



IMST – Innovationen machen Schulen Top

Informatik kreativ unterrichten

APP-ENTWICKLUNG IN DER SCHULE

ID 778

DI Mag. Michael Gyarmati

BG/BRG Lerchenfeld

Klagenfurt, Juni 2012

Inhaltsverzeichnis

ABSTRACT	3
1 EINLEITUNG	4
1.1 Motivation	4
1.2 Rahmenbedingungen	5
1.3 Ziele	5
1.3.1 Ziele auf SchülerInnenebene	5
1.3.2 Ziele auf LehrerInnenebene	7
1.3.3 Verbreitung.....	7
1.4 Vorgangsweise.....	7
1.4.1 Phase 1 – Analyse der Plattformen	7
1.4.2 Phase 2 – Evaluation im Kern-Projektteam	8
1.4.3 Phase 3 – Einsatz im Regelunterricht	8
1.5 Zeitplan.....	8
2 PROJEKTVERLAUF	9
2.1 Phase 1 - Analyse der Plattformen	9
2.1.1 Welche Apps?	9
2.1.2 Welche Plattform	9
2.1.3 iOS.....	9
2.1.4 Android.....	10
2.1.5 iOS oder Android im Schulbetrieb?	10
2.2 Phase 2 - Evaluation im Kern-Projektteam.....	11
2.2.1 Die Schul-App	11
2.2.2 Die Spiel-App	12
2.2.3 Die Android-App	13
2.3 Phase 3 - Einsatz im Regelunterricht.....	14
3 ERGEBNISSE (DATENMATERIAL)	16
3.1 Ziele auf SchülerInnenebene	16
3.2 Ziele auf LehrerInnenebene	16
3.3 Verbreitung.....	17
3.4 Gender	17
4 RESÜMEE	18
5 REFERENZEN	19

ABSTRACT

Mit diesem Projekt soll evaluiert werden, inwiefern sich die Programmierung von Apps im Schulunterricht zur Motivation der SchülerInnen bzw. zur Wissensvermittlung eignet. Es soll evaluiert werden, inwiefern sich ein modernes Thema „Apps“ eignet, SchülerInnen für die Informatik im allgemeinen zu motivieren. Weiters soll versucht werden, mittels Apps „informatische“ Inhalte einzuführen.

Zum Zeitpunkt des Projektes standen Android und iOS als Plattformen zur App-Entwicklung zur Verfügung.

iOS ist einerseits bei den SchülerInnen beliebter und bietet andererseits auch die besser integrierte Entwicklungsumgebung. Leider fokussiert Apple eher auf Universitäten als auf Schulen und bietet daher keine Unterstützung für Schulen, weder finanziell noch durch adäquate Entwicklungsumgebungen.

Android unterstützt sowohl im Rahmen der Entwicklungsumgebung als auch bei den Endgeräten für die Apps höchst unterschiedliche Geräte, was sowohl Vor- als auch Nachteile mit sich bringt. Mit dem MIT App Inventor Edu existiert auch eine für den Unterricht angepasste, jedoch eingeschränkte, Entwicklungsumgebung.

In diesem Projekt stellte sich heraus, dass das Thema „Apps“ zwar aktuell und modern ist, jedoch die SchülerInnen sehr verwöhnt sind, was die Qualität von Apps anbelangt. Einerseits wollen SchülerInnen hochqualitative Apps erstellen, andererseits stellte sich ein Zeitaufwand von mehr als 2 Stunden bereits als demotivierend heraus, wenn danach nicht bereits alles den gewohnten Ansprüchen genügte.

Viele SchülerInnen ließen sich durch den für die Erstellung von Apps notwendigen Aufwand relativ rasch abschrecken. Weiters akzeptierten sie die Einschränkungen, die die Lehrumgebung „App Inventor Edu“ zur Zeit dieses Projektes noch mit sich brachte nicht.

Wenn die SchülerInnen schon nicht zum Entwickeln von eigenen Apps angeregt werden können, so kann zumindest eine Art Wertschätzung für die gewohnten Apps erwirkt werden.

Schulstufe: 9, 10, 11, 12

Fächer: Informatik

Kontaktperson: DI Mag. Michael Gyarmati

Kontaktadresse: BG/BRG Lerchenfeld, Lerchenfeldstr. 22, 9020 Klagenfurt

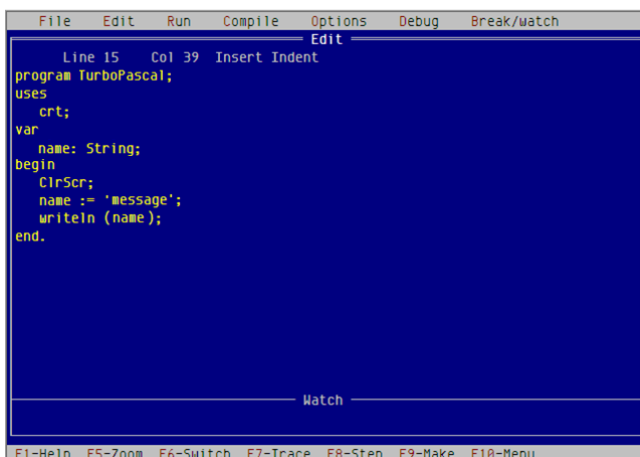
1 EINLEITUNG

1.1 Motivation

Smartphones prägen den heutigen Schulalltag. Der Großteil der SchülerInnen am BG/BRG Lerchenfeld besitzt ein modernes Smartphone, zumeist das iPhone 4 oder einen aktuellen Androiden. Vor dem Unterricht, in den Pausen, teilweise auch während und nach dem Unterricht kann man viele SchülerInnen mit ihren Smartphones beobachten. Die „Digital Natives“ (Palfrey & Gasser, 2008) sind angekommen, der ständige Einsatz des Computers ist heute „hip“, wer sich dem Internet, sozialen Netzen und Smartphones verweigert ist „out“.

Der Informatikunterricht leidet unter dem großen Problem, dass „coole“, sinnvolle Anwendungen so aufwendig sind, dass sie in der zur Verfügung stehenden Zeit und dem zur Verfügung stehenden Know-How der SchülerInnen nicht umsetzbar sind. Auch ist der Informatikunterricht nur ein Fach unter vielen, die alle um die Aufmerksamkeit und das Interesse der SchülerInnen buhlen.

Gerade in der heutigen Zeit lockt man niemanden mehr hinter dem Ofen hervor, wenn man ein Programm gestaltet, das den größten gemeinsamen Teiler (ggT) berechnen kann und auf der Konsole ausgibt. Computer sind allgegenwärtig und SchülerInnen wie Eltern sind hochprofessionelle Spiele und Anwendungen gewöhnt, haben aber in der Regel keine Vorstellung wie viel Personenmonate Entwicklungsarbeit in einem Produkt stecken.



```
File Edit Run Compile Options Debug Break/watch
Edit
Line 15 Col 39 Insert Indent
program TurboPascal;
uses
  crt;
var
  name: String;
begin
  ClrScr;
  name := 'message';
  writeLn (name);
end.

Watch

F1-Help F5-Zoom F6-Switch F7-Trace F8-Step F9-Make F10-Menu
```

Abbildung 1 - Turbo Pascal



Abbildung 2 - Lego Mindstorms NXT

Mikrowelten wie Scratch (MIT Media Lab, 2012) ermöglichen es SchülerInnen schnell und relativ einfach graphisch und akustisch ansprechende kleine Spiele zu erstellen. Aufgrund der graphischen Programmierweise fallen auch die hohen Einstiegshürden Befehle auswendig lernen zu müssen und diese syntaktisch richtig in einen Texteditor tippen zu müssen. Nach anfänglicher Euphorie zeigt sich dann jedoch, dass diese selbst erstellten Spiele stark von der vordefinierten Funktionalität der Mikrowelt abhängen und die gewohnte Professionalität von allgegenwärtigen Flash-Spielen und Apps nicht erreichbar ist. Weiters sind Spiele nur in dieser Mikrowelt lauffähig und können so schwer mit nach Hause genommen und schnell demonstriert werden. BYOB (Mönig, 2012) erweitert die Möglichkeiten von Scratch und erlaubt auch das Erstellen eines eigenständigen Programmes. Insgesamt entsteht jedoch zwar ein nettes Spiel, aber eine professionellere Variante gibt es mehrfach im Internet.

Das aktuell ebenfalls sehr beliebte Thema der Roboter z.B. Lego Mindstorms (Lego, 2012) ist zwar sehr spannend, kann aber nur schwer Freunden und Eltern demonstriert werden, da die Roboter nicht nach Hause mitgenommen werden können. Weiters muss beim Start des nächsten Projekts das

aktuelle Gebilde wieder zerstört werden. Auch die Genauigkeit der zur Verfügung stehenden Sensoren entspricht nicht immer den Erwartungen der SchülerInnen.

Während nun kaum jemand am ggT interessiert ist oder gar einen Roboter besitzt, ist die anfänglich erwähnte Durchdringung und Faszination mit Smartphones augenscheinlich. Es liegt daher nahe, es den SchülerInnen zu ermöglichen, eine App zu erstellen, die ihnen einerseits selbst nützt, andererseits aber auch "nicht-informatischen" SchülerInnen hilfreich ist und auf einfache Art und Weise den Eltern, Freunden und Bekannten demonstriert werden kann. Beispielsweise gibt es bereits Apps für den Zugriff auf WebUntis, das an vielen Schulen den Stunden- und Vertretungsplan liefert und somit eine Brücke zwischen Aushang von Zetteln bzw. Informationsmonitoren in der Schule zum Smartphone schlägt.

Der Haken an der App-Programmierung besteht jedoch in der hohen Komplexität der dazu notwendigen Konzepte, der Schnelllebigkeit der Plattformen und den Anforderungen an die Hardwareausstattung. Inwiefern und wie die App-Programmierung im Umfeld der AHS-Oberstufe gewinnbringend eingesetzt werden kann wurde in diesem Projekt untersucht.

1.2 Rahmenbedingungen

Das Projekt wird am BG/BRG Lerchenfeld in Klagenfurt im Schuljahr 2011/12 durchgeführt. SchülerInnen unterschiedlicher Schulstufen werden unterschiedlich stark eingebunden. Das Kernteam besteht aus 6 Schülern des Freifaches „Netzwerktechnik“, die aus der 5., 6. und 8. Klasse stammen und ein entsprechend differenziertes informatisches Vorwissen mitbringen. Mit dieser Gruppe werden bestehende Tools und Tutorials in Kleingruppen relativ eigenständig evaluiert.

Das im Kernteam erarbeitete Wissen und Material wird im zweiten Semester in der 5. Klasse im Informatikunterricht eingesetzt.

Als Lehrperson ist nur der Projektnehmer direkt in das Projekt involviert.

Folgende SchülerInnen sind im Projekt in unterschiedlichem Ausmaß involviert:

Kerngruppe: 6 Schüler der 5., 6. und 8. Klasse (Freifach 2 Wochenstunden)

Informatikklassen: 9 Schülerinnen und 17 Schüler der 5.-Klassen

1.3 Ziele

Die Ziele des Projektes werden in drei Ebenen für SchülerInnen, LehrerInnen und der Verbreitung getrennt betrachtet.

1.3.1 Ziele auf SchülerInnenebene

Die Ziele auf SchülerInnen-Ebene lassen sich in drei Hauptzielgruppen unterteilen.

1. Das Kern-Projektteam, das sich am stärksten mit dem Projekt befasst und die Vorarbeit für den Einsatz in den 5. Klassen und in Folgejahren leistet.
2. SchülerInnen der 5. Klassen, denen im 2. Semester ein Kurzeinstieg in die App-Programmierung ermöglicht werden soll.
3. Alle anderen SchülerInnen, die nicht unmittelbar im Projekt involviert sind, aber davon hören, bzw. vielleicht die eine oder andere entstandene App verwenden.

1.3.1.1 Das Kern-Projektteam

1. SchülerInnen sollen einen Einblick in die Komplexität der Softwareprojekte erhalten. Die Anzahl der Apps steigt zwar geradezu explosionsartig, die SchülerInnen sollen aber erkennen, dass hinter jeder Kleinigkeit viel Arbeit und Zeit steckt. Dadurch soll die Wertschätzung für

Projekte anderer gesteigert werden.

Überprüfung: Fragebogen zu Projektende. z.B. „Wie aufwendig dachtest du vor Projektstart dass eine App wird? Wie aufwendig war es wirklich?“

2. SchülerInnen sollen Teamwork steigern und Projektmanagement kennen lernen. Ein erstes Beispiel-App wird in Teilbereichen (Graphik, Sound, Programmierung, Dokumentation, Projektleitung) entwickelt. Dadurch soll die Zusammenarbeit und Kommunikation verbessert werden.

Überprüfung: Beobachtung des Projektfortganges. Gespräche mit den SchülerInnen. Beobachtung ob die Arbeit verteilt wird und inwiefern Deadlines gesetzt und auch eingehalten werden. Der Erfolg dieses Aspektes stellt sich dadurch dar, wenn das Beispiel-App ohne größere Eingriffe seitens der Lehrperson fertig gestellt wird.

3. SchülerInnen sollen erkennen, dass nicht alles Wissen von Lehrenden präsentiert werden kann sondern insbesondere aktuelles Wissen nur über selbständige Recherche im Internet anzueignen ist. Der Lehrende weist dazu auf Tutorials und Hilfen hin und versucht die SchülerInnen zum selbständigen Erarbeiten zu animieren.

Überprüfung: Tutorials und Hilfen werden durchgearbeitet und führen zu einem abgewandelten App. SchülerInnen sind in der Lage mit den Vorgaben eine eigenständige Anleitung und Tutorials zu verfassen.

4. SchülerInnen sollen lernen ihr angeeignetes Wissen anderen SchülerInnen in einer adäquaten Sprache weiterzugeben. Insbesondere müssen SchülerInnen der 8. Klasse einen Weg finden, den komplexen Inhalt soweit möglich und sinnvoll für SchülerInnen der 5. Klasse zugänglich zu machen.

Überprüfung: Die entstandenen Unterlagen und Tutorials können in einer 5. Klasse ohne grobe Verständnisschwierigkeiten durchgearbeitet und verstanden werden.

5. Abweichend zur häufigen Programmierpraxis „weltfremde“, abstrakte Programmieraufgaben zu lösen, sollen SchülerInnen einen Sinn in ihrer Arbeit sehen und daher für moderne Smartphones bzw. Tablets entwickeln. Dadurch wird eine höhere Motivation zur Mitarbeit erhofft und die Möglichkeit geboten, die entstandenen Projekte auch Freunden außerhalb der Schule und Eltern und Verwandten zu zeigen.

Überprüfung: Fragebogen zu Projektende: z.B. „Wem hast du die Apps gezeigt?“

6. SchülerInnen sollen erkunden welche schulbezogenen Apps (Lern- und Anschauungsprogramme aus allen Fachbereichen, etc.) bereits existieren und inwiefern sich diese für den Schulalltag tatsächlich eignen. Hierzu soll eine Liste empfehlenswerter Apps entstehen bzw. ein kurzer Bericht ob sich die Anschaffung eines Smartphones bzw. Tablets für den Schulgebrauch eignet. Insbesondere die App-Liste soll allen SchülerInnen und FachlehrerInnen als Empfehlung zugänglich gemacht werden.

Überprüfung: Bericht bzw. Liste geeigneter Apps.

1.3.1.2 Informatikgruppen der 5. Klassen

1. SchülerInnen sollen Informatik als allgegenwärtig und nützlich erkennen.

Überprüfung: Fragebogen über die Einstellung zur Informatik vor und nach der Teilnahme am Projekt. Auch die Tendenz zur Anmeldung zum Wahlpflichtfach wird als Kriterium herangezogen.

2. SchülerInnen sollen das im Informatikunterricht der 5. Klasse Gelernte (z.B. Bildbearbeitung, Mikrowelten) für die Erstellung eines Apps einsetzen können und somit einen Sinn hinter dem Informatikunterricht jenseits der Bedienung von Anwendungsprogrammen bemerken.

Überprüfung: Fragebogen vor und nach dem Projekt.

3. Ähnlich Scratch und anderen Mikrowelten sollen SchülerInnen schnell sichtbare Erfolge in Form eines kleinen Apps ermöglicht werden, welches sie dann auch zuhause herzeigen kön-

nen.

Überprüfung: Im Rahmen des Fragebogens, z.B. „Hast du dein App jemandem gezeigt?“

1.3.1.3 Andere SchülerInnen

1. Das Fach Informatik soll auch an SchülerInnen anderer Schulstufen als „aufregend“, modern und interessant herangetragen werden.

Überprüfung: Verbreitung der entwickelten App(s) an der Schule. Rückmeldungen von SchülerInnen und Eltern.

2. SchülerInnen sollen für die Informatik motiviert werden und erkennen, dass Informatik mehr als nur „Programmieren“ ist. Das Thema „Apps“ ist sehr modern und das tägliche Leben scheint sich allgemein um Facebook, Smartphones und Tablets zu drehen.

Überprüfung: Interesse an informatischen Freifächern bzw. Meldungen zum Wahlpflichtfach

1.3.2 Ziele auf LehrerInnenebene

1. Mit diesem Projekt soll die Eignung der modernen Apps für Unterrichtszwecke erkundet werden. Es nützt nichts, dass Apps modern sind, wenn die Entwicklung für Lernende und Lehrende zu kompliziert ist und auch nicht vereinfacht werden kann. Darum soll erkundet werden, welche Plattformen und Tools sich für den Einsatz in der Schule eignen und welche nicht.

Überprüfung: Insbesondere der Einsatz in den 5. Klassen wird zeigen, inwiefern die Mikrowelt „App-Entwicklung“ geglückt ist.

2. Es soll erkundet werden, inwiefern SchülerInnen komplexe Inhalte einander besser erklären können als dies durch die Lehrperson möglich ist. Der geringere Wissensunterschied und geringere Berührungshürden sprechen auf den ersten Blick dafür. Des Weiteren sollen die Unterlagen und Erklärungen von SchülerInnen für SchülerInnen auch als Hinweise und Muster für die Lehrperson dienen, um sich orientieren zu können, was gut verstanden wird.

Überprüfung: Im Idealfall ist ein direktes Erklären seitens der Lehrperson nicht mehr notwendig und die Unterlagen und Erklärungen der MitschülerInnen reichen im Großen und Ganzen aus.

1.3.3 Verbreitung

1. Da Apps theoretisch sehr einfach im App-Store weltweit publiziert werden können, kann die „Leistung“ der SchülerInnen auch sehr einfach verbreitet werden. Durch Angebot einer nützlichen "BG Lerchenfeld App" wird die Leistung für SchülerInnen, LehrerInnen und Eltern der Schule ersichtlich.

Überprüfung: Veröffentlichung im App-Store und Downloadzahlen.

1.4 Vorgangsweise

Die Projektdurchführung kann in drei Hauptphasen aufgeteilt werden.

1.4.1 Phase 1 – Analyse der Plattformen

In der ersten Phase werden in Frage kommende Plattformen identifiziert und bezüglich Vor- und Nachteile analysiert. Aufgrund der aktuellen Marktlage zum Zeitpunkt des Projektstartes und dem Vorhandensein von Endgeräten in der Schule kommen iOS und Android in Frage, da Windows Phone erst in den Startlöchern stand. Es gilt zu identifizieren, welche Hardware- und Softwarevoraussetzungen zu erfüllen sind und welche formalen Hindernisse zu bewältigen sind.

1.4.2 Phase 2 – Evaluation im Kern-Projektteam

Basierend auf den Erkenntnissen aus der Analysephase werden die notwendige Hardware ausgewählt, Angebote eingeholt und die Hardware besorgt. Außerdem werden notwendige Entwicklerzugänge registriert und bestehende Tutorials, Howtos und Dokumentationen recherchiert. Weiters werden die Entwicklungsumgebungen für den Schulgebrauch vorbereitet und evaluiert.

Nach dem Setup und der Recherche- und Einlesephase werden durch das Kernteam Ideen für erste eigene Apps gesammelt, die Apps geplant und in Kleingruppen auf verschiedenen Plattformen möglichst selbstständig durch die SchülerInnen umgesetzt.

Die Erkenntnisse und Erfahrungen der SchülerInnen des Kernteams sollen in Dokumentationen und Tutorials bzw. Beispiele für andere SchülerInnen einfließen und in der folgenden Phase eingesetzt werden können.

1.4.3 Phase 3 – Einsatz im Regelunterricht

Die am besten geeigneten Tools und Plattformen werden im zweiten Semester im Informatikunterricht ausgewählter 5. Klassen eingesetzt. Diese sollen wenn möglich mit den Dokumentationen der Kerngruppe in die Lage versetzt werden, einfache schnelle Erfolge in der App-Programmierung zu erlangen. Nötigenfalls erfolgt eine Unterstützung durch die SchülerInnen des Kernteams bzw. durch die Lehrperson.

Neben einem allgemeinen, verpflichteten Schnuppern in die App-Programmierung für alle SchülerInnen besteht sodann die Wahlmöglichkeit ein informatisches Thema zu recherchieren und darüber zu referieren oder alternativ etwas tiefer in die App-Programmierung vorzudringen.

1.5 Zeitplan

Juni – September:	Vorbereitung durch die Lehrperson
September:	Analyse der Plattformen, organisieren der notwendigen Hard- und Software
Oktober:	Planung der Apps, Zuteilung der Kleingruppen
November – Jänner:	Recherche, Einlesen und Entwicklung der erst Apps
Jänner:	Zwischenevaluation
Februar – März:	Fertigstellung der Apps und Dokumentation
April – Mai:	Einsatz der App-Programmierung in den 5. Klassen
Juni:	Endevaluation

2 PROJEKTVERLAUF

2.1 Phase 1 - Analyse der Plattformen

2.1.1 Welche Apps?

In einer Brainstorming-Runde kristallisierten sich relativ rasch 2 Apps heraus, die die Schüler gerne machen wollten. Eine eigene Schul-App soll neben Stundplan und Supplierplan auch aktuelle Informationen von Läufern und Termine der Homepage anzeigen. Eine andere Idee war es, ein Spiel zu programmieren. Während die Ideen für die erstere App relativ rasch konkreter wurden, änderte sich die Spiel-Idee noch relativ häufig.

2.1.2 Welche Plattform

Bezüglich der Plattform standen eigentlich nur zwei Systeme zur Auswahl: iOS einerseits und Android andererseits. Die Verbreitung von Windows Phone war zu Beginn des Projektes noch vernachlässigbar.

Aufgrund der Vorlieben der SchülerInnen und der hohen Dichte von iPhones an der Schule sympathisierten die Schüler relativ rasch mit der Idee für iOS zu entwickeln. Insbesondere auch bezüglich der Idee einer Schul-App mit Zugriff auf den Stundenplan (Webuntis¹) war iOS vielversprechender, da es für Android bereits eine akzeptable Stundenplan-App (SchoolPlanner4Untis²) gibt.

Um beide Plattformen im Schulbetrieb vergleichen zu können sollten 3 Apps entstehen: 1 Schul-App in iOS, ein graphisches Spiel für iOS und ein Quiz für Android.

2.1.3 iOS

Da weder der Lehrer noch die Schüler auf Erfahrung mit iOS zurückblicken konnten, galt es geeignete Tutorials und Literatur zu finden und eine iOS-Entwicklungsumgebung aufzusetzen.

Leider stellte sich recht rasch heraus, dass es Apple angehenden Entwicklern nicht gerade einfach macht. Im Gegenteil hat man das Gefühl dass vieles unnötig verkompliziert wird. Neben der Notwendigkeit einer kostenpflichtigen Inskription im iOS-Entwickler-Programm (USD 100.- pro Jahr) stellt sich auch der Zwang auf Macs zu arbeiten als unvorteilhaft heraus. Letzteres versuchten wir zunächst durch das Arbeiten in virtuellen Maschinen zu umgehen, was sich aber auch als nicht praktikabel herausstellte. Glücklicherweise besitzen einige Schüler aber Macbooks.

Zugute halten muss man iOS hingegen, dass sofern die administrativen Hürden seitens Apple gemeistert sind, die Entwicklungsumgebung ohne Probleme in Betrieb genommen werden kann und auch für SchülerInnen ein schneller erster Erfolg in Form einer Beispiel-App möglich ist. Weiters existieren viele, gute Tutorials im Internet, die den Schülern auch einen selbständigen Einstieg ermöglichen.

Leider schreitet die Entwicklung von iOS rasant voran. So kam kurz nach Projektstart iOS 5 heraus, und mittlerweile gibt es bereits die ersten Beta-Version von iOS 6. Die Konsequenz aus dieser Entwicklung ist, dass Bücher, Tutorials und Beschreibungen relativ rasch veralten.

Als weiters ungünstig stellte sich heraus, dass Apps nur auf „Entwicklergeräten“ installiert werden können. D.h. jedes Gerät muss am Lehrergehärt für die App durch Aufbringung eines Zertifikates „freigeschaltet“ werden. Dies gestaltet sich etwas umständlich und ist zudem auf 50 Geräte limitiert.

¹ Webuntis bietet Zugriff über das Internet auf Untis, eine gebräuchliche Stundenplansoftware von Gruber & Petters (<http://www.grupet.at/>).

² SchoolPlanner4Untis ist eine Android-App von Schülern der HTL Rennweg, die den Zugriff auf Webuntis bietet (<http://www.schoolplanner.at/>).

2.1.4 Android

Im Gegensatz zu iOS begnügt sich Android mit einem beliebigen PC (Windows, Linux, Mac OS) und baut auf der bekannten Eclipse Entwicklungsumgebung auf. Leider gestaltete sich das Aufsetzen einer Entwicklungsumgebung als etwas komplizierter.

Bei Android gestaltet sich das Aufsetzen der Umgebung etwas komplizierter. Nachdem mehrere Versuche die Umgebung im Unterricht durch die Schüler auf den Windows basierten Schulrechnern einzurichten, scheiterten wurde schlussendlich ein Linux basiertes USB-Live-System durch die Lehrkraft vorbereitet und den SchülerInnen (auch für zuhause) überlassen. Um nur einige Probleme zu nennen:

- Das Android-SDK kommt mit der Umleitung diverser Ordner auf das Netzlaufwerk nicht zu recht
- Für das Nachinstallieren von Paketen in Eclipse ist der verpflichtete Proxy-Server recht ungünstig
- Die gesamte Entwicklungsumgebung ist größer als das zur Verfügung stehende Quota der Schüler
- Unter Windows ist für jedes Smartphone-Modell ein eigener Treiber nötig

Nach dem Umstieg auf Linux (Ubuntu 12.04) auf 8 GB USB Sticks lösten sich all diese Probleme. Die Umgebung wird nur einmal durch die Lehrkraft vorbereitet und dann 1:1 auf Sticks kopiert. Treiber sind unter Linux keine nötig und das System kann so von den SchülerInnen auch zuhause verwendet werden.

Im Gegensatz zu iOS kostet das Android Developer Programm einmalig 25 USD für die Registrierung und keine jährliche Gebühr.

2.1.5 iOS oder Android im Schulbetrieb?

Obwohl dieses Projekt mit Android und iOS durchgeführt wurde und der Schwerpunkt eher auf iOS lag, stellte sich heraus, dass für den Schulbetrieb eher Android der Vorzug zu geben ist. Die Begründung dafür soll im Folgenden gegeben werden.

Für die App-Entwicklung benötigt man nicht unbedingt ein Smartphone. Unbedingt notwendig ist jedoch ein geeigneter Computer. Für iOS kommt nur ein Mac in Frage. Dazu sollten mehrere Macs in der Schule verfügbar sein. Sofern diese nicht vorhanden sind müssten z.B. Mac Minis im Wert von min. EUR 500 pro Stück angeschafft werden. Für Android ist jeder moderne PC ausreichend, wobei der Emulator relativ viel Ressourcen benötigt.

Der Emulator für Android stellt eine echte virtuelle Maschine dar, d.h. es wird ein tatsächliches Android am Rechner ausgeführt. Dadurch ist besonders der erste Start sehr langsam. Für iOS wird im Simulator die App nativ unter Mac OS ausgeführt. Das ist zwar schneller, entspricht dann aber nicht dem Verhalten am echten Gerät.

Als Entwicklungssprache kommt entweder Objective C (iOS) oder Java (Android) zum Einsatz. Objective C stellt sich zumindest bis iOS 5 als sehr ungünstig für den Schulbetrieb dar, da sich SchülerInnen selber um die Speicherverwaltung kümmern müssen. Ab iOS 5 nimmt Automatic Reference Counting (ARC) dem Entwickler die Speicherverwaltung größtenteils ab. Jedoch sind manche externe Bibliotheken noch nicht ARC-fähig. Java hingegen kommt mit einem Garbage Collector, so dass man sich nicht um die Speicherverwaltung kümmern muss. Es ist auch anzunehmen, dass mehr LehrerInnen bereits Kontakt mit Java als mit Objective C hatten.

Als Entwicklungsumgebung kommt für iOS XCode zum Einsatz. Diese Umgebung ist gratis, aber nur unter MacOS verfügbar. Sie ist sehr mächtig und gut integriert, d.h. alles passt zusammen. Für Android wird Eclipse als Entwicklungsumgebung empfohlen. Auch diese Umgebung ist sehr mächtig, gratis und auf verschiedenen Betriebssystemen lauffähig.

Um überhaupt iOS Apps installieren zu dürfen, muss man die jährlichen 99 USD für den Development Account ausgeben. Dieser berechtigt dazu, auf 50 Geräten eigene Apps zu installieren und Apps in den App Store einzureichen. Insbesondere die im Unterricht entstehenden Apps werden die Ansprüche für den Einzug in den App Store nicht erfüllen, so dass die SchülerInnen ihre Apps nicht installieren werden können. Für Android ist eine einmalige 25 USD Gebühr nur dann fällig, wenn man in den Play Store möchte. Apps lassen sich jedoch auch ohne diesen Development Account direkt als APK installieren. Davon abgesehen ist es unter Android viel einfacher in den Play Store zu gelangen, da Google Apps erst im Nachhinein im Anlassfall überprüft und wieder entfernt.

Endgeräte (Smartphones, Tablets) gibt es für Android sehr viele. Dies ist bezogen auf den Preis ein großer Vorteil. Für die SchülerInnen ist dies jedoch auch ein Nachteil, da die App viele verschiedene Auflösungen und Geräte unterstützen muss. Unter iOS gibt es im Prinzip nur einige wenige Geräte.

Insgesamt ist iOS im Schulbetrieb nur dann zu empfehlen, wenn die Schule bereits Apple-Geräte einsetzt und die Lehrkraft Apple-affin ist. Andernfalls sollte ein Linux-Live-System auf USB vorbereitet werden um nicht zu viel Zeit mit dem Setup der Entwicklungsumgebung zu verwenden.

2.2 Phase 2 - Evaluation im Kern-Projektteam

Der ursprüngliche Plan hatte vorgesehen, dass in einem Kern-Projektteam im ersten Semester die App-Entwicklung evaluiert und erarbeitet wird. Allerdings ergaben sich mehrere Verzögerungen, die dazu führten, dass die Arbeit im Kernteam sich in das zweite Semester erstreckte. Beispielsweise benötigte bereits die Anforderung eines Apple Development Accounts mehrere Wochen, da es im automatisierten Ablauf zu „Ungereimtheiten“ kam. Bei Android kam es zu Verzögerungen durch Probleme mit den Windows-Rechnern im Schulnetz, die die Entwicklungsumgebung nicht einfach kopieren ließen.

Jedoch entstanden im Kern-Projektteam schlussendlich drei Apps, die im Folgenden kurz vorgestellt werden sollen.

2.2.1 Die Schul-App

Die Schul-App sollte ursprünglich für das BG/BRG Lerchenfeld spezifische Informationen anbieten, auf die sowohl SchülerInnen als auch LehrerInnen und Eltern Zugriff haben. Dazu gehört einerseits der aktuelle Stundenplan, aktuelle Supplierungen und sonstige Stundenplanänderungen, aber auch spezieller aktuelle Termine von der Homepage, Schularbeitentermine, Läufer und Hausübungen.

Diese App wird von zwei Schülern der Maturaklasse erstellt. Während einer sich um die eigentliche App unter iOS kümmert, beschäftigt sich der andere mit dem Server und der Herkunft der Daten.



Abbildung 3 - School Organizer

Die Daten für den aktuellen Stundenplan werden über JSON-RPC von WebUntis geholt und in einer lokalen SQLite-Datenbank zwischengespeichert, damit sie auch ohne Internetzugriff verfügbar sind. Ergänzt werden die Stundenplandaten durch weitere Informationen von einem eigenen Schulserver, der Termine, den Speiseplan der Kantine, und Läufer (Nachrichten der Schulverwaltung an Schülerinnen) bereithält.

Als besondere Herausforderung für den Schüler stellte sich die Übertragung der Daten über JSON-RPC und das lokale Speichern in SQLite dar. Hier war es notwendig, dass die Lehrkraft unterstützend eingriff. Die restliche App konnte selbständig erarbeitet werden. Im App Store ist die App noch nicht zu finden, sie soll jedoch in den nächsten Wochen bis Monaten „professionalisiert“ werden und dann hoffentlich ihren Weg in den App Store finden.

2.2.2 Die Spiel-App

Nach anfänglichen Schwierigkeiten ein geeignetes Spielkonzept zu erarbeiten das es einerseits noch nicht gab und andererseits auch machbar erschien, kam ein Schüler der Maturaklasse schlussendlich auf ein nettes kleines Spiel und ein geeignetes Framework zur Spieleentwicklung namens cocos2d.

Bei dem Spiel „20 Seconds“ handelt es sich um ein Lern-/Geschicklichkeitsspiel, bei dem es gilt ein Wort aus einem Schulfach mittels eines Hinweises möglichst schnell zu Erraten und in einem Buchstabenfeld zu finden. Nach korrekter Eingabe des Lösungswortes erhält man noch Zusatzinformationen zum aktuellen Thema. Es bedient sich dabei auch den ansprechenden optischen und musikalischen Effekten die das Spieleframework cocos2d relativ einfach anbietet.

Auch dieses Spiel findet man derzeit noch nicht im App-Store, da noch nicht alle Fragen und Themenbereiche ausgearbeitet sind und in manchen Fällen noch das Memory-Management zu einem „Einfrieren“ führt.

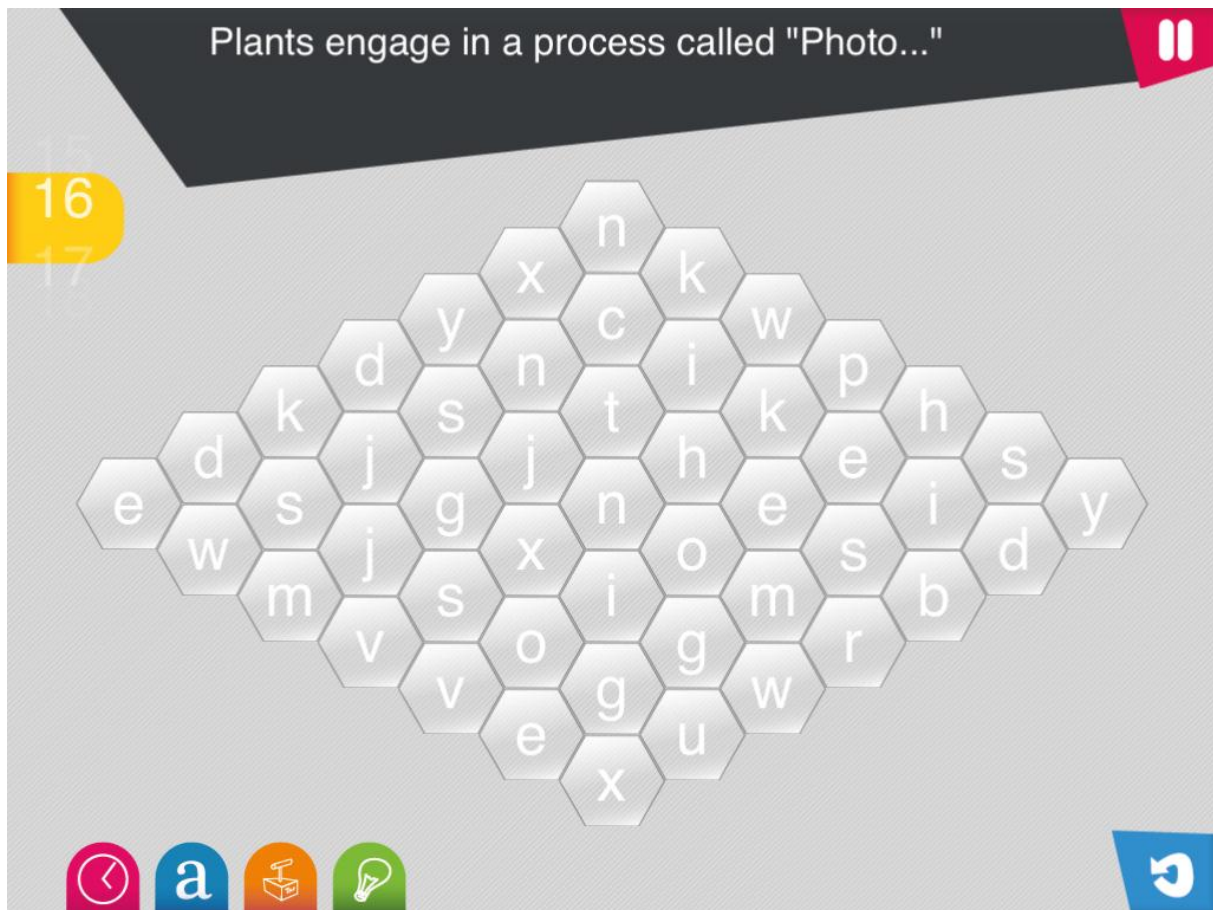


Abbildung 4 - 20 Seconds

2.2.3 Die Android-App

Da die vorangehenden Apps bereits einige Programmierkenntnisse und informatisches Wissen erfordern suchten wir für die jüngeren Schüler ein geeignetes Thema und fanden eines in Form eines Spieles „Wer wird Wissionär“, das auf der Fernsehsendung „Wer wird Millionär“ basiert. Die Fragen sollten aus dem Schulstoff stammen und entsprechend dem Vorbild 4 Antwortmöglichkeiten und Joker bieten.

Die Erarbeitung der dazu nötigen Fragen und Antwortmöglichkeiten aus verschiedenen Fachbereichen wurde einerseits durch das Projektteam begonnen. Andererseits wurden und werden im regulären Informatikunterricht der 5. Klasse informatische Fragen und Antworten von den SchülerInnen zusammengestellt. Diese sollen schlussendlich kombiniert und in das Spiel integriert werden.

Die App selbst wird von 2 Schülern der 6. Klasse mit Unterstützung durch die Lehrkraft erarbeitet, da die Schüler, die diese App entwickeln noch keine Programmierkenntnisse haben und auch nicht den Wahlpflichtunterricht besuchen konnten. Insofern wird neben der App-Entwicklung auch gleich die allgemeine Programmierung und objektorientierte Programmierung behandelt.

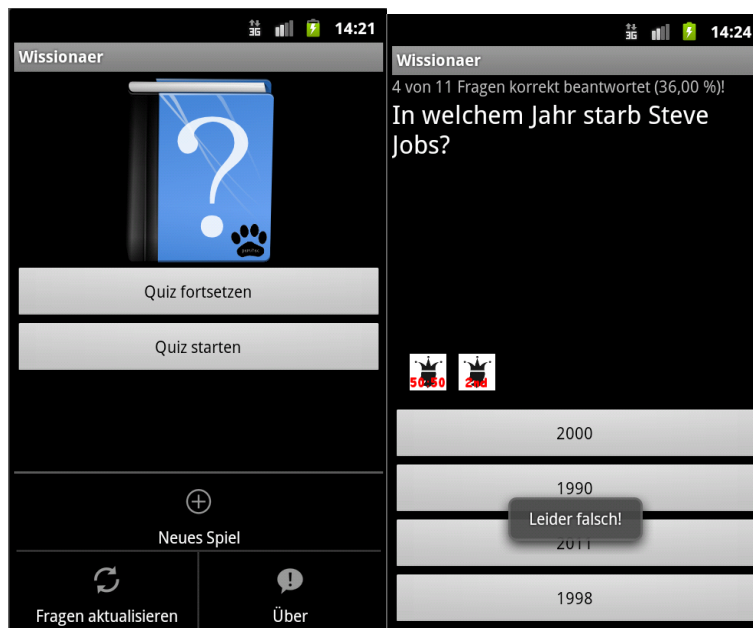


Abbildung 5 - Wissionär

Das Erstellen der GUI der App stellte sich als machbar heraus. Die Verknüpfung der GUI mit dem Programmcode war erwartungsgemäß schwierig, da die Grundlagen erst erarbeitet werden mussten. Jedoch eignet sich das App gut als Motivation für die SchülerInnen, um neue Ideen zu erarbeiten und diese dann mit Unterstützung durch die Lehrkraft umzusetzen.

2.3 Phase 3 - Einsatz im Regelunterricht

Im zweiten Semester sollte das Thema App-Programmierung in den Regelunterricht in der 5. Klasse ausgeweitet werden. Da sich die Arbeit im Kern-Projektteam verzögerte, wurde in der 5. Klasse zunächst ein sanfter Einstieg in die Programmierung mittels Scratch bzw. BYOB durchgeführt.

Leider musste ich feststellen, dass bereits dies bis auf wenige Ausnahmen nicht bei sonderlich vielen SchülerInnen gut ankam. Als Projektarbeit im Unterricht stellte ich den SchülerInnen frei, entweder ein Thema zu recherchieren und eine schriftliche Arbeit und einen Vortrag dazu zu verfassen und zu halten, oder sich mit Scratch-Programmierung oder App-Programmierung zu beschäftigen. Leider konnte ich in diesem Schuljahr niemanden für die App-Programmierung begeistern und wollte aber andererseits auch niemanden zwingen.

Vielversprechend für die 5. Klasse klang zwar „App Inventor Edu³“, eine Entwicklungsumgebung vom MIT, das auf dem mittlerweile eingestellten „Google App Inventor“ basiert. „App Inventor Edu“ ähnelt Scratch bzw. LEGO Mindstorms und soll das Entwickeln von Apps für die Schule vereinfachen. Leider war das Tool noch sehr neu und noch nicht öffentlich verfügbar sondern musste selbst installiert werden. Erst in den letzten Wochen des Projektes wurde die Entwicklungsumgebung öffentlich über die Seite des MIT zugänglich.

Bei einem kurzen Einsatz im Unterricht musste ich feststellen, dass die Umgebung im damaligen Zustand nicht den hohen Anforderungen der SchülerInnen genügen konnte und die SchülerInnen dementsprechend relativ rasch demotiviert waren, da die gewohnte Qualität einfach nicht erreichbar war. Nach einer kurzen Demonstration der Möglichkeiten und des Umganges mit App Inventor Edu ließ sich zumindest in diesem Schuljahr leider niemand dazu motivieren ein kleines Projekt damit zu machen.

³ siehe <http://appinventoredu.mit.edu/>

Zumindest wurden in den 5. Klassen für behandelte Themen durch die SchülerInnen Quizfragen für das App „Wer wird Wissionär“ ausgearbeitet. Hier fehlte die unmittelbare Motivation, da das App noch nicht existierte und somit die Fragen einstweilen nur in einer Excel-Arbeitsmappe verblieben.

3 ERGEBNISSE (DATENMATERIAL)

Leider gelang es in diesem Schuljahr nicht, mit App-Programmierung in der 5. Klasse SchülerInnen für unmittelbare App-Projekte zu motivieren. Andererseits gab es in diesem Schuljahr (2011/12) keine Wahlpflichtfachgruppe Informatik, im folgenden Schuljahr (2012/13) kommen jedoch zwei Gruppen zustande. Auch die beiden Matura-Projekte in diesem Schuljahr können als Erfolg gewertet werden.

3.1 Ziele auf SchülerInnenebene

Das Kern-Projektteam hat sicherlich Einblick in die Komplexität der Softwareentwicklung, insbesondere der App-Entwicklung erhalten. Im Rahmen des Fragebogens haben alle befragten SchülerInnen zugestimmt, dass die App-Entwicklung viel komplizierter und aufwändiger war als erwartet.

Bezüglich des Teamworks der SchülerInnen konnte beobachtet werden, dass dies im Maturajahrgang sehr gut funktionierte. Ein anfänglich geplantes grosses Projektteam mit verschiedenen Aufgaben (Graphik, Sound, Programmierung, Dokumentation, etc.) verlief leider im Sand. Hier stellten sich die verschiedenen anderen Fächer als zu große „Ablenkung“ dar, da immer eine Schularbeit, ein Test oder eine sonstige Ablenkung als „Ausrede“ angeführt werden konnte.

Generell wurden Tutorials und Hilfen durchgearbeitet und somit Wissen selbständig erarbeitet. Dies gilt insbesondere für die Maturanten, die ihre Projekte grossteils selbständig erfüllten. Aber auch in der 6. Klasse wurden grundlegende Tutorials durchgearbeitet, obgleich diese nicht vollständig ohne Eingreifen durch die Lehrkraft verstanden wurden.

Der Wunsch, dass SchülerInnen Ihr erarbeitetes Wissen den anderen SchülerInnen weitergeben, konnte im Rahmen dieses Projektes nur teilweise erreicht werden, da sich die Erarbeitung des Wissens als aufwendiger als erwartet erwies und dadurch wenig Zeit zur Wissensweitergabe blieb. Die Hoffnung besteht, dies im nächsten Schuljahr nachzuholen.

Leider wurden in den 5. Klassen keine Apps erstellt, die den Eltern oder sonstigen Aussenstehenden hätten gezeigt werden können. Die beiden Maturanten hatten die Möglichkeit Ihr Projekt im Rahmen der Matura zu präsentieren. Weiters scheinen beide Schüler motiviert über den Sommer weiter an ihren Apps zu arbeiten und diese zur App-Store-Reife zu führen.

Im Zuge dieses Projektes hatten die SchülerInnen auch die Möglichkeit im normalen Schulunterricht die Einsatzmöglichkeiten von Tablets und Smartphones zu evaluieren. Nach anfänglicher Begeisterung stellte sich jedoch relativ rasch heraus, dass die meisten Lern-Apps relativ rasch uninteressant werden bzw. trotz Tablet weiterhin am Papiausdruck besser gelernt werden kann.

In den 5. Klassen war zwar nicht unmittelbar jemand zur Teilnahme an der App-Entwicklung zu begeistern, jedoch konnten für das nächste Schuljahr statt 0 gleich 2 Wahlpflichtfach-Gruppen motiviert werden.

Weder das Spiel-App noch das Schul-App sind bereits veröffentlicht. Dadurch kann leider noch keine „Verbesserung“ des Bildes der Informatik in der restlichen Schulwelt evaluiert werden. Allerdings ist geplant, dass beide Apps spätestens im nächsten Schuljahr im App-Store zur Verfügung stehen.

3.2 Ziele auf LehrerInnenebene

Der erwartete und erhoffte Effekt der Motivation der Lernenden durch die App-Programmierung fiel leider nicht so stark wie erwünscht aus. Lediglich vereinzelt konnte ein Motivationsschub erreicht werden. Insbesondere in den 5. Klassen wählten die SchülerInnen den Weg des vermeintlich geringsten Widerstandes.

Das Feedback der Eltern im Rahmen der Elternsprechtage fiel hinsichtlich des Themas App-Programmierung durchaus positiv aus. Hier ist jedoch anzumerken, dass erwartungsgemäß nur sehr wenige Eltern überhaupt den Elternsprechtage nutzten, da Informatik kein Hauptfach darstellt.

Da die älteren Schüler relativ lang mit ihren eigenen Problemen (Matura) beschäftigt waren, konnte die Wissensweitergabe in diesem Schuljahr leider nicht wie erwünscht beobachtet werden. Es bleibt die Hoffnung, dass in den nächsten Schuljahren im Rahmen von Informatik Wahlpflicht und Freifächern noch mehr Wissen von den Schülern an SchüelrInnen weitergegeben werden kann.

3.3 Verbreitung

Leider haben die Apps noch nicht die Ansprüche des App-Stores erreicht so dass sie veröffentlicht werden könnten. Es ist jedoch sowohl im Interesse der Maturanten als auch der Lehrkraft, dass zumindest die beiden Maturaprojekte in den nächsten Wochen bis Monaten einen erfolgreichen Einstieg in den App-Store finden und im nächsten Schuljahr zur Verfügung stehen.

3.4 Gender

Im Kern-Projektteam war leider keine Schülerin gemeldet. Auch in den 5. Klassen gelang es trotz mehrmaligem Nachfragen nicht, eine der Schülerinnen zum Thema Scratch oder App-Programmierung zu motivieren. Sie wählten die vermeintlich einfachere Tätigkeit der Recherche eines „informatischen“ Themas mit anschließendem Vortrag.

4 RESÜMEE

App-Entwicklung in der Schule ist ein modernes Thema, das durchaus Potential besitzt SchülerInnen für Informatik zu motivieren. Sowohl iOS als auch Android bieten einen gewissen Reiz. Leider erfordern beide Systeme ein gewisses Grundverständnis der Programmierung und besonders ein Mindestmaß an Standhaftigkeit, da sich nach anfänglichen Erfolgen relativ rasch demotivierende Schwierigkeiten in der App-Entwicklung einstellen. Dafür ist ein gewisses Durchhaltungsvermögen erforderlich, welches sich in diesem Schuljahr eigentlich nur in der Maturaklasse feststellen lies.

Während Apple zwar bei den SchülerInnen beliebt ist, interessiert sich Apple nicht für den Nachwuchs in der Schule, da es keine schulbezogenen Programme und Vergünstigungen gibt. Der Fokus im App-Store liegt eher auf Qualität als auf Motivation von Nachwuchsprogrammierern. Diesbezüglich erweist sich Android als die geeignetere Plattform, da damit zumindest die Installation abseits des offiziellen Play Stores ermöglicht wird. Außerdem gibt es für Android eine vielversprechende Lehrumgebung ähnlich zu Scratch, die während dieses Projekts leider erst im Entwicklungsstadium war und daher nur eingeschränkte Möglichkeiten bot.

Durch die Einschränkung der iOS-Entwicklung auf Mac-Geräte erscheint die Entwicklung für iOS aufgrund der erhöhten Hardware-Anschaffungskosten in der Schule als nachteilig.

Der Einsatz der App-Programmierung in der 5. Klasse erscheint aufgrund der gemachten Erfahrungen zu verfrüht. Im Wahlpflichtunterricht ab der 6. Klasse könnten Apps zur Einführung und Motivation von objektorientierten Programmierparadigmen dienen. Aber auch hier hat sich zumindest in diesem Schuljahr der notwendige Aufwand zur Erfüllung der sehr hohen Erwartungen der SchülerInnen eher als demotivierend und abschreckend herausgestellt. Hinzu kommt noch, dass die SchülerInnen durch Schularbeiten und Tests in anderen Fächern „abgelenkt“ werden und somit bei den meisten letztlich nur 2 Stunden pro Woche übrig bleiben. Lediglich im Maturajahrgang konnte ein Erfolg mit zwei Apps verbucht werden, da hier sowohl die Motivation als auch die Grundlagen der SchülerInnen weiter fortgeschritten ist.

5 REFERENZEN

Apple Inc.: *iOS Developer Library*, unter:

<https://developer.apple.com/library/ios/navigation/index.html> (abgerufen am 10.07.2012).

Conway Joe, Hillegass Aaron: *IOS Programming: The Big Nerd Ranch Guide*, 3. Auflage, Amsterdam, 2012.

Itterheim Steffen, Löw Andreas: *Learn cocos2d Game Development with iOS 5*, New York, 2011.

Nahavandipoor Vandad: *iOS 4 Programming Cookbook*, Sebastopol, 2011.

Google Inc.: *Android API Guides*, unter: <http://developer.android.com/guide/components/index.html> (abgerufen am 10.07.2012).

Wolber David, et. al: *App Inventor: Create Your Own Android Apps*, Sebastopol, 2011.

Louis Dirk, Müller Peter: *Jetzt lerne ich Android: Der schnelle und einfache Einstieg in die Programmierung und Entwicklungsumgebung*, München, 2011.

Woltmann Sven: *Android Anfänger Workshop*, unter:

http://www.androidpit.de/de/android/wiki/view/Android_Anf%C3%A4nger_Workshop (abgerufen am 10.07.2012).