anzeige einer Kurzanleitung für das Spiel ist der AboutViewController zuständig.

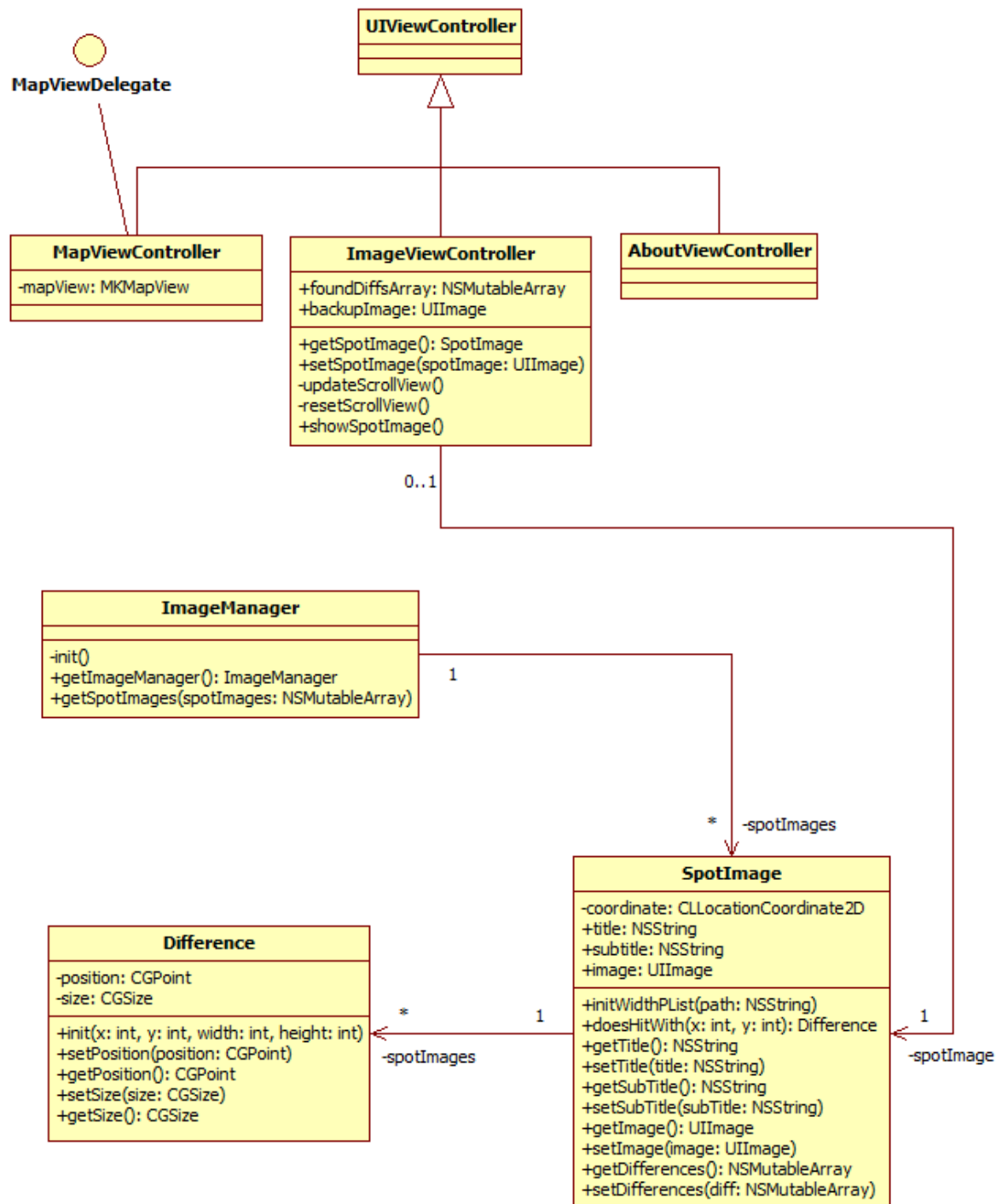


Abbildung 2: UML-Klassendiagramm

4 Umsetzung

5 Lösung

Literatur und Quellenverzeichnis

[1] Titel, Autor

<http://www.example.com>

Stand: 17.03.2010