



IMST – Innovationen machen Schulen Top
Informatik kreativ unterrichten

APP-PROGRAMMIERUNG

ID 929

Prof. Mag. Cornelia Stadtmann

HLW St. Veit

St. Veit an der Glan, Juli 2013

INHALTSVERZEICHNIS

ABSTRACT	3
1 MOTIVATION.....	4
1.1 Ausgangssituation	4
1.1.1 Schulische Rahmenbedingungen an der HLW.....	4
1.1.2 Schulische Rahmenbedingungen an der NMS 1.....	4
1.2 Lehrplan.....	4
1.3 Projektziel	5
1.4 Ziele auf SchülerInnenebene	5
1.5 Ziele auf LehrerInnenebene	6
1.6 Genderebene.....	6
1.7 Zeitplan	7
2 APP-PROGRAMMIERUNG	8
2.1 Voraussetzungen der Hard- und Software	8
2.2 Voraussetzungen der Lehrperson	8
2.3 Grundlegende Vorbereitungen für die SchülerInnen.....	8
3 PROJEKTVERLAUF.....	9
3.1 App-Programmierung mit den SchülerInnen der HLW	9
3.2 Durchführung und Beispiele	9
3.2.1 Zusatz zu 2. Phase.....	9
3.2.2 Zusatz zu 3. Phase – Beginn der App-Programmierung	10
3.2.3 Zusatz zu Phase 4.....	20
3.2.4 Zusatz zu Phase 5 – Ergebnisse Programmcode.....	21
3.2.5 Zusatz zu Phase 6 - Kriterien zur Bewertung der Apps.....	22
4 PROJEKTEVALUATION.....	24
4.1 SchülerInnenbeobachtungen von HLW und NMS 1	24
4.2 LehrerInnenbeobachtungen.....	25
4.3 Rückmeldung der Funktionstüchtigkeit der Apps	26
5 REFLEXION UND AUSBLICK.....	27
6 LITERATUR	28

ABSTRACT

Die HLW St. Veit ist eine Humanberufliche Schule, die die Schwerpunkte zwar in Informatik setzt, jedoch nur in der Anwenderschulung. Mit diesem Projekt wird eine Einführung in die Programmierung in Apps gegeben, wobei die SchülerInnen grundlegende Konzepte (bedingte-Anweisung, Variablen,...) erlernen sollen. Die Apps, die die SchülerInnen programmieren werden, werden dann den SchülerInnen einer NMS gezeigt. Die Bewertung wird von den SchülerInnen der NMS durchgeführt.

Schulstufe: 11. Schulstufe (3. Klasse HLW) und 6. Schulstufe (2. Klasse NMS)

Fächer: Angewandte Informatik

Kontaktperson: Mag. Cornelia Stadtmann

Kontaktadresse: Dr.-Arthur-Lemisch-Str. 14, 9300 St. Veit

1 MOTIVATION

1.1 Ausgangssituation

Die Höhere Lehranstalt für wirtschaftliche Berufe (HLW) St. Veit an der Glan ist eine humanberufliche Schule mit den Schwerpunktzeigen Umweltökonomie und 3. Lebende Fremdsprache. Die Informatik wird in der 9. und 10. Schulstufe officelastig unterrichtet. In der 11. Schulstufe sind die Schwerpunktthemen Datenbanken und Bildbearbeitung.

Das Projekt an der HLW wird mit der 3C (11. Schulstufe) durchgeführt. In dieser Klasse befinden sich 8 Schülerinnen. Der Unterricht findet zwei Stunden pro Woche statt.

Die Partnerschule der HLW ist die Neue Mittelschule 1 St. Veit an der Glan. Die durchführende Lehrperson dieses Projekts unterrichtet an beiden Schulen und möchte die Kooperation dieser vertiefen.

1.1.1 Schulische Rahmenbedingungen an der HLW

Mit jeweils zwei Wochenstunden ab der 11. Schulstufe wird das Fach „Angewandte Informatik“ unterrichtet. Die Teilungszahlen in diesem Fach sind sehr niedrig. Die Gruppe, die an diesem Projekt teilnimmt, besteht aus 8 Mädchen.

Die Schülerinnen der 3CHW haben keine Programmierkenntnisse. Sie verwenden Apps und haben auch Spaß sie zu kritisieren. Die Lehrperson der zweiten Gruppe dieser Klasse fand das Projekt interessant. Daher wurden den Schülerinnen einige Apps und deren Entstehung gezeigt. Leider konnte man organisatorisch die Gruppen nicht zusammenlegen und auch das Programmieren in der zweiten Gruppe war zeitlich nicht durchführbar.

Die Schule ist mit W-LAN ausgestattet, das bei der App-Programmierung benötigt wird.

1.1.2 Schulische Rahmenbedingungen an der NMS 1

Die Neue Mittelschule 1 (NMS 1) ist zwei Gehminuten von der HLW entfernt. Daher kann man in einer Unterrichtsstunde viel an Kooperationsarbeit leisten. Die SchülerInnen der NMS 1 werden als BewerterInnen für die Apps, die die HLW – Schülerinnen kreieren, fungieren.

Die Klasse besteht aus 25 SchülerInnen, 10 Mädchen und 15 Burschen, und hat den Schwerpunkt Sport. Die SchülerInnen bewerten die Apps nach bestimmten Kriterien und eigener Beobachtung.

1.2 Lehrplan

Der Lehrplan der HLW sieht in der 11. Schulstufe die Themen Datenbanken und Bildbearbeitung, sowie Informationsrecherche vor. Es gibt jedoch keinen Programmierunterricht. Viele der SchülerInnen arbeiten Tag für Tag mit Apps und kritisieren sie.

Die Schwerpunktthemen in diesem Jahrgang sind Grundlage für die Erarbeitung der Apps. Die Bildbearbeitung im speziellen Photoshop wird für die Erstellung des Designs der Apps benötigt.

In der Bildungs- und Lehraufgabe des Lehrplans wird angegeben:

„Kenntnisse aus den Bereichen Publishing und digitale Bildbearbeitung praxisgerecht anwenden können“¹

Eine App wird einerseits nach der Funktionalität aber auch andererseits nach dem Design bewertet. Die Bildbearbeitung findet sich in der App-Programmierung wieder.

¹ Lehrplan

Der Lehrstoff für den 3. Lehrgang (11. Schulstufe) verteilt sich folgendermaßen:

Gebiete	Lehrinhalte
Standardsoftware	- Datenbanken (Grundlagen)
Informationsanalyse	- Informationstheorie - Informationsrecherche und –prüfung - Analyse - Verdichtung der Informationen
Bildbearbeitung	- Einführung in ein Grafikprogramm - Grafikformate - Bildauflösung - Farbtiefe - Nachbearbeitung von digitalem Bildmaterial - Scannen

Die App-Programmierung ist im Lehrplan nicht vorhanden, dennoch kann man die Gebiete der 11. Schulstufe gut in die App-Erstellung einbauen. Dabei lernen die SchülerInnen ein Anwendungsgebiet kennen.

1.3 Projektziel

- Die Schülerinnen der HLW St. Veit sollen in diesem Projekt eine App (Android Basis) programmieren und diese mit SchülerInnen der NMS 1 St. Veit testen.

Der Informatikunterricht im Schultyp der wirtschaftlichen Schulen zielt grundsätzlich auf die Eingabe von Daten in den Computer ab. Die Schülerinnen der HLW sollen grundlegende Konzepte der Programmierung durch die Erstellung einer App erfahren. Diese App soll im weiteren Schritt von SchülerInnen der NMS (potentielle Kunden) getestet werden. Einerseits findet eine Vertiefung im Wissen der Informatik statt und andererseits werden die sozialen Kompetenzen der SchülerInnen gefordert und gefördert.

1.4 Ziele auf SchülerInnenebene

Auf SchülerInnenebene werden folgende Ziele gesetzt:

- Jede Schülerin der HLW wird die grundlegenden Konzepte der Programmierung am Beispiel einer App erlernen.
- Die Schülerinnen der HLW sollen eine App designen, was bedeutet, dass sie Ideen bringen müssen, um das Aussehen der App zu kreieren. (Programmierarbeit im Vorfeld)
- Die SchülerInnen der NMS 1 bewerten die Apps auf ihre Funktionalität und das „Aussehen“ („Würdest du diese App haben wollen?“).
- Es soll geprüft werden, inwieweit die SchülerInnen der NMS 1 durch die App einen Lernerfolg vorweisen können.

Am Beginn werden die Schülerinnen der HLW Datenbanken sowie Datenbankstrukturen, Photoshop also Bildbearbeitung und Grundstrukturen der Programmierung erlernen. Sie sollen in der Lage sein ein App zu programmieren. In weiterer Folge sollen sie die vorhandenen Apps auf Nutzbarkeit und Funktionalität bewerten, indem sie diese den NMS 1 SchülerInnen zum Ausprobieren geben.

Neben dem inhaltlichen Aspekt werden auch soziale Kompetenzen gefordert werden. Ziel ist es, dass jede der Schülerinnen drei bis vier NMS-SchülerInnen zugeteilt bekommt, um die App und das Tablet mit ihnen zu besprechen. Dabei übernehmen sie eine Lehrposition – Lehrerrolle.

1.5 Ziele auf LehrerInnenebene

Auf LehrerInnenebene werden folgende Ziele gesetzt:

- Die Zusammenarbeit der Lehrerin aus der HLW St. Veit, Mag. Cornelia Stadtmann, mit der Kollegin der NMS 1 St. Veit, Beate Hermanik, bewirkt, dass auch andere LehrerInnen bereit sind, Projekte mit LehrerInnen der Kooperationsschule HLW durch zu führen.
- Mag. Cornelia Stadtmann erweitert ihren Wissensstand in Bezug auf neue Technologien (Tablet und Apps) und Programmierkenntnissen.

Der Kooperationspartner der HLW St. Veit ist die NMS1 St. Veit. Die Kooperation verläuft aber teilweise etwas holprig. Daher soll dieses Projekt die Zusammenarbeit der LehrerInnen an HLW und NMS fördern. Die Planung, Vorbereitung und Nachbereitung des Unterrichts soll an beiden Schulen durch die jeweiligen Lehrerinnen stattfinden. Damit soll der weiteren Lehrerbelegschaft der Beweis für gute Zusammenarbeit bei Projekten vorgeführt werden und diese sollen auch angeregt werden, neue Projekte mit der HLW St. Veit zu initiieren bzw. selbst Projekte durchzuführen.

Die Schülerinnen der HLW St. Veit sollen vorab ein App programmiert haben, welche sie mit den NMS-SchülerInnen ausprobieren. Die Vertiefung des Informatikunterrichts an der HLW St. Veit muss erfolgen, wobei sich die Lehrperson besonders gründlich auf die App-Programmierung vorbereiten muss.

1.6 Genderebene

Es werden folgende Ziele gesetzt:

- Die Bewertungskriterien von Burschen und Mädchen bezüglich einer App werden beobachtet um markante Unterschiede herauszufiltern.

Die Schülerinnen aus der HLW sind weiblich. Dort kann keine Gendertrennung erfolgen. Die Schülerinnen an der NMS bestehen aus 10 Mädchen und 15 Burschen. Diese sollen die App auf ihre eigenen Vorlieben bewerten.

1.7 Zeitplan

Der Zeitplan zu Projektbeginn war wie folgt geplant:

Zeit	Einheiten
September – Oktober	Datenbanken (Grundlagen)
November - Dezember	Photoshop (Bildbearbeitung)
Jänner - März	Beginn mit der App-Programmierung
April	Auswertung der Apps
Mai	Datenbanken

Der Zeitplan wurde während der Projektdurchführung geändert, da die Schülerinnen der HLW auch an anderen Projekten teilgenommen haben und dadurch die Unterrichtsstunden ausgefallen sind. Der Zeitplan wurde folgendermaßen geändert:

Zeit	Einheiten
September – Oktober	Datenbanken (Grundlagen)
November - Dezember	Photoshop (Bildbearbeitung)
Jänner - März	Beginn mit der App-Programmierung
April	Apps-Programmierung
Mai	NMS SchülerInnen bewerten die Apps

2 APP-PROGRAMMIERUNG

Die Hard- und Software für die App-Programmierung wurden nach dem Vorwissen der Schülerinnen in Bezug auf die Programmierung ausgewählt. Da keine Programmierkenntnisse vorhanden waren, musste eine Plattform her, die so schnell wie möglich Ergebnisse liefert, um die Schülerinnen zu motivieren. Dafür schien der Lehrperson der App-Inventor von MIT (Massachusetts Institute of Technology) geeignet. Die Software im Web bietet die Entwicklung einer App für Android-Geräte. Der Code entspricht „Puzzle-Teilen“, die einerseits Funktionalitäten darstellen (z.B. das bekannte Wischen auf dem Touchscreen) aber andererseits gibt es einige Strukturen der Programmierung wieder (z. B. globale Variablen, bedingte-Anweisung, ...). Diese ersten Erfahrungen sollten die Schülerinnen an die Konzepte der Programmierung herañführen.

2.1 Voraussetzungen der Hard- und Software

Der App-Inventor ist eine Webapplikation. Der Rechner benötigt dazu folgende Einstellungen:

- Driver des Android Geräts (eventuell)
- Neueste Java Version
- App-Inventor (appinventor.mit.edu). Hier muss eine 90 MB große Datei auf dem Computer installiert sein. Diese Datei befindet sich auf der Homepage appinventor.mit.edu zum kostenlosen Download.
- Google Konto

Da die Apps auf Android Basis erstellt werden, wurden aus dem Projektbudget 8 Tablets (Samsung Galaxy 2) gekauft. Die Tablets gewährleisten das gleiche Betriebssystem für alle Schülerinnen, da die Hersteller der Hardwaregeräte Eigenschaften des Android-Betriebssystems auf ihr Gerät abstimmen.

2.2 Voraussetzungen der Lehrperson

Die Lehrperson (Mag. Cornelia Stadtmann) besuchte im November 2012 ein Seminar „App-Programmierung“ an der CHS-Villach. Dabei erlernte sie die grundlegenden Handfertigkeiten der Software. Die Programmierung wurde während des Studiums der Informatik erlernt und konnte im App-Inventor umgesetzt werden.

2.3 Grundlegende Vorbereitungen für die SchülerInnen

Derzeit sind die SchülerInnen mit der Hardware gut vertraut, da kaum jemand kein Smartphone besitzt. Post in [POST, 2012, S. 25] erklärt den Erfolg so: „Hauptgrund dafür sind die unzähligen Apps, die sich in Sekundenschnelle installieren lassen.“ Es ist dennoch ratsam sich mit den Tablets auseinanderzusetzen und die Schülerinnen einige Apps probieren zu lassen.

Die Software für die Programmierumgebung (siehe 2.1) wurde mit der Lehrerin an den Schulcomputern installiert. Dadurch erfahren die Schülerinnen, welche Einstellungen und Rechte man für eine Installation benötigt.

3 PROJEKTVERLAUF

3.1 App-Programmierung mit den SchülerInnen der HLW

Die Schülerinnen der HLW sollten aus ihren eigenen Ideen eine App kreieren und sie durch die NMS-SchülerInnen bewerten lassen. Die Schwierigkeit war, dass die Schülerinnen der HLW keine Vorkenntnisse in der Programmierung hatten.

Im ersten Schritt wurden die Schülerinnen der HLW über Photoshop und Bildbearbeitung unterrichtet. Dabei erlangten sie neben den Umgang mit der Software Photoshop auch die grundlegenden Konzepte der Bildbearbeitung.

Die App-Programmierung begann mit dem Ausprobieren des Tablets (Samsung Galaxy S2). Die erste Aufgabe war es, einige Apps herunterzuladen und auszuprobieren um mit dem Tablet umgehen zu können. Die Computer in der Schule wurden mit der Lehrerin konfiguriert.

Die Stunden mit den NMS-SchülerInnen konnten im Herbst und im Frühjahr stattfinden. Die SchülerInnen waren auf die Unterrichtseinheiten von den jeweiligen Lehrpersonen vorbereitet worden.

3.2 Durchführung und Beispiele

Das Projekt wurde in folgenden Phasen durchgeführt:

1. Phase: Den Schülerinnen der HLW wurde Photoshop und die Bildbearbeitung erklärt. Sie konnten nach der Einführung Bilder für die Apps korrekt formatieren und in die benötigte Größe umspeichern.
2. Phase: Die Schülerinnen der HLW St. Veit wurden mit den SchülerInnen der NMS 1 zusammengebracht und konnten sich in einer Unterrichtseinheit kennen lernen.
3. Phase: Die Schülerinnen konnten durch die Webapplikation des App-Inventors die Apps kreieren. Damit die Schülerinnen mit dem App-Inventor umgehen können, wurde eine Einführung durch Beispiele gegeben. Die Aufgabenstellungen und deren Lösungen stammen aus dem Tutorial der Homepage, dem Seminar und aus [WOLBER, 2011].
4. Phase: In dieser Phase wurden die Schülerinnen motiviert selbst das Design ihrer Apps zu kreieren.
5. Phase: Mit dem App-Inventor vom MIT wurden die Vorstellungen der Schülerinnen umgesetzt. Bei der Programmierung wurde von der Lehrperson geholfen.
6. Phase: Die Apps wurden den NMS SchülerInnen gezeigt, um sie zu bewerten.

In den folgenden Absätzen werden für das Projekt wichtige Phasen genauer beschrieben.

3.2.1 Zusatz zu 2. Phase

Das Projekt konnte organisatorisch ausschließlich in den Informatikstunden durchgeführt werden. Die Schülerinnen der HLW gingen im Herbst in einer Informatikstunde zu den SchülerInnen der NMS. Diese hatten gerade Unterricht in Leibesübungen. Den Schülerinnen der HLW wurden 4-5 SchülerInnen der NMS zugeteilt. Sie sollten in einem kurzen Gespräch Informationen über das Gegenüber herausfinden. Die NMS-SchülerInnen mussten danach die Schülerin der HLW beschreiben. In den Abbildungen sind die Gespräche und die nachfolgende Austauschrunde zu sehen.



Abb 1 Gespräche von HLW- und NMS-SchülerInnen



Abb 2 Informationsaustausch in der Gesamtgruppe

3.2.2 Zusatz zu 3. Phase – Beginn der App-Programmierung

Die Einführung der Programmierung muss schnell und effektiv sein, da der reguläre Unterrichtsstoff genauso besprochen werden muss. Die Programmierumgebung sollte bald erfasst sein. Aus dem Seminar „App-Programmierung“ im Herbst konnte die Lehrperson die Umgebung von MIT für die Schülerinnen der HLW bereitstellen. Die konkreten Einstellungen, die für diese Webapplikation benötigt werden, sind in 2.3 beschrieben. Die Driver-Software der Tablets konnte am Schulcomputer nicht installiert werden. Daher wurde das Angebot genutzt, die App im Web abzuspeichern, zu generieren und mit einem Barcode herunter zu laden. Die Schülerinnen mussten einen Barcodescanner am Tablet installieren.

Das Betriebssystem des Tablets kontrolliert lizenzierte Apps. Sollte eine App nicht registriert sein, wird ein Fehler ausgegeben und die App nicht installiert. Für unser Vorhaben muss dieser Mechanismus, der unter den Einstellungen jedes Gerätes zu finden ist, ausgeschaltet werden.

Die Seite „appinventor.mit.edu“ wird geöffnet und auf Inventor gedrückt. Danach erfolgt die Anmeldung am System mit dem Google-Account. Man kommt auf die in Abb 3 gezeigte Oberfläche. Diese stellt das Design der App zur Verfügung.

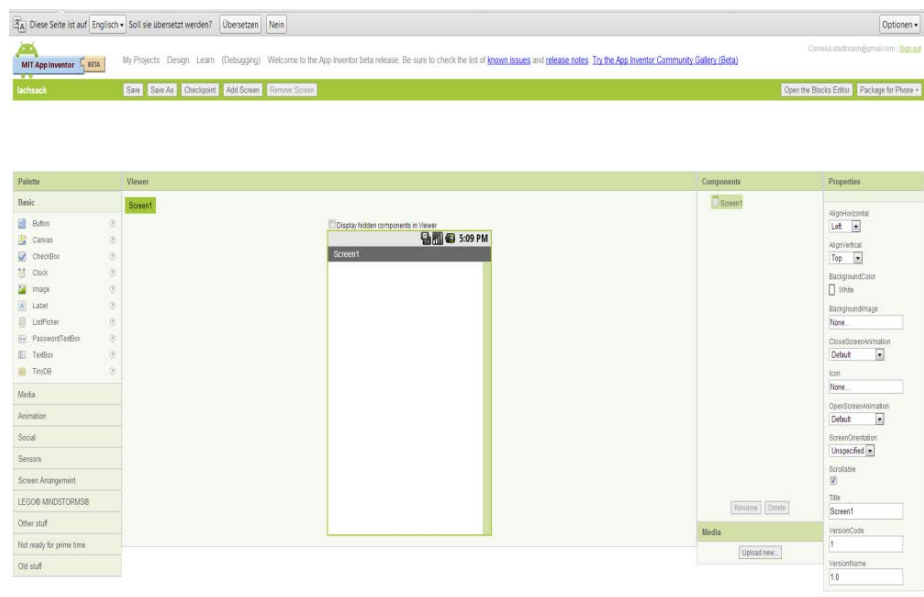


Abb 3 Designseite der App

Aus dem Design-Editor wird der Block-Editor (Abb 4) geöffnet. Der Button befindet sich oben rechts.

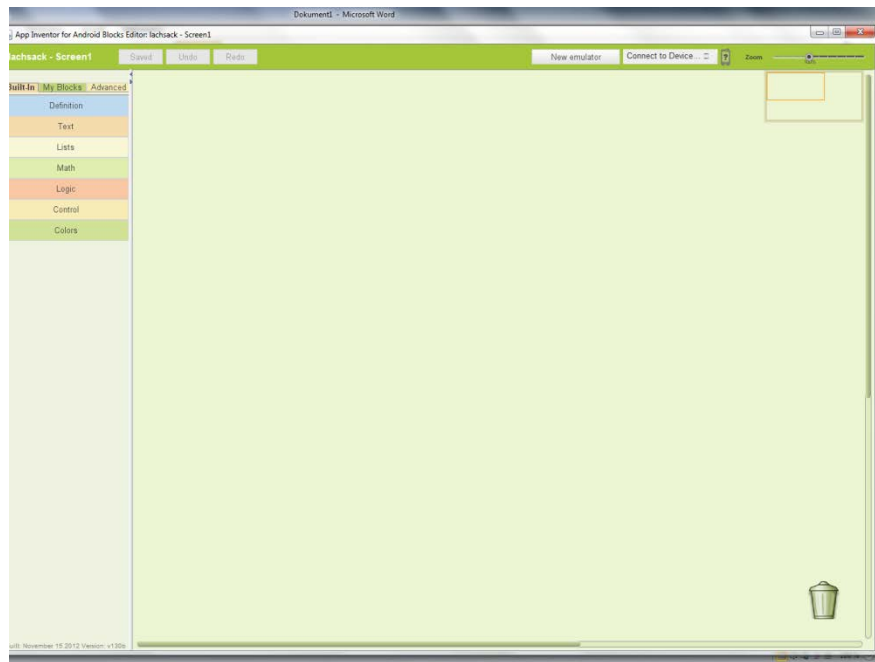


Abb 4 Blocks-Editor (Programmierungsumgebung)

Für die immerwährende Kontrolle der App bietet der Block-Editor ein simuliertes Android-Gerät an. Es heißt „Emulator“ und wird über den Button oben rechts erstellt. Danach muss man diesen noch mit dem Editor verbinden. Dies geschieht wieder über einen Button am oberen rechten Rand.

In Abb 5 sieht man das Öffnen des Emulators und die beiden anderen Editoren.

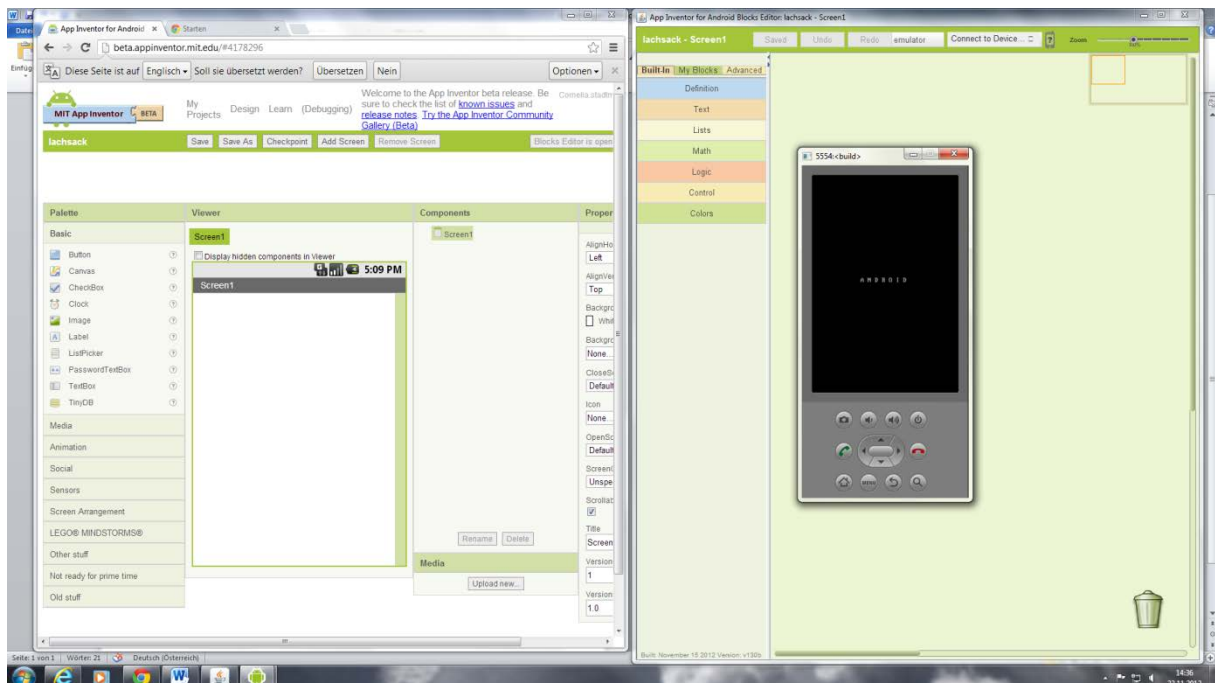


Abb 5 Der Emulator und deren Verbindung

Bevor mit dem Programmieren begonnen wird, sollte eine Idee der App vorhanden sein. Bei den ersten Beispielen ist die Grundstruktur vorgegeben und die SchülerInnen können sich auf die Program-

mierung im Block-Editor konzentrieren. Erst bei der eignen App muss über das Design nachgedacht werden.

3.2.2.1 Die erste App – Lachsack

Die erste App ist ein Lachsack der durch Berührung der Finger beginnt zu lachen. Für das Design werden ein Bild mit einem Sack (vorzugsweise in *.png) und Sound (*.mp3 oder *.mp4) verwendet. Das Design wird im App-Inventor erstellt. Dabei können die Eigenschaften auf der rechten Seite des Editors eingestellt werden. Die restlichen Veränderungen können durch den Block-Editor erfolgen.

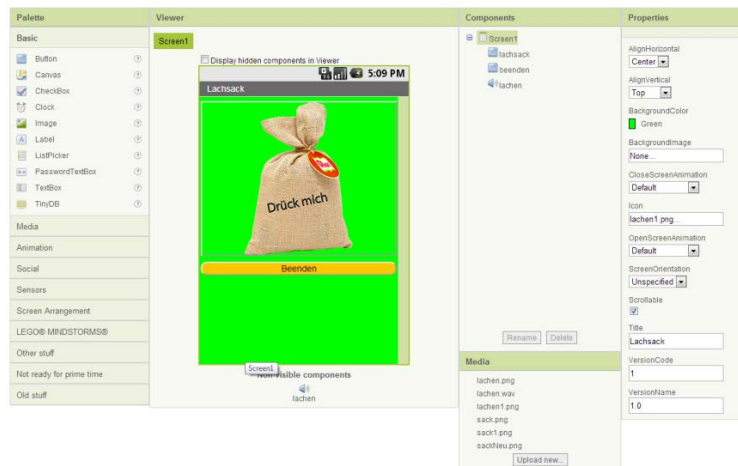
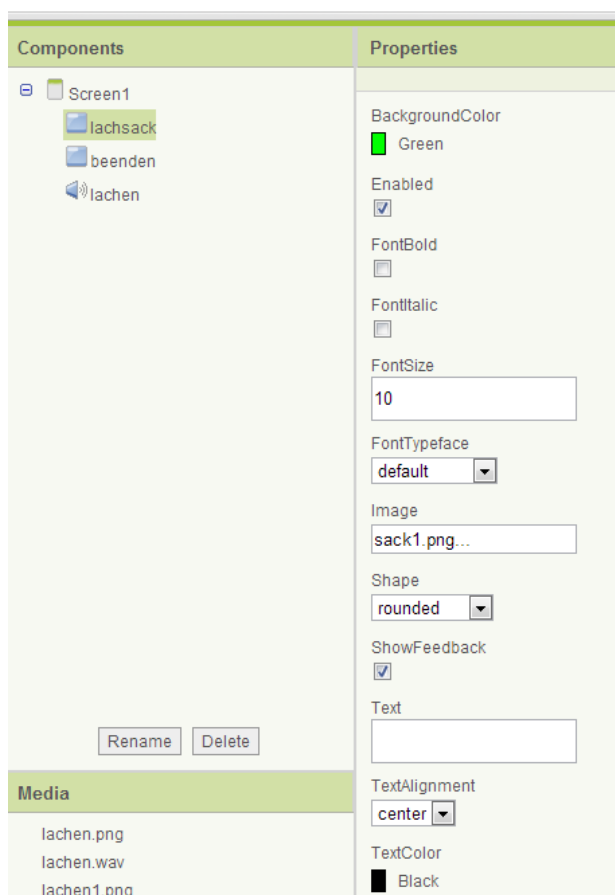


Abb 6 Design des Lachsacks



Auf dieser Seite werden zwei Buttons benötigt. Einer wird als „beenden“-Button konfiguriert und der Zweite beinhaltet als Hintergrund den Lachsack. Der Sound wird zwar upgeloadet, ist am Schirm aber nicht sichtbar. Die Eigenschaften des Buttons mit dem Bild „Lachsack“ werden in Abb 7 angezeigt.



Built: May 6 2013 Version: v134

Abb 7 Einstellungen Bild Lachsack

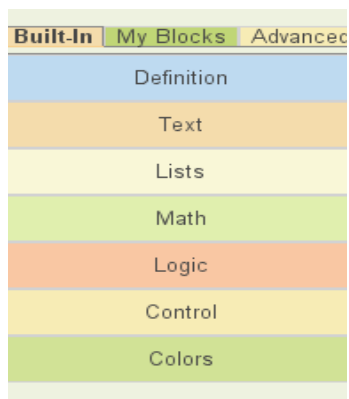


Abb 8 allgemeine Programmierstrukturen

Danach wird der Block-Editor geöffnet und die einzelnen Buttons werden programmiert. Im Editor befinden sich links zwei große Strukturblöcke, die in Abb 8 gezeigt werden.

In dem Block „Built-In“ befinden sich die allgemeinen Programmierteile. D. h. Definition bedeutet, dass man eine Variable (global) definieren kann und damit z. B. eine Laufvariable erschafft. Die Blöcke Text, Math, Logic und Color geben Funktionalitäten der Datentypen an. In Lists können Listen gespeichert werden und mit ihnen gearbeitet werden. Der Block Control gibt Anweisungen und Schleifen an. Hier befindet sich die bedingte-Anweisung bzw. eine While-Schleife.

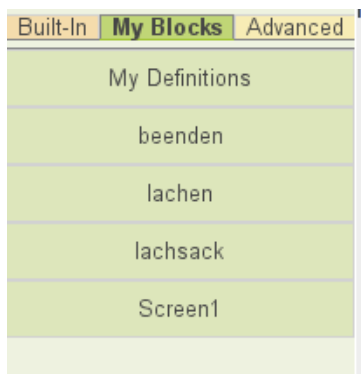


Abb 9 Meine eingefügte Blöcke

In MyBlocks sind jene Elemente vorhanden, die im Design eingebaut worden sind. Beispielsweise gibt es für den Button eine Funktion „button.click“, in der überprüft wird, ob auf den Button gedrückt wird. Diese Anweisung benötigt man nun für die Programmierung. Wenn man auf das Bild (Button) drückt, dann soll ein Lachen zu hören sein. Die dazugehörige Puzzlestruktur sieht wie in Abb 10 aus. Wenn man auf den Button „beenden“ drückt, dann soll sich die Applikation schließen (siehe Abb 10).

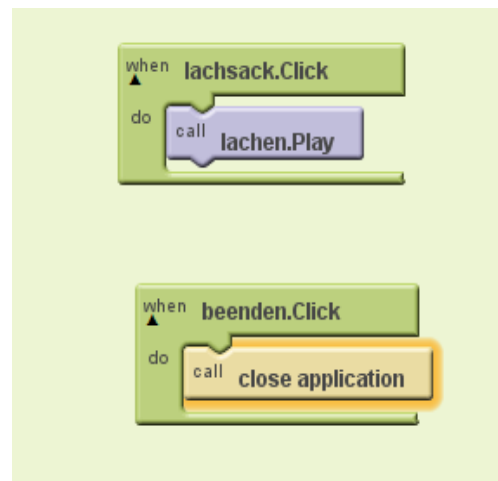


Abb 10 Codefragment des Lachsacks

Nach diesen Schritten soll das App ausprobiert werden. Da die Driver-Software nicht auf dem Computern installiert wurde, wird der Barcode-Generator verwendet. Dazu befindet sich im App-Inventor im oberen rechten Eck eine Schaltfläche in der ein Barcode generiert wird. Mit dem Barcodeapp auf den Tablets wird das App aus dem Internet heruntergeladen. Das produzierte App kann nun ausprobiert werden.

Der Lachsack wurde in weiterer Folge noch um ein Feature verbessert. Der Button in dem der Lachsack abgebildet wurde, wurde durch ein „Canvas“ (eine Zeichenfläche) ersetzt. Das Element „Canvas“ enthält die bekannten „Wischfähigkeiten“, die für ein Smartphone typisch sind. In der Zeichenfläche wird das Bild „Lachsack“ als Hintergrundbild eingefügt. Danach kann die Funktion des „Wischens“ im Block Editor genutzt werden. Der Programmcode sieht folgend aus (Abb 11):

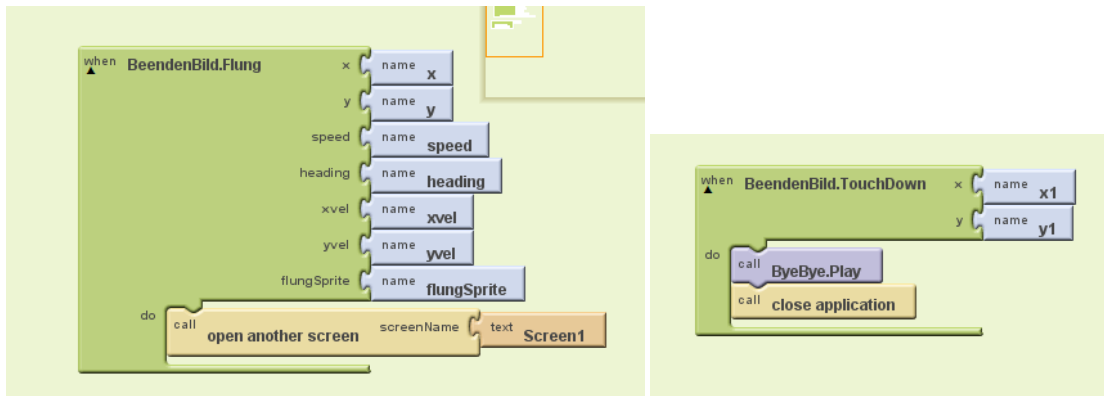


Abb 11 Codefragmente vom Lachsack 2

Das „Canvas“ bietet die Funktion „*.Flung“ an, die das Wischen wiedergibt. Die Parameter geben die Position des Fingers und die neue Position der Finger wieder. Wenn über die Fläche gewischt wird, dann öffnet sich ein neuer Schirm (Screen 1). Der BeendenButton mit der Funktion „*.TouchDown“ führt zur Beendigung der Applikation mit einem kleinen musikalischen Abgang.

Nach diesem Beispiel kannten die Schülerinnen die Grundfunktionalitäten des App-Inventors. Damit die Schülerinnen für ihre Apps auch andere Ideen bekommen und die Puzzle-Teile näher kennen lernen, wurden weitere Beispiele mit der Lehrerin durchgenommen.

Die angeführten Beispiele werden kurz beschrieben und der Code dargelegt. Die fertigen Apps können im Buch [WOLBER, 2011] oder auf der Homepage nachgeschlagen werden:

3.2.2.2 Tom und Jerry – Wer fängt wen?

Dieses Beispiel sollen die Schülerinnen selbst lösen, um die Verwendung des App-Inventors abzuprüfen. Die Aufgabenstellung sieht wie folgt aus:

Erstelle eine kleine App mit folgenden Features:
 2 Screens: Diese wechseln durch Wischen am Schirm hin und her. Bei Bedarf kann man auch Sounds einbauen. Der Programm-Icon soll verwendet werden.
Screen 1: Bild von Tom, Button zum beenden der App
Screen 2: Bild von Jerry, Button zum beenden der App

Die dazugehörenden Lösungen:



Abb 12 Der Screen der Katze



Abb 13 Der Screen der Maus

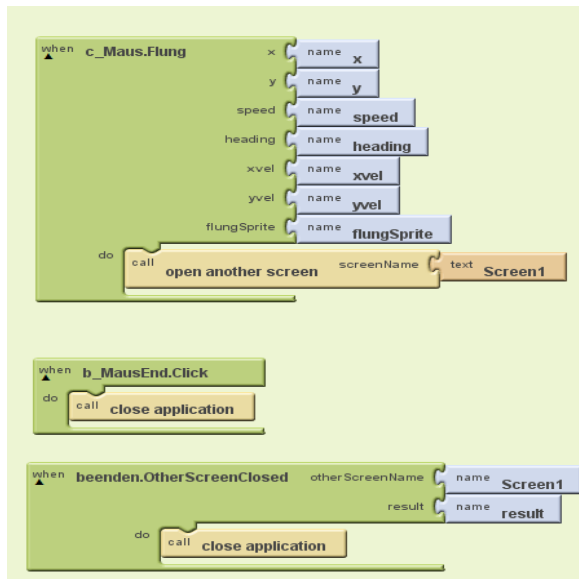


Abb 14 Programmcode Katze

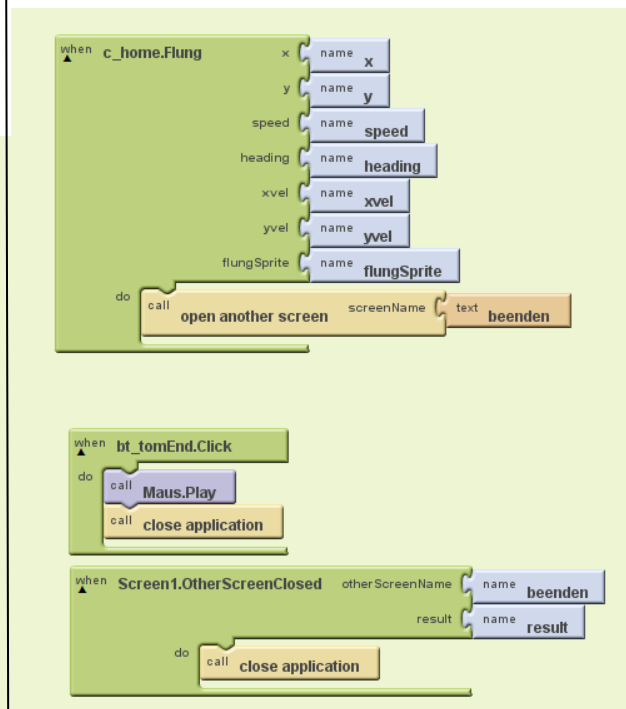


Abb 15 Programmcode Maus

3.2.2.3 Malen am Bildschirm

Dieses Beispiel befindet sich komplexer in [WOLBER, 2012]. In diesem Beispiel werden folgende Komponenten eingebaut:

- Screen Arrangement (zum Positionieren der Elemente)
- Buttons (für Aktionen)
- Canvas (als Zeichenebene)
- Camera (um Bilder anzumalen)

Die Abfolge für das Design und die Programmierung verläuft in Reihenfolge der aufgezählten Elemente. Die einzelnen Elemente werden nach ihrer Funktionalität benannt. Die Aufgabenstellung lautet wie folgt:

Man kann Punkte zeichnen, die entweder rot, grün oder blau sind. Am Beginn der Anwendung ist die Farbe pink eingestellt. Der Radius der Kreise (Punkte) beträgt 5 Pixel.

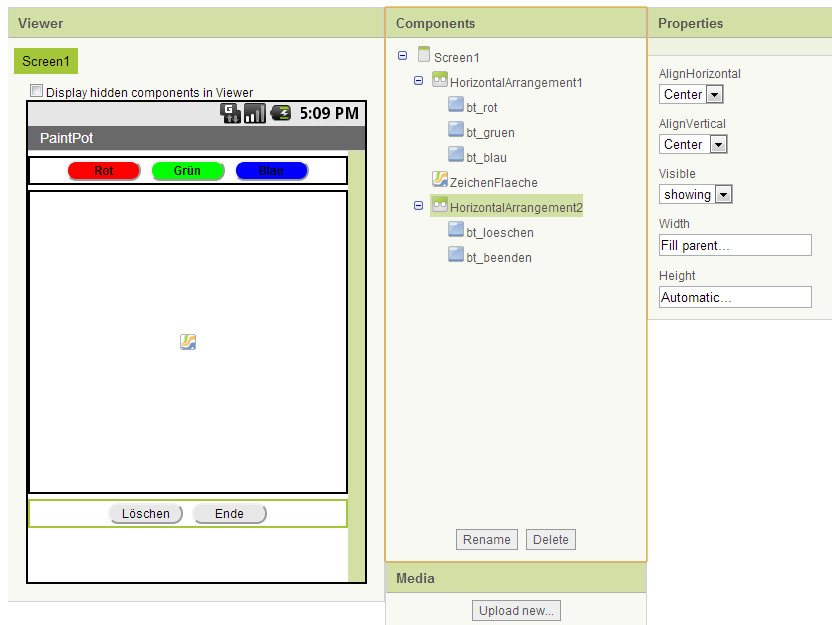


Abb 16 Das Layout der Malfläche

Die Programmierung wird in Abb 17 dargestellt:

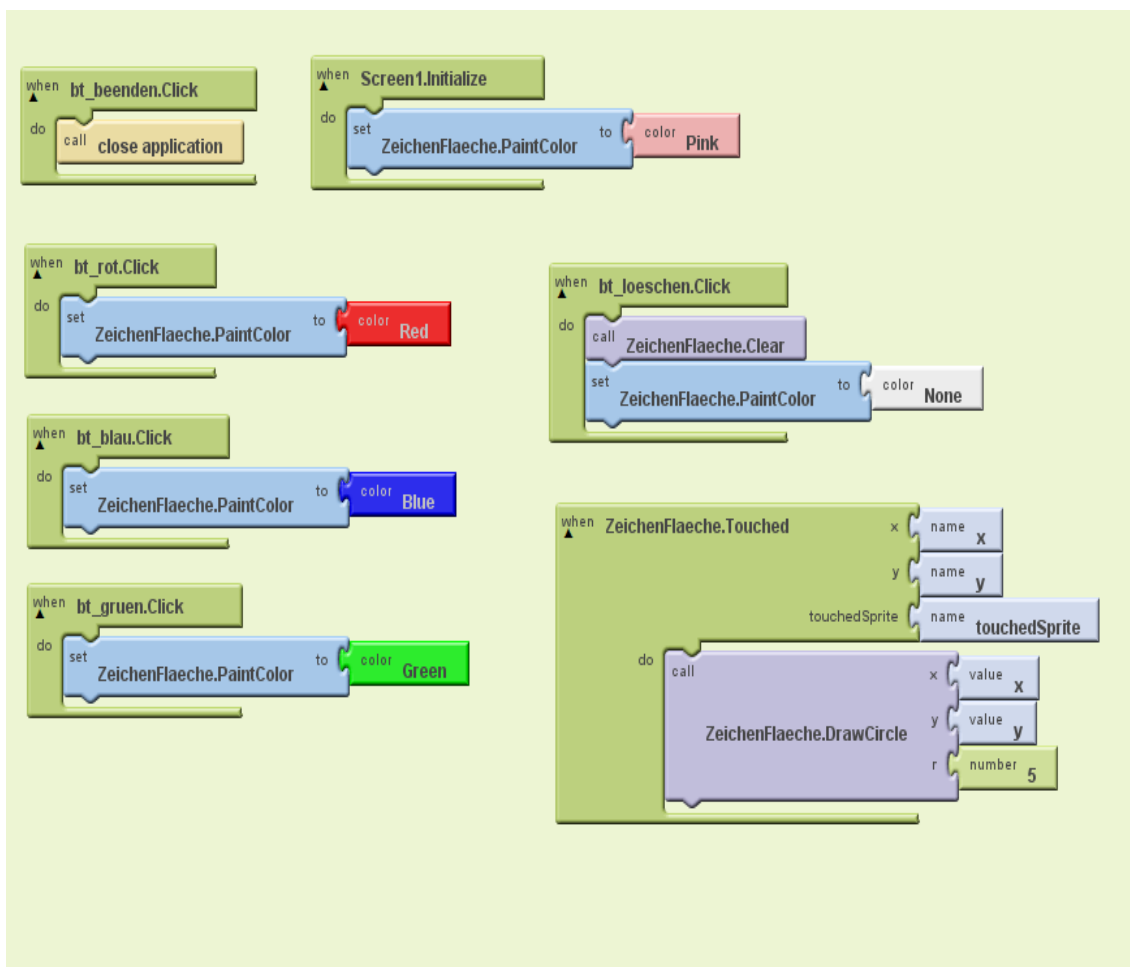


Abb 17 Programmcode Punkte malen

3.2.2.4 Ball spielen

In diesem Beispiel sollen die Schülerinnen eine App programmieren, die Punkte zählt. Hier kommt es zur Definition einer globalen Variable. Die Aufgabenstellung lautet:

Es sollen zwei Bälle vorhanden sein, wobei ein Ball den anderen fängt. Triff dieser den anderen Ball so bekommt man Punkte. Nach einigen Punkten kommt ein zweiter Ball hinzu, der nicht getroffen werden darf, da er sonst die Punkte wieder abzieht.

Es werden folgende Komponenten benötigt:

- Animationen (Ball)
- Canvas ist die Spielfläche
- Wir spielen nur den Sound, wenn man auf Laut geschaltet hat, ansonsten wird die Vibration eingeschalten

Das Design dieses Spiel:

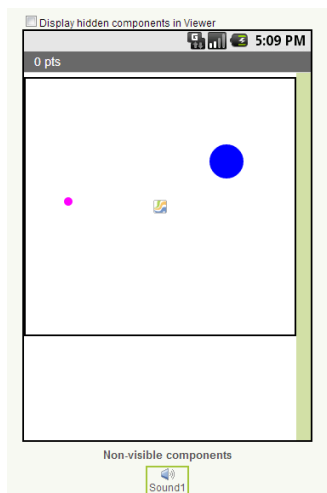


Abb 18 Design Ball spielen

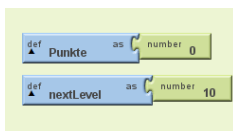


Abb 19 globale Variablen

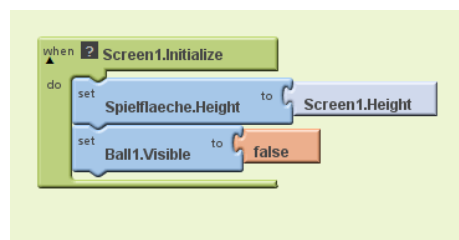


Abb 20 Initialisieren des Bildschirms

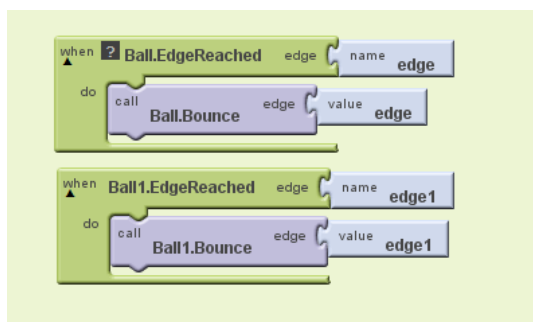


Abb 21 Bälle prallen von Wand ab

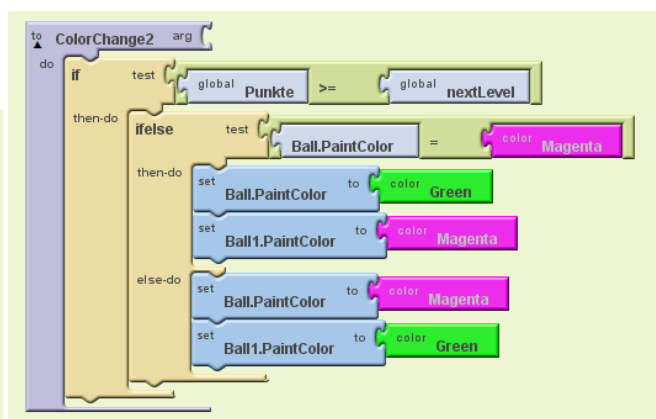


Abb 22 eigene Methode für das Farben wechseln

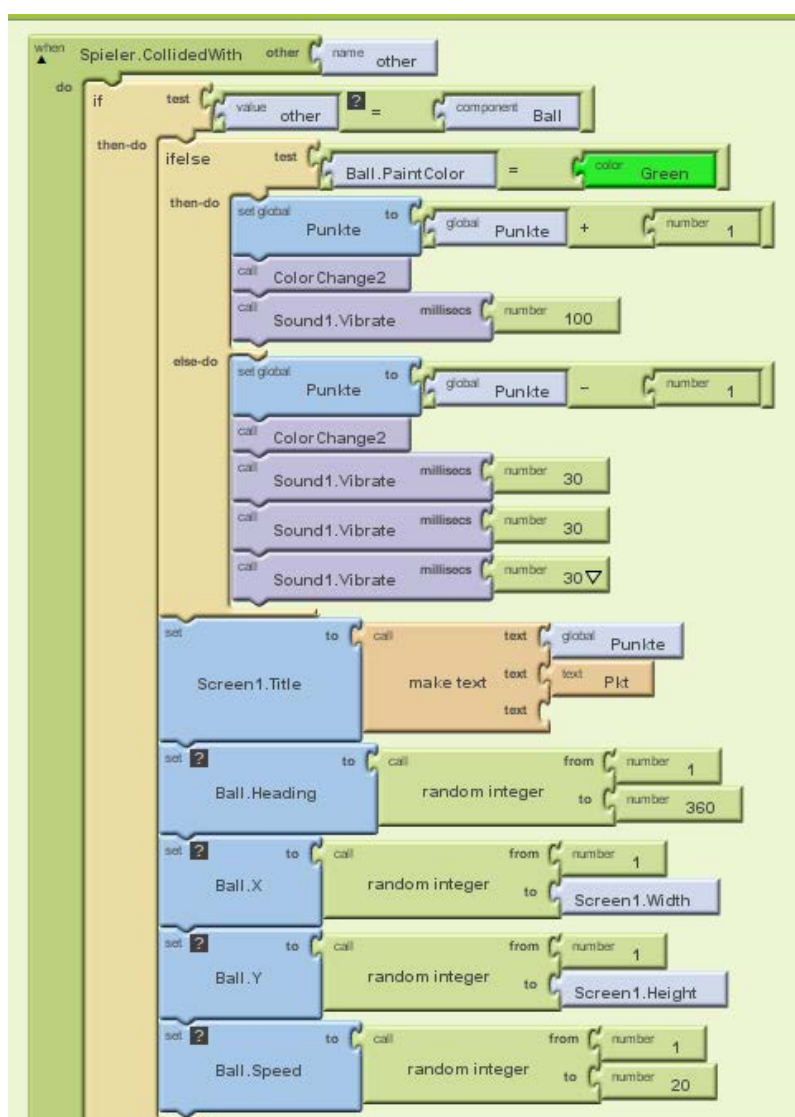


Abb 23 Bedingte-Anweisung 1. Teil

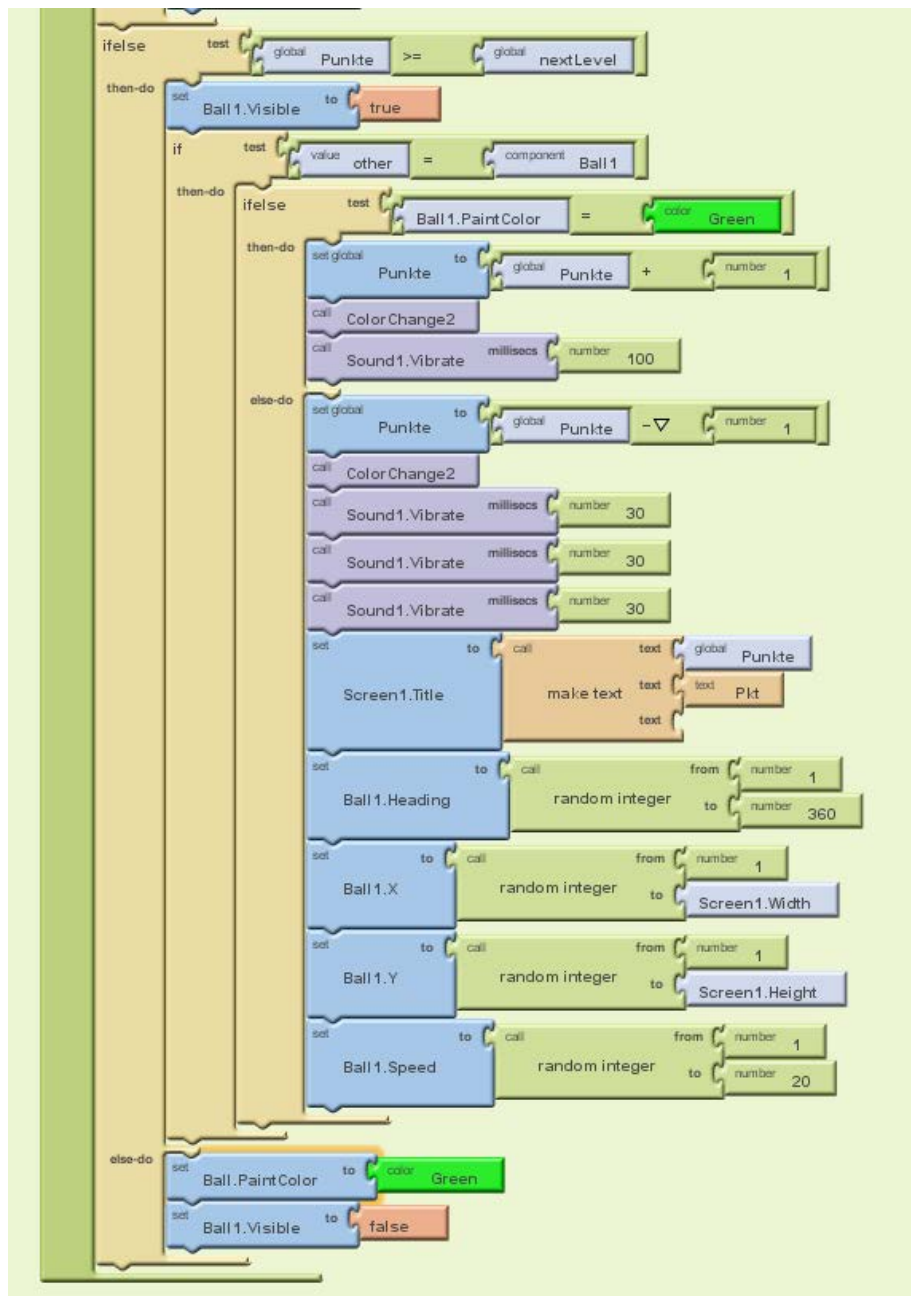


Abb 24 Bedingte-Anweisung 2. Teil

3.2.2.5 Magic Ball

Die Aufgabenstellung und der Code ist von der MIT-Seite herunter zu laden (<http://appinventor.mit.edu/explore/tutorials.html>).

3.2.3 Zusatz zu Phase 4

Nach der Einführung durch einige Beispiele im App-Inventor begann die Arbeit an den eigenen Apps. Zuerst mussten Ideen her, die mit der Lehrperson besprochen wurden.

Es konnten folgende Apps kreiert werden: (Kurzbeschreibungen von den Schülerinnen)

Schülerin	Idee
1	Satzspiel für Kinder in Englisch: Zuerst wird eine Figur ausgewählt, dann kommt man zum Spielfeld. Es sind verschiedene englische Wörter in Blöcken am Bildschirm. Man muss die richtigen Wörter anklicken, damit es einen Satz ergibt. Tippt man falsch wird das Feld rot, ist es richtig wird es grün. Am Ende erhält man eine Krone.
2	Ein Spiel über gesundes Essen, der Smiley wird traurig wenn man ihn mit Ungesundem füttert und glücklich wenn man ihn mit Gesundem füttert.
3	Hau den Maulwurf Der Maulwurf kommt auf den Bildschirm und wird vom Spieler durch antippen erwischt. Dafür erhält er Punkte. Damit es schwerer wird, kommt nicht nur ein Maulwurf sondern mehrere und es können auch Mäuse auftauchen. Wenn eine Maus angetippt wird, dann bekommt man negative Punkte
4	In meinem App geht es darum, die Englisch-Kenntnisse zu verbessern. Es scheint ein Bild auf mit 4 verschiedenen Sätzen darunter und nur ein Satz ist der Richtige um zum nächsten Bild weiter zu springen.
5	Sprachenapp auf Spanisch: Vier Bilder beschreiben ein spanisches Wort und der Benutzer muss das Wort eintippen. Ist es falsch, dann verschwindet sein Wort. Ist es richtig, dann kommt er zum nächsten Level.
6	Ich habe eine Rechenapp programmiert, in der Kinder ihre Kenntnisse in Mathematik mit Kopfrechnen verbessern können. Für jede richtige Rechnung bekommt man Punkte und eine neue Rechnung. Hat man das Ergebnis falsch, werden Punkte abgezogen und man kommt nicht ins nächste Level und somit auch keine neue Aufgabe.
7	Ich habe eine App programmiert, bei der man englische Wörter und Sätze mit den dazugehörigen deutschen Wörtern und Sätzen verbinden muss. Ist die Aufgabe richtig gelöst, erscheint ein grünes Feld, ist es falsch ein Rotes.
8	Englisch App zum üben Es werden Bilder gezeigt und der Benutzer muss das richtige Wort anklicken. Ist es falsch wird es auf rot gesetzt, ist es richtig wird es grün und man kommt auf einen neuen Screen.

3.2.4 Zusatz zu Phase 5 – Ergebnisse Programmcode

Exemplarisch werden Programmcodes und Layouts der Schülerinnen der HLW gezeigt.

Programmcode der Schülerin 8:

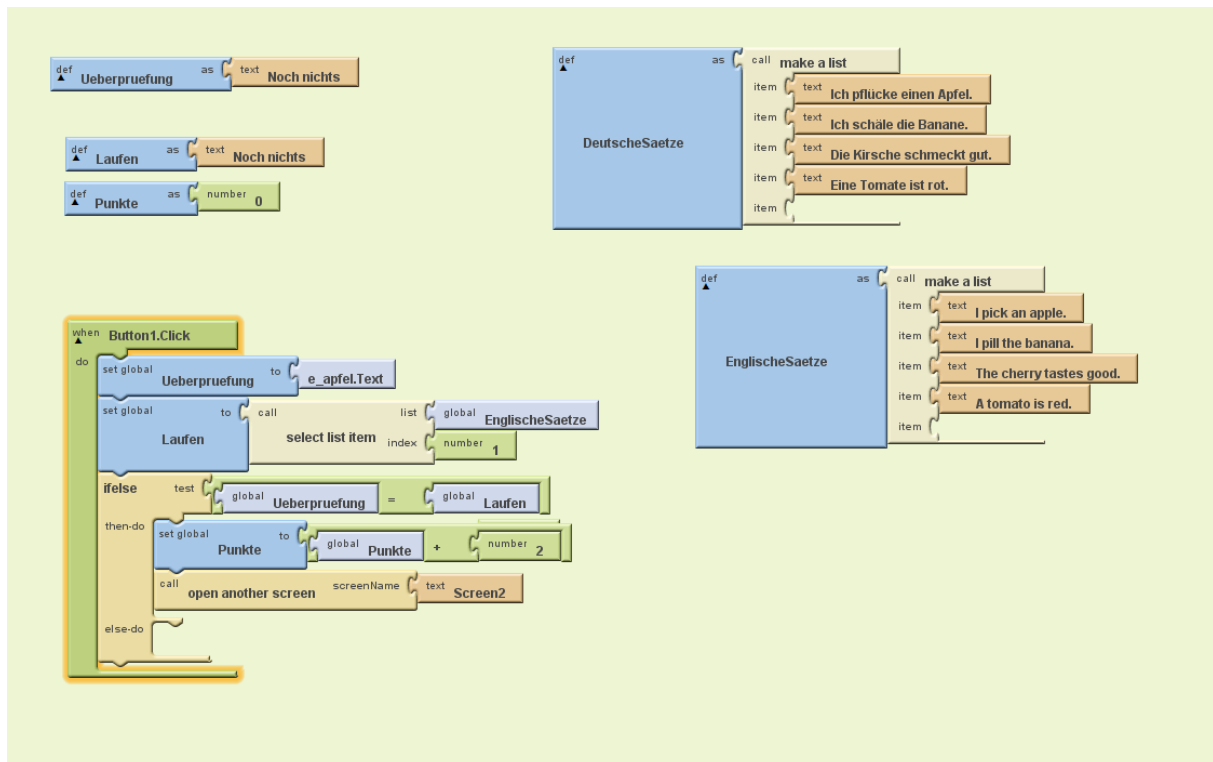


Abb 25 Programmcode Englische Bilder zuordnen

Programmcode der Schülerin 4:

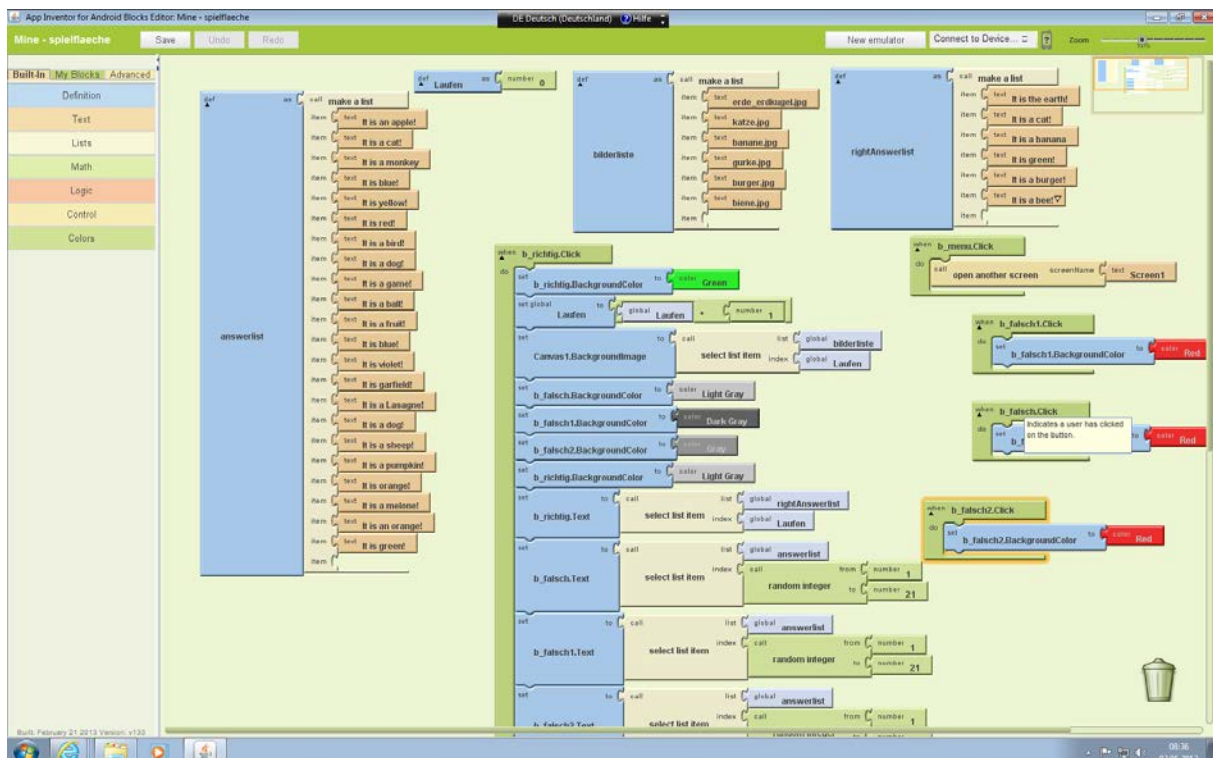


Abb 26 Lernapp

Layout der Schülerin 3:



Abb 27 Layout der Schülerin 3

Programmcode der Schülerin 6:

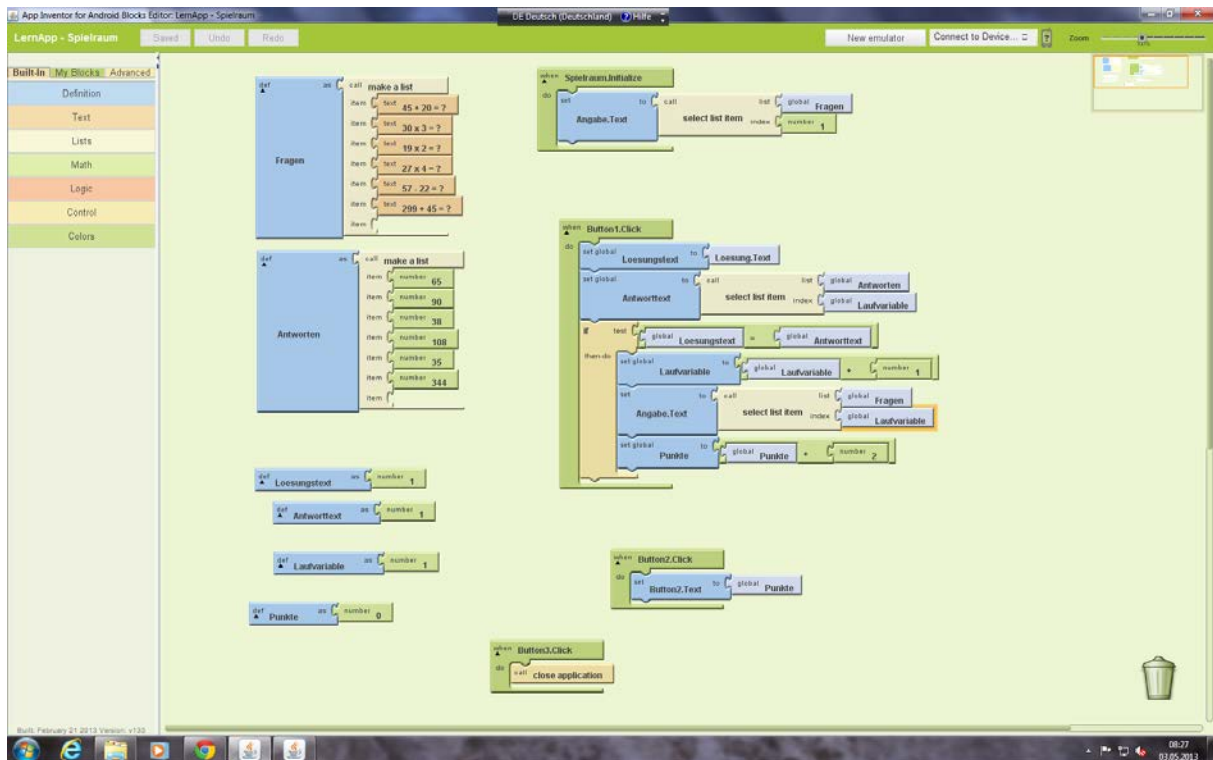


Abb 28 Programmcode Matheprogramm

3.2.5 Zusatz zu Phase 6 - Kriterien zur Bewertung der Apps

Die SchülerInnen der NMS sollten die Apps bewerten. Dazu wurde von den HLW-Schülerinnen ein Feedbackblatt ausgearbeitet. Mit diesem Blatt und den Apps wurde am Freitag, den 16.05.2013 die NMS von den HLW-Schülerinnen besucht. Die Lehrerinnen hatten die SchülerInnen beider Schulen im Vorfeld vorbereitet.

Die Feedbackfragen:

1.) Welche App hast du dir angeschaut oder welche Programmiererin steht hinter der App?

Name der App _____

2.) Gib den jeweiligen Fragen eine Bewertung zwischen 1 und 10 Punkten.

- Wie hat dir die Grafik/ das Layout gefallen? _____
- Wie hat dir die Aufgabenstellung gefallen? _____
- Hast du sofort gewusst, was für einen Sinn das App hat? _____
- Wie schwer war die App für dich? _____
- Hat dir die App Spaß gemacht? _____
- Kann man durch die App etwas lernen? _____

3.) Gib hier deine Meinung über die App an!

4 PROJEKTEVALUATION

4.1 SchülerInnenbeobachtungen von HLW und NMS 1

Aus dem ersten Kennenlernen der SchülerInnen konnte man von den HLW Schülerinnen folgendes Feedback erfahren (schriftlich):

- „Die Kinder waren sehr aufgeweckt und ich kann mir vorstellen, dass ihnen beim Arbeiten schnell die Konzentration ausgeht. In der kleinen Gruppe ist es mir mit den Kindern leichter gefallen, da konnte ich sagen, wer reden soll, da haben sie nicht durcheinander geredet.“ Schülerin 1
- „Ein paar waren wirklich frech und konnten nie leise sein. Die anderen waren jedoch mit Begeisterung dabei und machten sofort mit.“ Schülerin 4
- „Die Schüler waren relativ laut und aufgedreht. Sie waren auch etwas frech.“ Schülerin 4
- „Am Anfang habe ich sehr stark gemerkt, dass diese Kinder wirklich hyperaktiv sind. Nichts Neues für eine Sportklasse. Ich dachte mir, dass das Arbeiten mit den Schülern vielleicht etwas anstrengend werden könnte. Im Nachhinein bin ich aber draufgekommen, dass unsere „Versuchskaninchen“ total freundlich und lieb sind. Wir konnten nette Gespräche führen und sie waren generell an dem Ganzen interessiert.“ Schülerin 7

Die SchülerInnen der NMS waren sehr begeistert von den Schülerinnen der HLW. Sie fanden sie nett.

In Bezug auf die Apps-Programmierung wurden folgende Ergebnisse erlangt:

- 1.) Was hat dir beim App-Programmieren am meisten gefallen?

Die Aussagen der Schülerinnen der HLW waren:

- „Am meisten hat mir das Ausprobieren der App gefallen.
- Das man der eigenen Kreativität freien Lauf lassen konnte. Es hat wirklich Spaß gemacht im APP Inventor ein bisschen zu tüfteln. Interessant fand ich außerdem, dass unsere Apps zu den anderen die man frei im APP Store herunterladen so klein wie eine Mücke sind und wir hatten teilweise mit denen schon Schwierigkeiten.
- Mir hat es gut gefallen, dass ich mir die einzelnen Bilder mit den Sätzen darunter selbst aussuchen konnte und konnte so zu sagen meine kreative Ader fließen lassen.
- Mir hat am meisten gefallen, etwas selbstständig zu erarbeiten und es dann auch umsetzen zu dürfen, obwohl es schwierig war.“

- 2.) Was hat dir beim App-Programmieren überhaupt nicht gefallen?

Die Aussagen der Schülerinnen der HLW waren:

- „Dass sich einige Fehler nicht beheben ließen (das Schließen der App)
- Die ewig langen Bearbeitungen, bis alles in Ordnung war, sind anstrengend und nervig gewesen.
- Es war anstrengend die einzelnen Lösungen, sowohl falsch als auch richtig einzutippen.
- Im Endeffekt hat mir das Projekt recht gut gefallen. Das Einzige, das mich ein bisschen gestört hat, war das ewige Zusammensetzen der Puzzleteile im APP-Inventor.“

3.) Welche Schwierigkeiten hattest du beim Programmieren?

Die Aussagen der Schülerinnen waren:

- Das Schließen der App war nicht möglich sondern man kam immer wieder zurück zum traurigen Smiley.
- Das Erstellen der App war ganz einfach, nur das Zusammenstellen der einzelnen Aufgabenschritte im Block Editor ist sehr kompliziert gewesen, denn wenn man eine englische Frage mit der Deutschen beantworten will, musste ich zuerst alle anderen englischen Möglichkeiten ausblenden.
- So einige. Die Buttons haben nicht ausgesehen wie ich wollte, sie waren an der falschen Stelle, die Maulwürfe konnte ich nicht so einfügen wie ich wollte, sie tauchten nicht auf und verschwanden nicht wann sie sollten, den Pause Button habe ich nicht geschafft zu machen und es wurde mir auch nicht erklärt wie das geht, zur Punkteverteilung bin ich gar nicht erst gekommen und nichts hat so funktioniert wie es sollte.
- Wenn ich Schwierigkeiten hatte, hat mir Frau Prof. geholfen und das Problem mit mir gemeinsam gelöst. Bei Kleinigkeiten habe ich sofort eine Erklärung bekommen.
- Es war relativ schwierig, die einzelnen Buttons mit den Sätzen so zu programmieren, dass der richtige Satz grün wird und sofort zur nächsten Ebene weiter hüpf.

4.2 LehrerInnenbeobachtungen

Programmierung App:

Die Schülerinnen der HLW St. Veit hatten keine Programmierkenntnisse. Sie verwenden zwar Apps, kennen aber die Konzepte und Programmierprogramme, die dahinter stehen, nicht. Für diese Voraussetzung konnten die Schülerinnen ein gutes App programmieren. Das Konzept der bedingte-Anweisung war ihnen durch Excel bekannt. Sie konnten dies auch gleich umsetzen.

Manche der Schülerinnen benötigten keine Hilfe der Lehrperson und hatten am Ende ein funktionierendes App. Bei anderen musste die Lehrperson eingreifen. Sie konnten jedoch nach einer bestimmten Zeit eine App selbst fertig stellen. Eine Schülerin konnte sich nicht vom Design wegreißen und wollte dieses perfekt haben. Leider funktionierte ihre App dann nicht, da das Layout zu überlastet war. Die App konnte nicht angezeigt werden. Sie wollte auch am Ende nicht verstehen, dass zu viel an Grafikobjekten das System überladen hat.

Apps probieren in der NMS:

Die SchülerInnen der NMS freuten sich sehr über das Ausprobieren der Apps. Es war für sie motivierend die Schülerinnen der HLW zu sehen. Die Apps wurden fleißig durchbesprochen und durchgespielt. Es war ein aufregendes Tun in dieser Stunde. In den nächsten Abbildungen sind einige Situationen dargestellt.





4.3 Rückmeldung der Funktionstüchtigkeit der Apps

Die Auswertung der Feedbackbögen ergab folgendes Ergebnis. Es ist anzumerken, dass die Namen der Apps von den SchülerInnen der NMS erfunden wurden.

Schülerin	AppsNamen der SchülerInnen der NMS	Bewertung und Zusatzinformationen (60 Punkte)
1	„Spielgrund“	Durchschnittswert: 27 Pkt. Die Kinder fordern dabei mehr an Aufgaben und diese sollten schwieriger und länger sein. Die SchülerInnen der HLW waren recht zufrieden, aber für ihre Erfahrungen an Apps-Anwendungen zu wenig.
2	APPAPP	Durchschnittswert: . 45,5 Pkt. Die SchülerInnen der NMS fordern eine längere App, aber ansonsten war die App lustig.
3	Magic Ball (Die andere App funktionierte noch nicht)	Durchschnittswert: 44,4 Pkt. Die App empfanden die Kinder als lustig. Die Grafik war toll. Sie gaben auch dieser App die Note 1.
4	Übersetzerapp	Durchschnittswert: 53,6 Pkt. Es ist zwar ausbaufähig, aber man kann English gut erlernen. Es ist lustig und cool. Die SchülerInnen bemängelten nur die Schwierigkeit. Es sei für sie der Stoff der Volksschule.
5	4 Bilder/ 1 Wort	Durchschnittswert: 49 Für die SchülerInnen ist es noch ausbaufähig und das man es in mehrere Sprachen ausführen könnte. Ein Schüler schrieb sogar: „Hut ab vor der Macherin!“
6	Mathe-Quiz	Durchschnittswert: 39,6 Die App hat gefallen. Es hat sehr viel Spaß gemacht und das Gerät ist toll!
7	Spielland	Durchschnittswert: 36,2 Folgende Liste wurde von den SchülerInnen erstellt: <ul style="list-style-type: none"> • mehr Aufgaben • gut gestaltet • sehr kurzes Spiel • Es ist gut, wenn man English lernen möchte.

8	English App	Durchschnittswert: 59 Es könnten noch mehr Fragen sein, ansonsten ist die App klasse!
---	-------------	--

Die nachfolgende Abbildung zeigt die Bewertungen in einer Grafik. Dabei stellt man fest, dass die English App der Schülerin 8 am besten bei den SchülerInnen der NMS bewertet wurden. Die Schülerin 1 konnte die SchülerInnen der NMS nicht mit ihrer App begeistern.

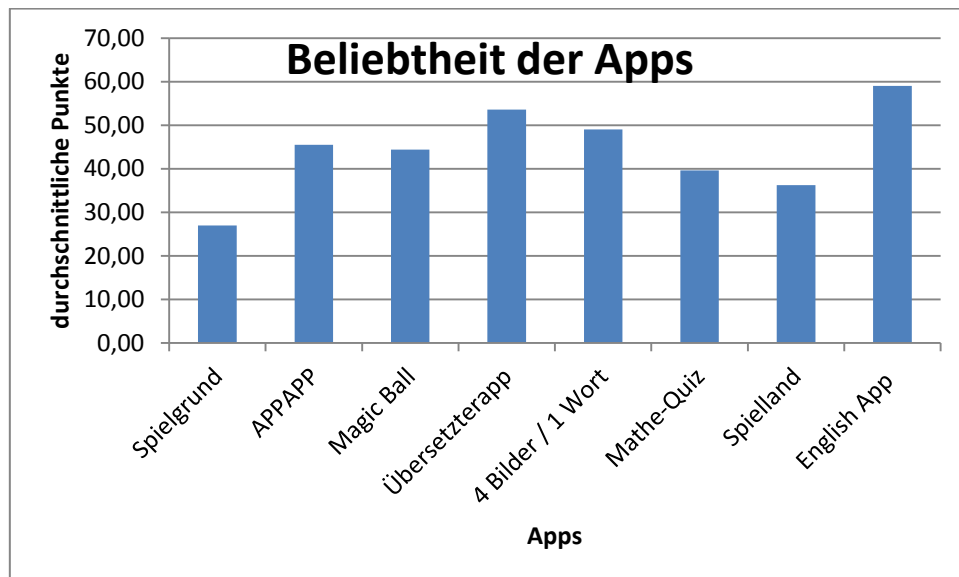


Abb 29 Beliebtheit der Apps

5 REFLEXION UND AUSBLICK

Die App-Programmierung an der HLW St. Veit war für die Lehrperson eine kleine Herausforderung, da die SchülerInnen in diesem Schultyp mit Programmierung nicht in Verbindung kommen. Das Projekt wurde jedoch mit Erfolg abgeschlossen.

Die Schülerinnen der HLW St. Veit hatten vorab keine Programmierkenntnisse und haben sich ohne Angst auf dieses Projekt gestürzt. Es war zeitweise sehr anstrengend für sie, jedoch haben sie es mit funktionierenden Apps beendet. Diese sind zwar nicht groß und ähneln den konventionellen Apps, aber es ist ihre Kreation. Die Reaktionen der SchülerInnen aus der NMS gaben ihnen zuletzt ein gutes Gefühl etwas Tolles erschaffen zu haben. Sie wissen jedoch auf der anderen Seite, dass es nur ein kleiner Teil war und haben nun Respekt vor den Apps in den Stores.

Eine weitere Erfahrung war das Zusammentreffen der Jugendlichen mit den Kindern. Die HLW-Schülerinnen konnten beinahe nicht fassen, dass auch sie in dieser Phase des Lebens gesteckt hatten. Die SchülerInnen der NMS waren so beeindruckt von den HLW Schülerinnen. Die Arbeit an den beiden Terminen war erfolgreich für die Kooperation der beiden Schulen. Es wird für das nächste Jahr ein Projekt angedacht.

Die Ziele auf SchülerInnen-Ebene wurden überwiegend erfüllt. Bei den Programmierkenntnissen kann man nicht sagen, inwieweit sie Variablen, Methoden usw. wissen und verstehen. Dennoch kann man behaupten, dass die bedingte-Anweisung für die Schülerinnen plausibel ist und gekonnt wird.

Die Programmierung wird in Zukunft nicht aus meinem Unterricht weg zu denken sein.

6 LITERATUR

Buch:

Bach, Mike (2012). *Apps entwickeln für Android 4. Das umfassende Training*. Galileo-Computing

Felker, Donn (2011). *Android Apps Entwicklung für Dummies*. Weinheim: Wiley-vch

Post, Uwe (2012). *Andorid-Apps entwickeln. Für Programmiereinsteiger geeignet*. Bonn: Galileo Press

Wolber, David, Abelson, Hal, Spertus, Ellen, Looney, Liz (2011). *App Inventor. Create your own Android Apps*. Canada: O'Reilly

Internet:

APP-Inventor: <http://appinventor.mit.edu/> [06.05.2013]

LEHRPLAN IOM/AINF Humanberufliche Schulen: <http://www.abc.berufsbildendeschulen.at> [29. 1. 2012].