

Lebenslauf

Zur Person

NAME Johannes Schriewer
ADRESSE Langemarckstr. 150, 28199 Bremen
TEL +49 160 90 39 21 17
SKYPE dunkelstern738
EMAIL hallo@dunkelstern.de
TWITTER @dunkelstern
GITHUB dunkelstern

GEBURTSDATUM 24.01.1984

Arbeitserfahrung

DATUM Juni 2006 - September 2008
POSITION Softwareentwickler
TÄTIGKEIT Programmierung in C/C++ (Video und Datenkomprimierung), UNIX-Administration, Linux Live-CD Wartung (Demo der Applikationen)
FIRMA X-Software

DATUM August 2010 - heute
POSITION Leitender Softwareentwickler
TÄTIGKEIT iOS Programmierung (u.A. CEWE-Fotowelt, Vazz), Server-Administration (Linux, Mail und Webserver), Web-Entwicklung (PHP, Ruby, Python), Entwicklung von Backendlösungen für iOS Apps, Low-Level Videoprocessing für Streaming-Plattform
FIRMA Planetmutlu

Lebenslauf

Ausbildung

DATUM Juni 2003
Fachabitur (Technischer Zweig)
PRAKTIKA Metallverarbeitung und Elektronik

DATUM November 2003 - Oktober 2004
TÄTIGKEIT Ausbildung Fachinformatiker
AUFGABENBEREICH Administration der IT Infrastruktur (Windows/Linux),
Wartung von Servern (Apache, Cyrus/Postfix, MySQL)
FIRMA AMAN Media

DATUM Oktober 2004 - Juni 2006
TÄTIGKEIT Ausbildung Fachinformatiker
AUFGABENBEREICH Plattformübergreifende (Unix, Linux) Programmierung
in C, Wartung der Website
FIRMA X-Software

DATUM Oktober 2008 - Mai 2010
TÄTIGKEIT Studium Kommunikations und Informationselektronik
(ehem. Nachrichtentechnik)
ABSCHLUSS nicht erreicht, Studium abgebrochen

Lebenslauf

Fähigkeiten

SPRACHKENNTNISSE	Deutsch (Muttersprache) Englisch (fließend in Wort und Schrift) Französisch (Anfängerkenntnisse, nur lesen)
PROGRAMMIERSPRACHEN	Objective C, C, C++, C#, Ruby, Python, Bash/Posix Shell PHP, Visual Basic
FRAMEWORKS	POSIX, Cocoa-Touch, Cocoa, .NET, Ruby on Rails, Flask (Python Web-Framework), Arduino (Mikrocontroller)
TECHNOLOGIE-ERFAHRUNG	Komprimierungsalgorithmen H.264 und AAC Codecs MPEG-TS und MP4 Containerformate XML und JSON Schnittstellen Netzwerkkommunikation und Synchronisierung Video- und Audio-Streaming Rechenzentrumsskalierbarkeit Unix Daemon- und Server-Prozesse programmieren Bildbearbeitung und OpenGL-Shader PDF-Erstellung auf iOS ORM (DSTStorageEngine, ActiveRecord) Datenbanken (MySQL, PostgreSQL, MongoDB) Modernes HTML, CSS (+LESS) und Javascript Websockets Build-Systeme (make, xcode, automake, imake) Versionsverwaltung GIT und SVN Continuous Integration Buildserver Jenkins Dokumentationstools AppleDoc, Doxygen Linux Administration (Apache, Varnish, Ruby on Rails, Postfix, Dovecot, Courier, Debian + Archlinux Pakete) Teilweise UNIX Administration (Solaris, HP-UX, AIX, IRIX)

Lebenslauf

SOFTWAREKENNTNISSE (KREATIV)	Adobe Photoshop (Sehr gute Kenntnisse) Adobe Lightroom (Sehr gute Kenntnisse) Adobe Illustrator (Gute Kenntnisse) Adobe AfterEffects (Anfänger) Adobe Audition (Anfänger) Final Cut Pro X (Anfänger) Motion (Anfänger) Blender (Anfänger)
SOFTWAREKENNTNISSE (TECHNISCH)	Autodesk Inventor (Fortgeschritten) EAGLE-CAD (Fortgeschritten) SolidWorks (Anfänger)
HOBBYS (KREATIV)	Schriftgestaltung Holz- und Metallbearbeitung Studio- und Produktfotografie Bildbearbeitung
HOBBYS (TECHNISCH)	3D-Drucker (Firmware und Hardware Hacking), Elektronik (Mikrocontroller, Arduino) CAD für 3D-Druck Programmierung div. Projekte
UNFERTIGE PROJEKTE	2D Geometrie Library für eigene Slicer-Implementation für 3D-Drucker (C, Shared Library OSX, Win und Linux) ORM Layer für iOS (DSTStorageEngine) Die Plastikschmelzer (3D-Drucker Podcast) The Iron Photographer (Fotografie Social Network Idee)
ABGESCHLOSSENE PROJEKTE	iOS Komponenten: - Tiled Map Renderer für Guildwars 2 auf iOS (ohne MapKit oder Google Frameworks, from Scratch) - SVG-Renderer basierend auf libSVGTiny für iOS - UIPickerView-Ersatz für iOS (pre iOS 7 Stil)

Lebenslauf

ABGESCHLOSSENE PROJEKTE (FORTSETZUNG)

JoyBoard (<http://projects.dunkelstern.de/JoyBoard/>)
Archlinux KDEMod (jetzt Chakra)
Blog-Software für Eigene Website (offline, PHP)

Microcontroller Projekte:

- Fernbedienungsempfänger
- Arduino like Experimentierplattform (bevor Arduino existierte)
- NES/SNES Controller auf USB-Adapter
- Div. LED Matrix-Steuerungen
- TIX-Clock
- Intervallometer für Canon EOS 350
- Atmel AVR Programmer

Apps

- CEWE-Fotowelt (Hauptverantwortlicher Programmierer)
- Vazz Videostreaming (Low-level Videoprocessing, Programmierer)
- PocketChris (Programmierer, 1. iOS Projekt)
- Nordsee (eingestellt)
- Jazzahead Festival
- SixDays Sportveranstaltung (eingestellt)
- Halle7 Veranstaltung (eingestellt)
- InstaNyan (offline wegen Copyright Konflikt)

EXPERIMENTALERFAHRUNG

Go (als es nur den x86_64 compiler gab)
Node.js
Laravel (PHP)
Codeigniter (PHP)
Wordpress Themes

Programmiersprachenkenntnisse

C

Mit C begann ich meine Ausbildung als Fachinformatiker (X11 ist C) und wegen der Flexibilität und den Möglichkeiten die Sprache auf jedem System als Bibliothekssprache einsetzen zu können ist C auch immer noch meine Lieblingssprache. Das manuelle Memory Management geht irgendwann in Fleisch und Blut über und erlaubt dann gerade auf Begrenzten Systemen eine sehr gute Kontrolle über die verfügbaren Ressourcen.

C++

Mit C++ hatte ich zum ersten mal zu tun als ich einen Bug in KDE fixen wollte, das Qt-Framework (damals Version 3) hat mich damals so begeistert, dass ich tiefer eingestiegen bin. Dadurch dass ich mit Qt gearbeitet habe bin ich jedoch nie in den "Genuss" von STL und Boost gekommen.

C#

C# habe ich für ein paar Projekte unter Windows benutzt, hauptsächlich Wegwerf-tools um aus Spielen Audio- und Bild-Daten zu extrahieren. Allerdings kam auch ein funktionierendes Projekt dabei heraus: Joyboard (Beschreibung weiter unten).

OBJECTIVE C

Mit Objective C habe ich im Januar 2010 begonnen als ich mein erstes iPhone (3GS) gekauft habe. Ich habe schon lange damit geliebäugelt Apps für ein Smartphone zu schreiben (ein Freund von mir hatte ein Windows Mobile 6.5 Smartphone), doch sagte mir das angestaubt wirkende Windows Mobile nicht zu. Zum gleichen Zeitpunkt ergab es sich, dass ich mein Studium würde abbrechen müssen und so investierte ich meine komplette Zeit in Objective C und bekam so auch den Job bei planetmutlu. Objective C ist nach C wohl die Sprache in der ich mich am ehesten zuhause fühle. Ich kenne die Apple-Frameworks gut und habe viel Erfahrung mit den iOS-Frameworks sammeln können.

RUBY

An Ruby bin ich geraten als wir bei planetmutlu für Nordsee ein neues CMS gesucht haben, Rails hat mich sofort in seinen Bann gezogen und das Tooling was für Ruby existiert ist einfach vielen anderen Deployment- und Dependency-Management-Systemen überlegen gewesen zu der Zeit. Mittlerweile nutze ich Ruby auch um Xcode-Builds zu automatisieren und für diverse Fire-And-Forget Scripte die so Anfallen. Es muss nicht immer Rails sein, manchmal reicht auch Rack oder ein anderes kleines Gem.

PYTHON

Python habe ich mir angeeignet weil ich eine Script-Sprache brauchte um schnell ein Datenformat in ein anderes zu konvertieren. Ruby war zu dem Zeitpunkt noch nicht auf meiner Liste der coolen Sprachen aufgetaucht. Die meisten Dinge in Python die ich geschrieben habe waren Daten-Konverter als Backend für die Apps bei planetmutlu, sowie ein Flask-Basiertes E-Mail Server-Management-System (Passwortreset,, Usermanagement)

Programmiersprachenkenntnisse (Fortsetzung)

BASH/POSIX SHELL

Bash und Posix-Shell Kenntnisse habe ich mir in der Ausbildung bei X-Software angeeignet um viele wiederkehrende Arbeiten zu automatisieren. Das Wissen um die Shell war von Vorteil für das Verständnis von Makefiles und anderen Build-Systemen (PKGBUILD von Archlinux) und ich bin tiefer eingestiegen als ich mit einem Bekannten das KDEmod-Projekt gegründet habe (siehe weiter unten) und sehr viele Archlinux Pakete bauen musste.

VISUAL BASIC

Visual Basic ist die Sprache mit der ich zum ersten mal ernsthaft programmiert habe (QBasic und GFA-Basic waren die Vorgänger). Mit 13 begann ich mit VB3 um dann 1998 auf VB6 umzusteigen. Ich bin dann über die Sprache hinaus gewachsen und mit Hilfe von Leuten aus dem ActiveVB-Forum habe ich angefangen Win32 Systemmessages zu missbrauchen um Assembler Code in VB zu bekommen. Ab dem Zeitpunkt sollte mir eigentlich bewusst geworden sein, dass eine andere Sprache evtl. besser geeignet gewesen wäre, doch ich kannte VB in und auswendig. Als ich damals dann auf Linux umgestiegen bin lernte ich C und C++ (siehe oben)

Softwarekenntnisse

ADOBE PHOTOSHOP UND LIGHTROOM

Mit Bildbearbeitungssoftware habe ich bereits sehr früh experimentiert (Schülerzeitung), aber richtig eingestiegen in Photoshop und Lightroom bin ich als ich begann zu fotografieren (ich hatte dann recht schnell eine kleine Studioausrüstung zusammen mit Blitzen, etc.).

Bei planetmutlu habe ich dann von meinem Chef Interface-Design und die technische Seite von Photoshop kennen gelernt und auch sehr viele iOS-Interfaces selbstständig umgesetzt.

ADOBE ILLUSTRATOR, AFTER EFFECTS UND BLENDER

Für diverse Demos und In-App Grafiken und Animationen eignete ich mir bei planetmutlu alles was ich zum Prototyping einer Animation brauchte an.

APPLE MOTION, FINAL CUT PRO X UND ADOBE AUDITION

Diese Software habe ich nur privat verwendet und unter Anderem auf zwei Foto-Workshops für meine Projektpräsentation verwendet. Videoschnitt ist ein Interessantes Feld, ohne entsprechendes Rohmaterial und eine Projektidee kommt man aber leider nicht weit. Adobe Audition verwende ich momentan um den Plastikschmelzer Podcast zu schneiden.

Softwarekenntnisse (Fortsetzung)

AUTODESK INVENTOR UND SOLIDWORKS

Als ich mir im November 2013 einen 3D-Drucker zulegte weil mich die Technologie faszinierte und ich das Bastelpotential erkannte musste eine Software her mit der ich CAD machen konnte um mir Teile zum Drucken zu bauen. Ich probierte alle möglichen Programme durch (u.A. SketchUp, FreeCAD, PTC/Creo Elements, OpenSCAD) und blieb an Inventor Fusion (mittlerweile eingestellt und ersetzt durch Fusion 360) hängen. Ich testete auch Solidworks (30 Tage Testversion) und erkannte, dass Autodesk Inventor beinahe das selbe Interface hatte aber mehr konnte und wechselte dann zu Inventor. Simple Plastikteile die sich zum Drucken eignen kann ich schnell und sicher erstellen, doch ist der Umfang der Software so groß, dass ich wahrscheinlich nur an der Oberfläche kratze mit meinem Wissen.

EAGLE CAD

Eagle benutze ich seit ca. 2005 für meine Elektronik-Projekte. Die meisten Schaltungen die ich bisher gebaut habe waren nie allzu umfangreich und die kostenlose Version hat eigentlich immer gereicht da ich noch nie eine Platine Professionell Produzieren ließ.

Hobbys

SCHRIFTGESTALTUNG

Seit 2010 versuche ich mich in Schriftgestaltung, mein erstes Projekt war es meine Handschrift zu digitalisieren. Ein weiteres Projekt befasste sich damit die Schrift aus Guildwars 2 nachzubauen. Ich habe Grundkenntnisse über Typografie und Satz aber wenig Erfahrung.

HOLZ- UND METALLBEARBEITUNG

Immer wenn etwas anfällt was man selber bauen kann versuche ich das zu tun. Handwerkliches Arbeiten ist ein schöner Ausgleich zum Job als Softwareentwickler.

STUDIO- UND PRODUKTFOTOGRAFIE, BILDBEARBEITUNG

Mit Fotografie befasst habe ich mich seit dem ich 16 bin, als ich dann nach der Ausbildung genug Geld verdiente um mir dieses teure Hobby zu leisten schaffte ich eine Kamera an und baute mir ein kleines Fotostudio (3 Blitze, Lichtformer, Hintergründe). Ich Fotografiere immer noch sehr gerne und habe auch einige Workshops mitgemacht aber habe momentan leider keine Zeit.

Hobbys (Fortsetzung)

3D-DRUCKER, CAD FÜR 3D-DRUCK

Seit November 2013 besitze ich einen 3D-Drucker (kaufte damals ein Mendel90 Kit) und verbringe viel Zeit damit Dinge des Alltags zu designen und zu Drucken. Natürlich fließt auch viel Zeit in Wartung und Optimierung des Druckers und in die Erweiterung und Verbesserung der vorhandenen Komponenten. Da ich im Bereich Mechanik noch nicht so gut bin bastele ich momentan gern an Firmware herum.

ELEKTRONIK

Es gibt eigentlich immer schon kleinere und größere Elektronikprojekte die ich versuche umzusetzen. Oftmals bleiben diese dann im Entwurfsstadium stecken weil andere Dinge wichtiger sind oder weil sich Fehler einschleichen die dann nachher nur schwer zu korrigieren sind. Die Affinität zur Elektronik habe ich von meinem Vater geerbt, der ist auch ein Bastler.

PROGRAMMIERUNG DIV. PROJEKTE

Was wäre ein Programmierer wenn er nicht auch privat irgendwelche Projekte umzusetzen hätte. Gerade weil ich bei planetmutlu keine Zeit habe mich fortzubilden außerhalb dessen was gerade direkt gebraucht wird gibt es natürlich viel zu experimentieren auch nach der Arbeit. Viel sinnvolles kommt dabei nicht raus, aber man lernt eine Menge.

Projekte

2D GEOMETRIE LIBRARY

Da ich mit dem aktuellen Zustand der verfügbaren Slicer-Software für 3D-Drucker nicht zufrieden bin (CURA lässt sich nicht gut erweitern, Skeinforge ist tot, Slic3r ist Perl) habe ich mir vorgenommen einen eigenen Slicer zu bauen. Dafür benötigt man natürlich eine Menge 2D-Mathematik und vor allem einen schnellen Polygon-Clipper. Die meist benutzte Library ist "Clipper" und ist in C++ geschrieben, allerdings wurde der Verwendbarkeit halber alles in ein C-File und ein H-File gepackt und die Doku ist ein wenig unübersichtlich. Meine Geometrie Library sollte in C geschrieben sein und somit überall laufen, außerdem wollte ich einen anderen Clipping-Algorithmus verwenden der noch mehr Geschwindigkeit verspricht. Da ich lange nichts mehr mit Geometrie gemacht habe entschied ich mich selbst einen Clipper zu schreiben. Momentan funktioniert die Library aber noch nicht und es gibt nicht genügend Test-Cases um die Verwendbarkeit zu prüfen daher ist sie noch nicht auf Github.

Projekte (Fortsetzung)

ORM LAYER FÜR IOS

Für die CEWE-Fotowelt benötigten wir eine Möglichkeit die Fotobücher möglichst automatisiert und mit wenig Aufwand zu speichern. Auf iOS kommt einem da sofort CoreData in den Sinn, doch damals entschieden wir uns dagegen weil Migrationen von einem auf ein anderes Modell relativ häßlich werden können und keine Sync-Möglichkeiten mit der Cloud zur Verfügung standen. Zu diesem Zweck wurde ein ORM Layer entwickelt der die Objective C runtime nutzt um via Introspection das Datenmodell live zu erstellen. Das Klassen-Layout ist also direkt das Datenmodell das in einer SQLite Datenbank abgelegt wird. Das Format mit dem die Daten gespeichert werden sollte möglichst Menschenlesbar bleiben was mir leider nicht ganz gelungen ist. In der CEWE-Fotowelt-App ist mittlerweile eine erweiterte Version des Frameworks eingebaut das datenbankgestütztes Undo und Transaktionssynchronisation unterstützt.

Github: <https://github.com/dunkelstern/DSTStorageEngine>

DIE PLASTIKSCHMELZER (3D-DRUCKER PODCAST)

Mein ehemaliger WG-Kumpel, meine Freundin und ich haben seit ein paar Monaten einen eigenen Podcast in dem wir über 3D-Drucker reden. Das ganze ist nicht wirklich als professionell anzusehen sondern mehr als Erfahrungsbericht von Enthusiasten. Aufgenommen wird als Double-Ended, mein Kumpel in Karlsruhe schickt mir nachher ein MP3 mit seiner Tonspur und ich schneide dann alles zusammen. Zur Kommunikation verwenden wir Mumble da Skype an den Mikrofoneinstellungen rumschraubt. Mumble an sich hat aber eine zu schlechte Audio-Qualität um die Streams direkt aufzunehmen.

Website: <http://plastikschmelzer.chaotin.org>

THE IRON PHOTOGRAPHER

Der Eiserne Fotograf ist die Idee eines Social Networks für Fotografen. Die Idee ist, dass die User jede Woche über 3 Stichworte abstimmen (z.B. grün, kein Baum, hoch) und dann eine Woche lang jeder genau ein Foto abgeben kann zu diesem Thema. Die User stimmen dann über das beste Foto ab. Die Herausforderung dabei ist, dass die Fotos nur mit einem Smartphone eingereicht werden können und nur live mit der Kamera aufgenommen werden können. Das schränkt auf einer Seite die User ein, auf der anderen Seite müssen sie sich dann mehr Gedanken über Komposition machen. Jeder hat exakt einen Versuch, das Bild geht instant auf den Server.

TILED MAP RENDERER FÜR GUILDWARS 2

Guildwars 2 bietet interessierten Entwicklern eine API um auf ausgesuchte Inhalte aus dem Spiel zugreifen zu können. Ich habe mir die Aufgabe gemacht die In-Game Karte auf dem iPad darzustellen mit allen wichtigen Orten. Ursprünglich sollte es eine komplette Guildwars 2 App werden, aber die API ist dafür leider nicht ausreichen, so blieb es bei der Anzeige der Karte. Diese ist wie in der Maps-App frei scroll und zoombar, die Map-Tiles werden entsprechend vom Guildwars 2 Server geladen und alle Points of Interest kommen über die API.

Projekte (Fortsetzung)

SVG-RENDERER FÜR IOS

Ebenfalls für die CEWE-Fotowelt-App benötigten wir bei planetmutlu einen SVG-Renderer der nicht nur in CALayer-Objekte rendern kann wie SVGKit das tut sondern auch in beliebige CGContext Objekte (z.B. PDF Kontexte), außerdem musste Schrift Vektorisiert werden und nicht eingebettet werden (Einschränkung von CEWE). Mit SVGKit habe ich das nicht wie gewünscht hinbekommen, so musste eine eigene Lösung her. Auf libSVGTiny des Netsurf-Browser-Projektes entstand so innerhalb von 3 Tagen dieser SVG renderer.

Github: <https://github.com/dunkelstern/tinySVGRenderer>

UIPICKERVIEW-ERSATZ FÜR IOS (PRE IOS 7 STIL)

Da der UIPickerView von iOS relativ unflexibel ist (nur 2 verschiedene Höhen, nur Text) implementierte ich als Experiment den View selbst neu. Das API ist bis auf ein paar Erweiterungen kompatibel zum original. Die GUI wird komplett mit CoreGraphics gezeichnet (keine Bilder im Projekt)

Github: <https://github.com/dunkelstern/DSTPickerView>

JOYBOARD

JoyBoard ist ein kleines Windows Programm das Joystick und Gamepad eingaben in Tastatur und Mauseingaben wandelt. Das ist dann sinnvoll wenn man Spiele spielen möchte die normalerweise keine Gamepads oder Joysticks als Eingabegeräte akzeptieren.

Wenn man z.B. alte DOS Spiele in der DosBox spielen möchte fehlt einem oft die Möglichkeit einen Joystick verwenden zu können, hier kommt JoyBoard ins Spiel indem es die Joystickbewegungen in einfache Tastenkommandos übersetzt, so können selbst alte Spiele von den verbesserten Eingabemöglichkeiten profitieren.

Website: <http://projects.dunkelstern.de/JoyBoard/>

ARCHLINUX KDEMOD

Als KDE 3.5 aktuell war gab es auf ArchLinux nur monolitische KDE-Pakete, d.h. man bekam alles oder gar nichts, ein Bekannter aus Berlin (der mittlerweile leider verstorben ist) hatte die Idee KDE in viele einzelne Pakete zu splitten damit man sich nicht alles installieren musste um ein einzelnes Programm nutzen zu können. Er übernahm den i386 Build, ich habe den 64Bit Build betreut. Als wir dann alle Pakete gesplittet hatten begannen wir KDE schöner zu machen, Bugs zu fixen und alle möglichen Patches von überall her einzusammeln und zu integrieren. Als wir auch das komplett hatten begannen wir selber Erweiterungen zu schreiben und weitere Entwickler kamen dazu. Daraus entwickelte sich Chakra, eine Sub-Distribution von ArchLinux mit eigener Install-CD, eigenem Installer und dem besten KDE was es damals gab (unserer Meinung nach). Als dann KDE4 raus kam machte das Projekt keinen Sinn mehr und wurde Stück für Stück von uns Aufgegeben.

Apps

CEWE-FOTOWELT

Den Prototyp der CEWE-Fotowelt App Entwickelte ich bei planetmutlu als einziger Programmierer, nach ca. einem Jahr wurde mir von CEWE ein weiterer Kollege zur Seite gestellt.

Die App enthält eine Menge verschiedener Technik, wichtig zu erwähnen wären:

- Ein selbst geschriebener ORM Layer (siehe oben)
- Einen Renderer für UIViews und einen für PDF, sowie einen für CoreGraphics für Thumbnails
- XML Basierte Serverkommunikation + Datenupload
- Bildbearbeitungsfunktionen wurden mittels GPUImage und einem proprietären Bildbearbeitungsalgorithmus erstellt.
- Um selbst mit sehr großen Bildern arbeiten zu können wurde libTurboJPEG eingesetzt da diese es erlaubt JPEG-Bilder in reduzierter Auflösung zu laden
- SVG und Schriftvektorisierer damit keine Schriftarten in die verschickten PDFs eingebettet werden.

VAZZ VIDEOSTREAMING

Vazz ist das aktuellste Projekt von planetmutlu. Meine Verantwortlichkeit liegt in der Videoverarbeitung auf den Processing-Servern (skalierbar). Um Streaming von den Geräten in die Cloud zu ermöglichen und trotzdem den Hardware-Encoder in den Mobilgeräten verwenden zu können mussten einige Hürden übersprungen werden.

Unter anderem umgesetzt wurde ein kompletter MPEG-TS Muxer, ein MP4 demuxer und ein libx264 basierter Transcoder. Ich kenne mich also mit den Containerformaten MP4 und MPEG-TS sehr gut aus. Es wurde aus diversen Gründen kein FFMPEG eingesetzt sondern die Containerformate selber implementiert.

Um stotter- und ruckelfreie Streams zu ermöglichen musste der Bitstream der übertragen wurde genau untersucht werden und ich musste mich daher auch oberflächlich in H.264 und AAC einarbeiten.

Ansonsten war ich auch für das Release-Management der iOS App zuständig und habe an Bugfixes für die komplette App mitgearbeitet. Die Server-Kommunikation sowie der Video-Recording-Teil stammen aber zu 90 Prozent von mir.

POCKETCHRIS (PROGRAMMIERER, 1. IOS PROJEKT)

Mein erstes iOS Projekt für einen Bekannten Fotografen. Meine Vorzeigarbeit für die Bewerbung bei planetmutlu. Lässt sich relativ einfach mit Content füllen ohne die App umzuprogrammieren.

Apps (Fortsetzung)

NORDSEE (EINGESTELLT)

Die iPhone App der Nordsee Fischrestaurant-Kette. War eine einfache Informations-App mit Rezepten, Store-Finder und Nährwertinformationen. Die Daten wurden direkt aus dem CMS von Nordsee aus der Website ausgelesen (Python Scripte auf dem Server von planetmutlu haben die Daten aus der MySQL-DB in ein kompaktes JSON Format konvertiert). Die App Synchronisierte einmal am Tag die Daten und war danach offline Verwendbar. Komplette Umsetzung der App und des Backends war meine Aufgabe.

JAZZAHEAD FESTIVAL, SIXDAYS SPORTVERANSTALTUNG, HALLE7 VERANSTALTUNG

Lokale Events in Bremen. Alle drei Apps sind einfache Informations-Apps die ein ähnliches Sync-Framework wie die Nordsee-App einsetzen. Alle Apps wurden direkt an unterschiedliche CMS-Systeme angedockt.

INSTANYAN

InstaNyan war ein kleines Privatprojekt, ein Mashup aus NyanCat und Instagram. Wurde als Proof of Concept innerhalb von 2 Tagen von mir Entwickelt. Was ich zu dem Zeitpunkt nicht wusste: NyanCat ist Copyright geschützt, und so musste ich die App wieder Offline nehmen.

Bildbearbeitungsfilter waren simple aneinandergereihte CoreImage Filter. Man konnte seine Bilder auf Twitter, Facebook und via Mail teilen.