



MathMind



FH Hagenberg - Master in Kommunikation, Wissen, Medien
User Experience & Interaction Design

Wintersemester 2015/16

Christian Hepp
David Kaisalgruber
Markus Raab

Inhaltsverzeichnis

- [1. Sifteo Cubes](#)
- [2. Spielidee ColorMind](#)
- [3. MathMind](#)
 - [3.1 Ideation: Spielidee und Spielbeschreibung](#)
 - [3.2 Interaktion](#)
 - [3.3 Experience](#)
- [4. Tools & Methoden](#)
- [5. Technische Vorbereitungen](#)
 - [5.1 Download Software](#)
 - [5.2 Inbetriebnahme](#)
 - [5.3 Erstellen & Starten einer App](#)
 - [5.4 Weitere wichtige Inhalte der SDK](#)
 - [5.5 Weitere wertvolle Links zur Installation / Ausführung](#)
- [6. Basiskenntnisse & Recherche](#)
- [7. Herausforderungen](#)
 - [7.1 Software](#)
 - [7.2 Hardware](#)
- [8. Implementierung / Ergebnis](#)
- [9. Funktionstest MathMind](#)
- [10. Reflexion](#)
 - [10.1 Christian Hepp](#)
 - [10.2 David Kaiselgruber](#)
 - [10.3 Markus Raab](#)

1. Sifteo Cubes

Bei Sifteo Cubes handelt es sich laut Definition um eine kleine Gamingplattform die von Sifteo Inc. hergestellt wird. Die Besonderheit dieser Cubes, die eigentlich nicht würfelförmig, sondern flache Hexaeder mit quadratischer Grundfläche sind, ist, dass sie auf mehrere Arten interaktiv sind. So reagieren sie unter anderem auf ...

- Druck (physischer Druck auf das Display, keine Touch-Bedienung)
- Rotation (vertikaler Lagesensor)
- Kippen (horizontaler Lagesensor)
- Umdrehen (horizontaler Lagesensor; umdrehen integriert ein vorangegangenes kippen)
- Schütteln (Bewegungssensor, Dauer in Sekunden)
- Aneinander reihen (Verbindung zwischen zwei oder mehreren Cubes wird hergestellt, "Domino spielen", also nacheinander in jede Richtung möglich)

Die Sifteo Cubes stellen Inhalte in einem eingeschränkten Farbspektrum dar und geben Inhalte Pixel für Pixel (mit freiem Auge sichtbar) wieder. Das Display der Cubes ist quadratisch und hat eine Seitenlänge von 3,8 cm. Die technischen Spezifikationen folgen nachstehend:

- 32-bit ARM CPU
- 128 x 128 TFT Farb-LCD
- 3-Achsen Sensoren
- 8MB Flash
- Lithium Polymer Akku
- Funkübertragung auf Frequenz 2.4 GHz (untereinander und zum Computer)
- Near-Field Objekterfassungstechnologie

Sifteo Cubes haben bereits eine Vielzahl an Anwendungsgebieten gefunden. Ganz besonders zur Freude von Kindern und Jugendlichen, da sich kleine Lernspiele in allen möglichen Bereichen mit 3-9 Cubes bestens anbieten. So können kleine Programme in vielfältigerweise helfen etwas zu lernen und dabei Spaß zu haben: Farben kennenlernen und mischen; Bilder anzeigen, dementsprechend kleine Memory-Spiele wiedergeben, Buchstaben und ganze Wörter anzeigen um diese einander zuzuordnen aber auch mathematische und viele weitere Fertigkeiten können damit gelehrt und gefestigt werden.

Offizielle Website (teilweise nicht mehr verfügbar): <https://www.sifteo.com/cubes>

Wikipedia: https://en.wikipedia.org/wiki/Sifteo_cubes



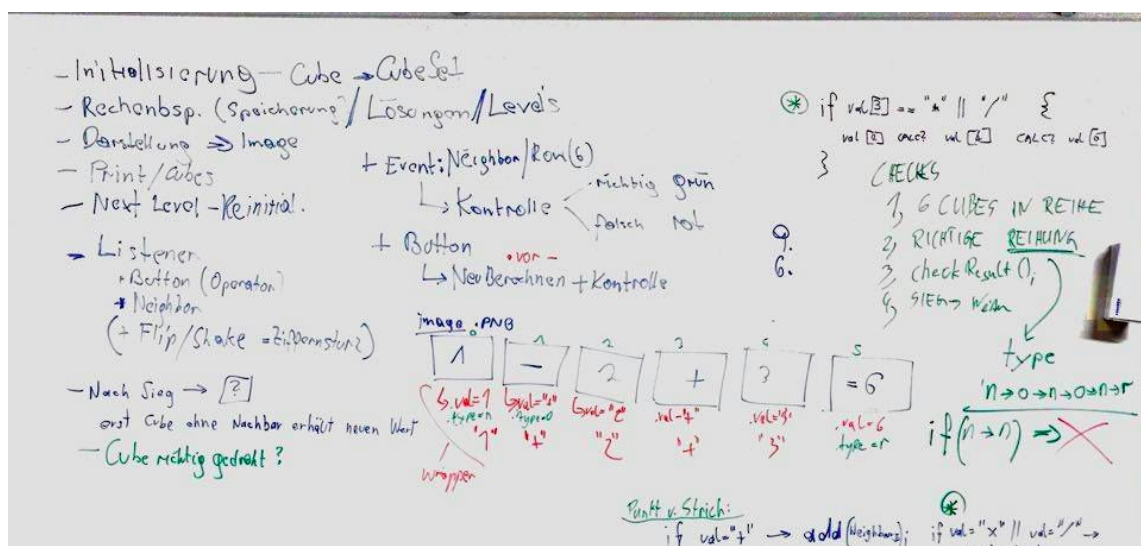
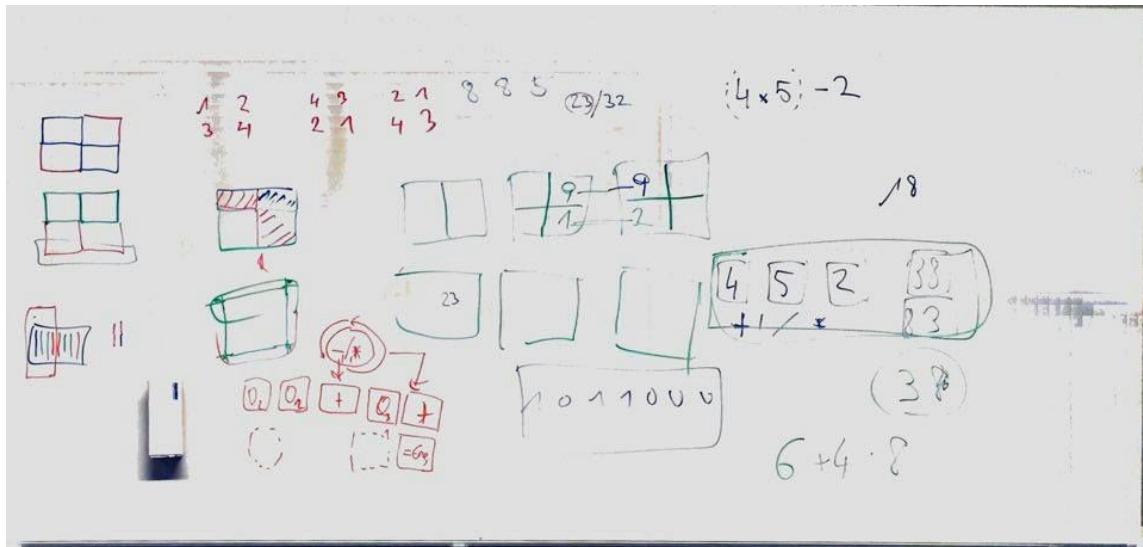
Nachfolgend noch ein allgemeines Beispiel zu Sortierung mit Hilfe von Sifteo-Cubes:
<https://www.youtube.com/watch?v=Qq1r9BCj7hc>

2. Spielidee ColorMind

Die erste Idee für ein Spiel mit den Sifteo Cubes war ein Spiel welches das Verständnis von der Zusammenführung von Farben fördern sollte. Mit ColorMind soll es also möglich sein eine vorgegebene Zielfarbe aufgrund von einer Mischung von zwei anderen Farben zu erreichen.

Obwohl die Idee (nicht die Umsetzung) des Spiels bereits weit vorangebracht wurde, zeigte sich kurz vor dem Start der Umsetzung, dass dieses Spiel in einer sehr ähnlichen Form bereits vorhanden war. Folglich musste eine neue Idee geboren werden, welche nach einem Brainstorming zu einem anderen Spiel namens MathMind führte.

Farbmischer: <https://vimeo.com/97997307> ("Colorigins": a tactile color mixing and matching game designed and developed for Sifteo Cubes)



3. MathMind

3.1 Ideation: Spielidee und Spielbeschreibung

Die Idee hinter MathMind ist es, spielerisch einfache Algebra zu vermitteln. Dabei werden die Sifteo-Cubes nach dem Prinzip von Operanden (Ziffern von 0-9) und Operatoren (+,-,x,/) verwendet, um ein vorgegebenes Endergebnis zu erreichen (zweistellige Zahl, 10-99, Ziffern können vertauscht werden). Es werden dafür sechs Sifteo Cubes benötigt:

- Eine Cube für das gewünschte Ergebnis: z. B.: "=83", nach dem Schütteln "=38"
- Drei Cubes für die einstelligen Operanden: z.B.: "4"
- Zwei Cubes für die Operatoren, welche durch Drücken des Displays durchgeschaltet werden können z.B.: "+", nach einem Klick z.B. "-"

3.2 Interaktion

Drücken, Zusammenführen, Schütteln

Die Spieler müssen nun am vorher beschriebenen Beispiel überlegen, wie die Würfel aneinanderzureihen sind, und mit welchen Operatoren sie verbunden werden müssen, um das angezeigte Endergebnis zu erzielen. Die Operatoren können durch das Drücken des jeweiligen Displays gewechselt werden. Will das Endergebnis beim zusammenführen der Cubes trotzdem einfach nicht stimmig werden, sind dort wohl die Ziffern durcheinandergewürfelt. Dies kann mit Hilfe von schütteln des Ergebniswürfels geändert werden. Somit stellt sich zu Beginn jeder Runde die Frage, ob tatsächlich das Endergebnis angezeigt wird, oder die beiden Ziffern noch vertauscht sind.

3.3 Experience

Der Zweck dieses Spiels ist es, grundlegende mathematische Kenntnisse zu lehren und dabei Spaß zu haben. Die angedachte Zielgruppe dahinter sind Hauptschüler und Hauptschülerinnen der unteren Schulstufen, denen diese Aufgaben durchaus einige Denkzeit beschern können. Besonders die Tatsache, dass nicht immer schlicht auf das Ergebnis hingeführt werden, da sich dieses im Ziffernsturzmodus befinden kann, erweitern die Denkaufgaben im Vergleich zu gewöhnlichen Rechenaufgaben. Das Kombinieren mehrere Cubes erwies sich im Test als spannende und amüsante Aufgabe, was zu einer länger andauernden Lernsession führt, als Aufgaben, die mit Stift und Papier gelöst werden müssen. Der automatisierte Wechsel in das jeweils nächste Level motiviert zusätzlich weiterzumachen. In vielerlei Hinsicht ist dieses Projekt also ein gänzlich neues, da es anhand von Spaßfaktoren zum Spielen einlädt, gleichzeitig aber fordernde Denkaufgaben für die entsprechende Zielgruppe darstellt und diese mit Hilfe eines Systems durchgeführt werden, welches keinen großen kommerziellen Erfolg feiern konnte. Es handelt sich also im Gegenzug zu Touch-Systemen um eine neue Erfahrung, die gesammelt werden kann und

welche wiederum neue, noch selten dagewesenen Interaktionen wie die bereits besagten (schütteln, physisch drücken, zusammenführen) ermöglicht.

4. Tools & Methoden

Bei den vorbereitenden Maßnahmen im Projekt (Ideation Phase) wurden vor allem auf Diskussion und Whiteboard Sketches gesetzt. Darüberhinaus wurde diese Phase auf mehrere Termine verteilt, um eine Idee reifen zu lassen.

Technisch wurden neben den Sifteo Cubes und den dazugehörigen Tools vor allem auch Dropbox, Google Docs und Github eingesetzt. Dropbox wurde, wie üblich, zum Datenaustausch genutzt. Dies bezieht sich vor allem auf Grafiken und andere Inhalte, die auf den Sifteo Cubes ausgegeben werden sollten. Zur laufenden Dokumentation wurden Google Docs herangezogen, um ein regelmäßiges und gleichzeitiges Arbeiten daran gewährleisten zu können. Github erwies sich bei der eigentlichen Umsetzung als hilfreiches Tool, da es auch hier eine gute Art der Zusammenarbeit beim Programmieren der Cubes bot. Auch was Versionierung, Dokumentation innerhalb des Codes oder schlichte Hinweiskommentare betraf, ist dies innerhalb von Github, seinen Branches und weiteren Optionen bestens gegeben.



Ob sich jedoch Github für so überschaubar kleine Projekte als Effizienzsteigerung uneingeschränkt empfehlen lässt, ist uns beim Sifteo Projekt nicht augenscheinlich geworden. Denn die Gelegenheiten, an denen getrennt und gleichzeitig am Code gearbeitet wurden, waren gering bis nicht existent und der für das Zusammenführen und Trennen der Branches und Master-Versionen benötigte gemessen an den verhältnismäßig geringen Änderungen am Code doch beträchtlichen Zeitaufwand und war immer wieder eine Herausforderung.

Für größere Projekte ist die Verwendung von Versionierungssystemen jedoch dahingehend bestimmt eine Effizienzsteigerung und ein nicht mehr weg zu denkendes Sicherungskriterium.

5. Technische Vorbereitungen

Da die Umsetzung auf einem Mac OS durchgeführt wurde, folgende Anleitung für das gewählte Betriebssystem. Damit es zu keinen Verwechslungen kommt ist es wichtig zu wissen, dass es 2 Generationen der Sifteo Cubes gibt.

- Generation 1: SDK basiert auf C#
- Generation 2: SDK basiert auf C++

Anleitungen im Web für die Installation beziehen sich nicht immer eindeutig auf die jeweilige Generation. Im Zuge des Projektes standen die Sifteo Cubes der Generation 1, auf Basis von C# zur Verfügung. Die folgende Anleitung bezieht sich daher ausschließlich auf Generation 1 für eine Mac OS Installation.

5.1 Download Software

- Mono Runtime: <http://www.mono-project.com/download/>
- Entwicklungsumgebung Xamarin Studio: <http://www.monodevelop.com/download> bzw. <https://xamarin.com/download>
- SDK Mac: https://github.com/stefansundin/The-Tangibles/blob/master/assets/Sifteo_SDK_mac_1_1_3.zip

Zusatzdownloads für Windows:

- .NET 4.0 Framework muss installiert sein
- SDK Windows: https://github.com/stefansundin/The-Tangibles/blob/master/assets/Sifteo_SDK_win_1_1_3.exe

Empfohlene Zusatzsoftware

Github Desktop für kollaboratives Arbeiten: <https://desktop.github.com>

Alternative zu Github Desktop: <https://www.sourcetreeapp.com>

5.2 Inbetriebnahme

1. Sifteo Connector per USB anschließen
2. Siftdev.app starten

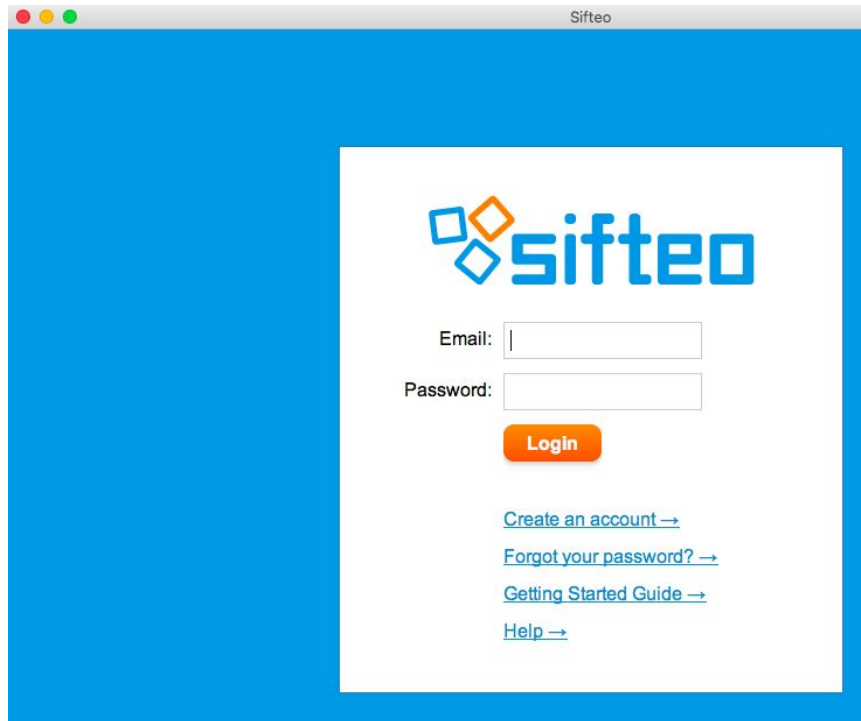


3. Anmelden / Login:

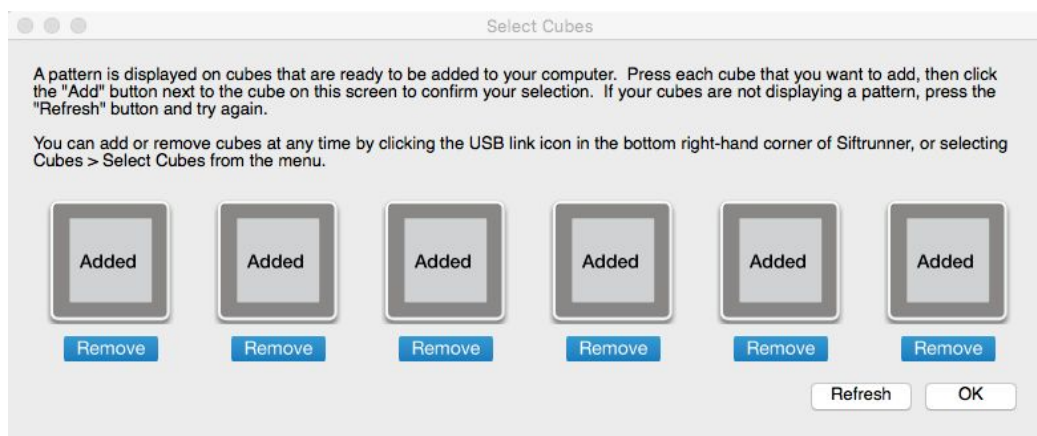
Da es aufgrund der nicht mehr vorhandenen Webseite nicht mehr möglich ist einen neuen Account anzulegen wird ein bereits vorhandener Account verwendet. Als nächsten Schritt daher mit folgenden Benutzerdaten anmelden.

User: alexander.meschtscherjakov@sbg.ac.at

PW: *****



4. Die Sifteos müssen eingeschaltet sein und nach Anleitung des Bildschirmdialogs mit der Siftdev.app verbunden werden.



5. Nachdem die Sifteos eingeschaltet und verbunden sind muss die gewünschte App auf die Sifteos geladen/installiert werden. Dazu wählt man über das Menü Developer/Load Apps... die gewünschte App aus.



6. Auswahl der zu installierenden App und Installationsprozess durchlaufen lassen.



7. Um die App tatsächlich auf den Sifteo Cubes zu starten muss das Sifteo Projekt (also der Ordner der zuvor bei LoadApps ausgewählt wurde) in der Entwicklungsumgebung geöffnet und ausgeführt werden. Mehr Infos dazu im folgenden Abschnitt.

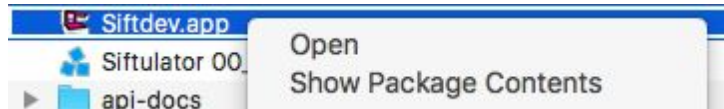
5.3 Erstellen & Starten einer App

Damit eine App erstellt werden kann muss entweder eine neue App über den Project Generator (Project_gen.exe) erstellt werden (nur für Windows), oder alternativ ein Demo-Template verwendet und umgeschrieben werden (hauptsächlich muss der Name der App von einzelnen Dateien und innerhalb von Dateien umgeschrieben werden, z.B. innerhalb Manifest.json und weiteren Dateien). Bei der zweiten Lösung zur Erstellung einer neuen App wird empfohlen alle vorhandenen Projektdateien des Templates mit einem Texteditor zu öffnen und mit "Suchen + Ersetzen" den Projektnamen umzubenennen.

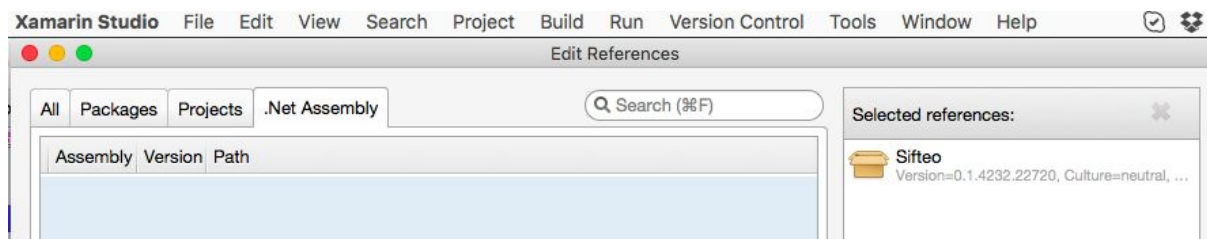
Zum Starten der App auf den Sifteo Cubes, muss diese im Hintergrund auch auf dem Rechner (quasi Server) laufen. Dazu wird das Projekt geöffnet - im Beispiel von Xamarin als Entwicklungsumgebung muss dazu die .SLN Datei geöffnet werden.

In der Entwicklungsumgebung muss vermutlich noch der korrekte Pfad zur Sifteo.dll angegeben werden (diese befindet sich bei der Mac SDK standardmäßig innerhalb der Siftdev.app → Show Package Contents:

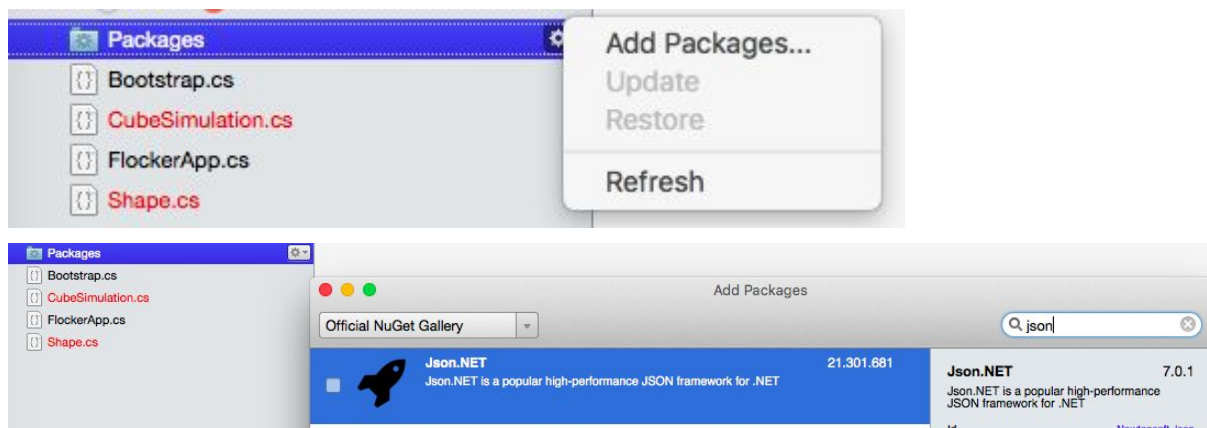
sifteosdk1/Siftdev.app/Contents/Resources/Runtimes/Mono/Current/sifteo/Sifteo.dll



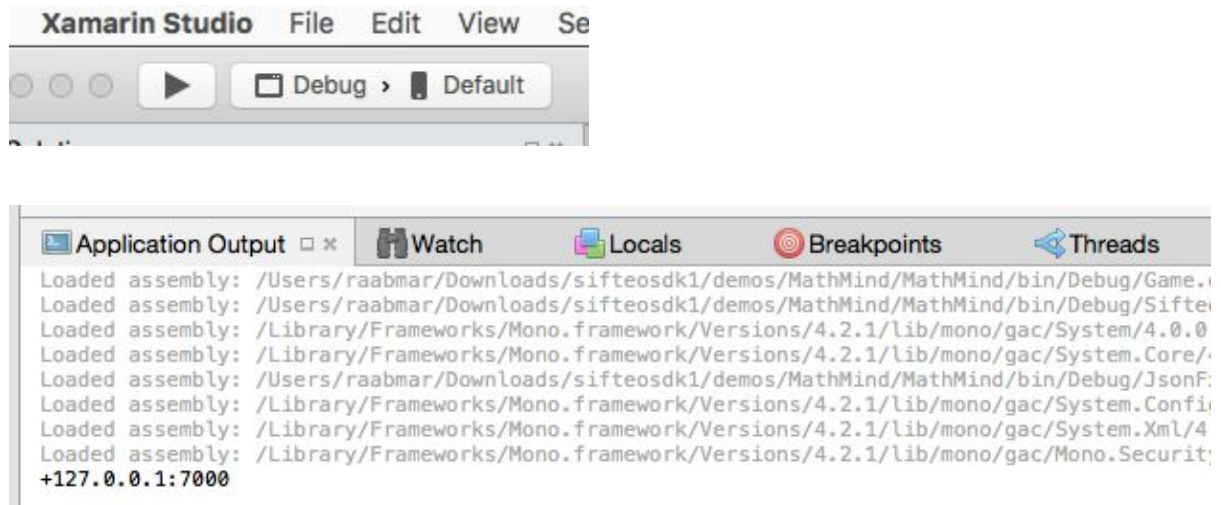
Der Einfachheit halber sollte die Sifteo.dll Datei einfach in das jeweilige Projektverzeichnis kopiert werden und danach in Xamarin über Project → Edit References → .Net Assembly dem Projekt hinzugefügt werden.



Mit sehr hoher Wahrscheinlichkeit wird auch noch das JSON FX Package fehlen. Die jeweilige Fehlermeldung dazu kommt normalerweise beim ersten Versuch das Programm auszuführen (run). Hinzugefügt werden kann das Package über das Kontextmenü von Packages (siehe Screenshots).



Nun ist alles vorbereitet um die Applikation zu starten - dazu einfach auf das Run-Symbol klicken. Das Resultat sollte folgendermaßen aussehen. Wenn dem so ist, kann nun in der Siftdev.app die ausgewählte Applikation mit "Play" gestartet werden.



5.4 Weitere wichtige Inhalte der SDK

- Dokumentation: ../sifteosdk1/api-docs/index.html
- Demos und Dokumentation der Demo Beispiele
- Project_gen.exe zum Erstellen neuer Projekte (muss nicht verwendet werden)
- Sifteo.dll (die Bibliothek muss von der App aus erreichbar sein)
- Siftulator 00_05_09 alpha.app (sehr schlechter Simulator der Sifteo Cubes)

5.5 Weitere wertvolle Links zur Installation / Ausführung

<http://sean.voisen.org/blog/2011/10/up-and-running-with-sifteo-sdk/>

http://jahej.com/alt/2011_10_04_getting-started-with-sifteo-cubes.html

Dokumentation: ../sifteosdk1/api-docs/index.html

6. Basiskenntnisse & Recherche

Um nach vorangehender Abstimmung hinsichtlich dem Projekt mit den technischen Aspekten überhaupt beginnen zu können, ist es essentiell, die Basis davon auf den eingesetzten Computern aufzusetzen. Dabei wurden zunächst die Sifteo SDK und eine passende IDE die damit kommunizieren und mit der für Sifteos der 1. Generation notwendigen Programmiersprache C# umgehen kann - in diesem Fall Xamarin Studio - installiert.

Um das Vorhaben erfolgreich umsetzen zu können, wurde ausführlich nach bereits vorhandenen Sifteo-Beispielen sowie Umsetzungsmöglichkeiten im mathematischen Bereich recherchiert und erste Gedankenexperimente, sowie Prototypen angefertigt, um etwas neues und einen optimalen Spielablauf garantieren zu können. Grundsätzlich gilt es zu beachten, dass die Displays der Sifteos relativ klein sind und sich somit womöglich nicht alles darstellen lässt, was man sich auf Anhieb vorstellt. Dahingehend wurde ebenfalls recherchiert, da es Funktionen gibt, die pixelgenaue Darstellungen von Inhalten (größtenteils) ermöglichen.

Die Recherchen führten auf viele Websites von begeisterten Programmierern und deren privaten, aber auch teilweise via GitHub veröffentlichten Projekten, welche für den Einstieg gut beschrieben und dokumentiert waren (z.B. <https://github.com/jbmartin/SifteoBlickets>). Auch die einführenden Demobeispiele sind dokumentiert und vermitteln eine gute Ausgangsbasis, um eigene Projekte anlegen und aktiv losstarten zu können.

7. Herausforderungen

7.1 Software

- Die offizielle Website der Sifteo Cubes ist größtenteils offline. Damit ist gemeint, dass Inhalte grundsätzlich ohne Styling oder Formatierung angezeigt werden, vielfach in nicht mehr verfügbaren Links enden und keinerlei Metadaten, Mediendateien oder Zusatzinformationen angezeigt werden.
- Damit geht einher, dass die SDK nicht mehr von der offiziellen Website erhältlich ist. Selbstverständlich geht etwas, was erst einmal im Internet war nicht so schnell verloren, nichtsdestotrotz gestaltete es sich als schwieriger als erwartet, die passende SDK für die Cubes der 1. Generation für mehrere Betriebssysteme zu finden und diese dann zum Laufen zu bringen, da mehrere Treiberprobleme bestanden und sich häuften (Kompatibilitätsproblem unter Windows bei installiertem Catalyst Grafikkartentreiber, x86-Kompatibilitätsproblem ebenfalls unter Windows, teilweise Downgrades von .NET Framework unter Mac notwendig etc.).
- Da es sich bei C# um eine Microsoft-Programmiersprache handelt, sind für andere Betriebssysteme wie Mac und Linux umfangreiche Treiberinstallationen und eigene IDEs notwendig.
- Es scheint keinerlei Support für die Cubes zu geben, da wie bereits erwähnt die Website praktisch down ist und auch Communities nur sehr spärlich existieren, oftmals auch nur noch ausgestorben weiterexistieren ohne ernsthaft betrieben zu werden.
- Selbiges gilt für die Social Media Auftritte der Sifteo Cubes. Auf der offiziellen Facebook Seite wurde zuletzt am 27. August 2014, also vor knapp 1,5 Jahren gepostet was für starke Überholung spricht. Auf Twitter entstand der letzte Tweet sogar bereits am 29. Juli 2014. Auch seitens aktiver User wurden letztere externe Beiträge auf besagten und weiteren sozialen Kanälen vor längerer Zeit gepostet, das allgemeine Interesse scheint gegenüber den Vorjahren stark gesunken zu sein.
- Leider gibt es keine Möglichkeit in Form einer vorgegebenen Funktion Buchstaben, Ziffern und andere Zeichen tatsächlich auf den Cubes auszugeben. Lediglich ein vorgefertigtes Gesamtbild aller Zeichen existiert. Dieses muss rangezoomt und an die richtige Stelle gerückt werden um einen jeweiligen Letter anzuzeigen. Dies erscheint angesichts eines 128*128 Pixel großen Displays unvernünftig, eine simple print-Funktion mit vorgefertigten Zeichenkodierungen wäre ebenso ressourcenschonend (wenn nicht sogar noch komprimierter) und würde vielmehr der menschlichen Logik und einem ansprechenden Workflow entsprechen.

7.2 Hardware

- Die Sifteo-Cubes (insbesondere die, der 1. Generation) entsprechen in vielerlei Hinsicht nicht dem aktuellen Stand der Technik. Dies betrifft unter anderem die Anzeige auf dem Display, auf welchem nur eine eingeschränkte Anzahl an Farben dargestellt werden kann und somit Inhalte eventuell nicht so ausgegeben werden, wie gewünscht. Weiters ist die Interaktion mit dem Display betroffen, da es nicht auf Touch-Gesten reagiert, was heutzutage aber oftmals vorausgesetzt wird. Außerdem galt es zu beachten, dass aufgrund der niedrigen Pixeldichte und der eher kleinen Displays der Geräte anzuzeigende Inhalte trotzdem klar und leserlich dargestellt werden.

8. Implementierung / Ergebnis

Das Endergebnis mit Source Code + weitere Dokumentation findet sich online auf Github wieder: <https://github.com/raabmar/MathMind>

Die Dokumentation der einzelnen Klassen ist daher dem Source Code (MathMind.cs) zu entnehmen.

Gelöste Bugs / neue Features (per 21.12.2015)

- Punkt vor Strich Regelung korrigieren (wenn Strichrechnung zu Beginn und Punktrechnung am Schluss)
→ Behoben: Fehlerhafte Berechnung korrigiert
- Bei richtiger Lösung warten bis OnNeighborRemove
→ Behoben: erst bei auseinandergeben der Würfel neu initialisieren
- Operator sofort zu Spielbeginn ersichtlich
→ Behoben: Question Mark als Operator, daher kein Operator fix vorgegeben
- Question Mark als Operator sollte nach Initialisierung bei onButton nicht mehr auswählbar sein
→ Behoben: Filename umbenannt "op" als Suffix bei Question Mark Datei entfernt
- Operator umschalten soll gleichmäßig sein
→ Behoben: Durch Deaktivierung Question Mark in der Reihe beim Durchschalten der Operatoren korrekte Modulo-Berechnung
- Reihenfolge der Rechnung muss egal sein (z.B. $8 + 7 * 3 = 29$ == $7 * 3 + 8 = 29$)
→ Behoben
- Richtige Lösung - grün hinterlegt, anstatt grüner Fläche; falsche Lösung - rot hinterlegt, anstatt roter Fläche
→ Behoben

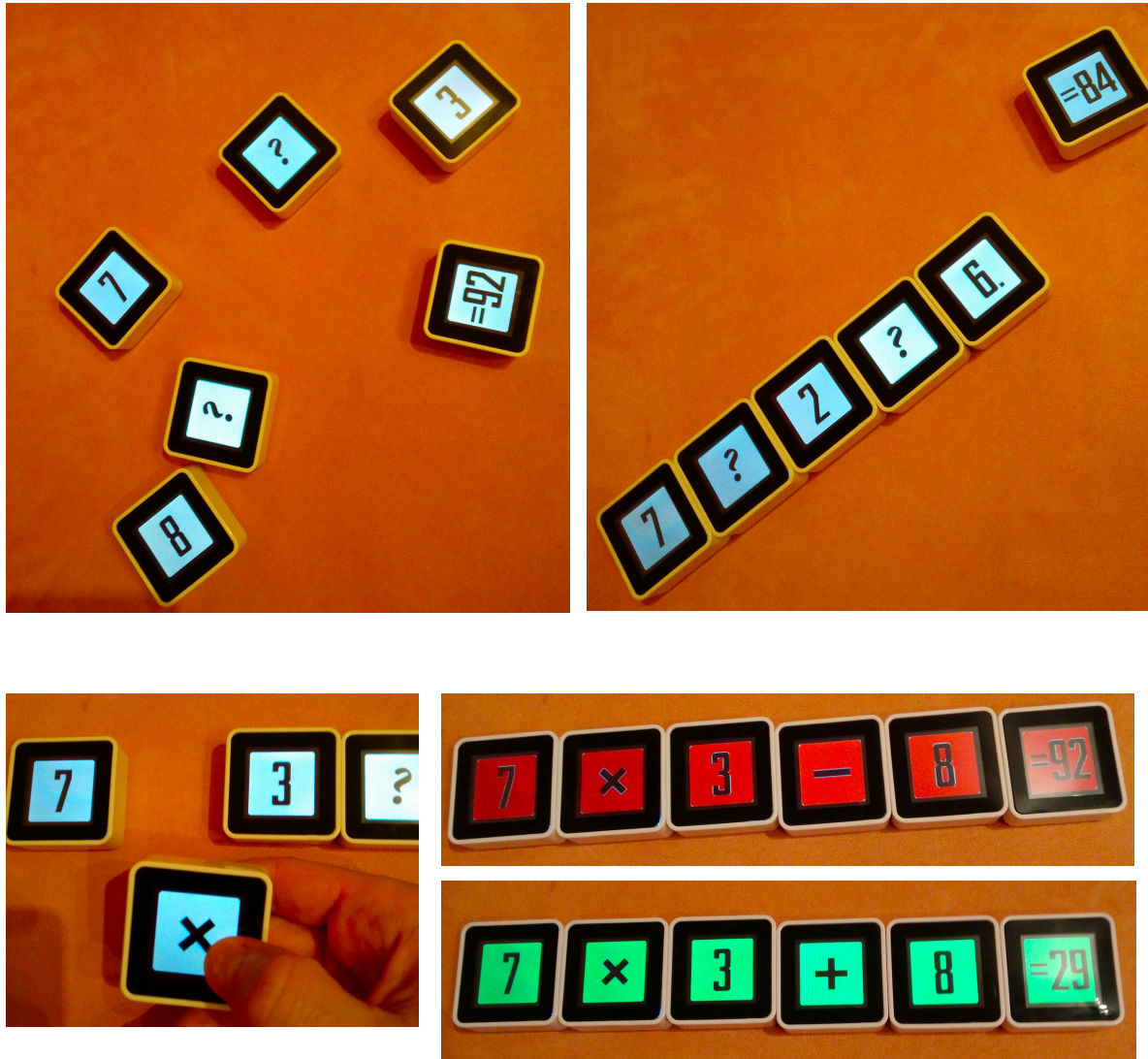
Offene Bugs / fehlende Features (per 21.12.2015)

- Reihenfolge Beispiel-Initialisierung
Wenn gelöst nach Musterlösung, dann liegen für die nächste Aufgabe die Cubes bereits in der richtigen Reihenfolge
→ Workaround: Bei den angegebenen Beispielen die Reihenfolge verdreht (statisch, nicht dynamisch)
- Drehen der Cubes (besonders bei Vorzeichen nicht erkennbar)
Wenn Cube gedreht wird es nicht als Cubereihe erkannt
→ Workaround: bei allen Cubes den unteren Rahmen grafisch kenntlich gemacht

- Nach Durchgang von allen 10 Beispielen (und Beginn von vorne) werden bei Operator-Cubes keine Fragezeichen mehr angezeigt
→ Open
- Cubes dürfen beim Spielstart (erster Durchgang, erstes Rechenbeispiel) nicht bereits vorab in einer Reihe zusammenliegen
→ Open (Anleitung Spiel)

9. Funktionstest MathMind

Das Spiel wurde mehrmals getestet und die meisten Fehler wurden beseitigt. Anbei einige Bilder und ein erstes Video von MathMind.



Video des funktionstüchtigen Spiels:

<https://goo.gl/photos/bV1FuZSKCVhQV5gr5>

10. Reflexion

10.1 Christian Hepp

Als die Sifteo Cubes im Rahmen der Lehrveranstaltung vorgestellt wurden, juckte es mir offen gestanden gleich von Beginn weg unter den Fingernägeln. Das Konzept von kleinen, mobilen und mit Sensoren ausgestatteten Spielebausteinen ließ für mich einen großen Schatz an interaktiven, aber kompakten Funktionsprototypen offen.

Tatsächlich war dann aber bei der Recherche ein gewisser Dämpfer vorhanden, da viele Projektideen (Farbmischungsspiele, etc.) entweder schon wie konzipiert oder teilweise sogar recht ausgereift umgesetzt waren. Da wir etwas Neues entwickeln wollten, beschlossen wir dann in der Ideation Phase auch einfache, vorher beiseite gestellte Ideen wieder aus der Schublade zu holen und neu zu überdenken. Der Prozess war dann durchs kleine motivierte Team relativ effizient und die Idee unseres Rechenspiels war geboren.

Die Funktionalität und die Aufgabenbereiche waren relativ schnell abgesteckt und das Arbeiten im Team funktionierte sehr gut. Unnötiger Abstimmungsaufwand wurde dadurch vermieden, dass wir die Hauptentwicklungsschritte stets gemeinsam in halbtägigen Sessions durchführten. Natürlich mussten wir uns erst in die Entwicklungsumgebung, die (wenn auch kompakte) API und die damit einhergehenden Restriktionen klar werden, da die Anzeige von konzipierten Spielescreens nicht so möglich war, wie von anderen Plattformen gewohnt. Da wir größtenteils gewohnt sind, per Textoutput und anschließendem Styling (Hinzufügen von Bildern, Farben und sonstigen ästhetischen Schmankerln) Schritt für Schritt brauchbare Screens zu produzieren, war die Herangehensweise der Sifteo API, die oftmals Ziffern und Buchstaben als entsprechende Bitmap-Grafik an die Cubes zu senden, eine gewöhnungsbedürftige.

Positiv hervorzuheben ist dabei aber, dass die API funktional noch vergleichsweise relativ überschaubar war, was neben vorher beschriebenem Umdenkpotential auch das Projekt zeitlich nicht überwuchern ließ. So konnten wir, im uns gesteckten Zeitfenster unser gestecktes Ziel erreichen.

Dass die Sifteo-Cubes offenbar vom Entwickler aufgegeben wurden, ist schade. Viel Potential würde in der Spieleentwicklung definitiv noch drin stecken, allerdings ist wohl die Zielgruppe der programmieraffinen Tüftler oder generell der Maker Community eine zu kleine, um kommerziell neben mobilen Plattformen wie iOS und Android bestehen zu können. Das aufgegebene Projekt merkt man hier und da auch wie oben beschrieben noch an etwas ungeschliffenen Methoden, den verschiedenen Hardwaregenerationen und den unterschiedlichen Herangehensweisen, die Cubes anzusprechen, oder Funktionalität umzusetzen.

10.2 David Kaiselgruber

Obgleich ich zunächst skeptisch war, ein Projekt mit einer mir gänzlich unbekannten Technologie durchzuführen, war das Kennenlernen dieser Sifteo Cubes definitiv spannend. Spätestens nach ersten Recherchen, die sich als äußerst vielversprechend erwiesen, da ich dabei auf Videos und Beispiele gestoßen bin, die ich mit einem solchen System nicht für möglich gehalten hätte, wurde meine Neugierde stark geweckt. Die Tatsache, dass damit nicht nur statische Spiele und Programme wie Farbmischung, Mathematikaufgaben oder Memoryapps möglich waren sondern vor allem dynamische Anwendungen die Super Mario oder anderen Jump and Run Games durchaus ähnlich waren, wertete dieses System für mich persönlich stark auf.

Leider wurde dieser Euphorie schnell ein starker Dämpfer versetzt, als wir uns im Projekt mit dem Suchen, Downloaden und Installieren aller notwendigen Quellen befassten. Dies erwies sich, wie im Bericht auch näher erläutert, als besonders schwierig, da weder die offizielle Website, noch deren Links, Communities oder auch externe Nutzergruppen erreichbar oder aktiv betrieben wurden. Diese Punkte sowie anfängliche Schwierigkeiten im Umgang mit einer unbekannten Technologie sind als negative Erfahrungen anzumerken. Trotz allem wurden diese Probleme überwunden was zu den ersten Konzeptionen für Umsetzungsideen führte. Obwohl das Team einige Zeit versuchte, an der ursprünglichen Idee des Farbmischers festzuhalten, waren adäquate Lösungsvorschläge für die neue Idee des Rechenprogramms relativ schnell und durchdacht gebrainstormed, was für die weitere Vorgehensweise im Projekt wiederum Motivation schaffte.

Die Tatsache, dass es sich bei den Sifteo Cubes um eine Technologie handelt, welche sich wohl nicht am Markt etablieren kann beziehungsweise auch nicht konnte, wie es anhand der mehr oder weniger vorhandenen Onlinepräsenzen wirkt, ist zwar schade, stellte aber zu keinem Zeitpunkt ein Problem dar. Aufgrund der Vielfältigkeit, dem oftmals relativ geringen Preis und dem hohen Marktanteil von Touchgeräten jeglicher Art, ist es für ein Unternehmen, das von der Norm abweicht, verständlicherweise schwierig sich durchzusetzen.

Besonders positiv in Erinnerung habe ich die bereits kurz erwähnte Phase der Konzeptionierung und Ideengenerierung. Trotz der vielen Gruppenarbeiten im Studium, ist es sehr selten, dass gemeinsam aktiv an Tafeln, Whiteboards und mit Hilfe von Grafiken, Brainstormings und weiteren Methoden gearbeitet wird. Dies führt zum Aufbau vieler guter Ideen, da sich nach dem Schneeballprinzip aus einer Idee die nächste, noch ausgereifere schnell ergibt. Doch nicht nur die Durchführung dieser Phase empfand ich als sehr positiv, sondern auch die Ergebnisse davon sprechen für sich und führten nicht zuletzt zu einem erfolgreichen Vorzeigeprojekt.

10.3 Markus Raab

Die Idee ein Projekt mit den Sifteo Cubes umzusetzen ist meiner Meinung nach wirklich gut, da es sich dabei um eine wirklich abgespeckt, eingeschränkte Entwicklungsumgebung handelt. Dies stellt keinesfalls einen Nachteil dar, sondern einen Vorteil, da man bei anderen Projekten, z.B. Android Programmierung, als Anfänger wohl eher überfordert ist. Man hat daher eine Chance in einer eher kompakten Entwicklungsumgebung zu experimentieren und sich Schritt für Schritt an eine Lösung heranzutasten - darüber hinaus erhält man Feedback auf Hardware-Bausteinen, was bei Programmierungen anderer Art häufig nicht der Fall ist und somit motivierend ist.

Die Sifteo Cubes bringen allerdings auch große Nachteile mit sich, allen voran da sie in dieser schnelllebigen Zeit der Technik nach nur wenigen Jahren bereits sehr veraltet sind. Merkbar ist das sowohl an der Hardware, als auch der Software, die online nicht mehr wirklich auffindbar war (Details dazu im Kapitel 5 der Doku).

Die Ideation Phase war meiner Meinung nach sehr spannend, da wir tatsächlich mit Recherche, Zeichnungen am Whiteboard und Diskussionen nach und nach unsere Idee entwickelt und verbessert haben.

Ich denke dass mit MathMind ein cooles, kleines Game rausgekommen ist, dass, wie Tests mit externen Personen zeigten, herausfordernder sein kann als wir zuerst dachten.

Das (relativ) veraltete System brachte jedoch einige Herausforderungen mit sich um das Projekt erfolgreich abzuschließen, weshalb ich von einer zukünftigen Entwicklung mit dieser Plattform eher abraten würde.