Kids Programming – Thalia fördert bereits die Jüngsten!

Nachdem wir in den letzten Jahren einige Sommerpartys und eine riesige Weihnachtsfeier hatten, hat sich unsere Geschäftsleitung entschieden, für Ende 2017 einen „Family Day“ abzuhalten. Und so durften nicht nur die Lebensgefährten sondern auch alle Kinder mal schauen, was Papa / Mama so im Büro machen. Es gab an den einzelnen Standorten ein umfangreiches Rahmenprogramm. Dabei wurde stets versucht, auch einen gewissen Bezug zum Unternehmen zu wahren. So bot sich die Bücherecke mit Vorlesungen für Kinder natürlich an, was vor allem einige der jüngeren Kinder sehr genossen haben. Aber was macht man an einem Software Standort und wie erklärt man die Entwicklung von Programmen seinen Kindern?

Inspiriert von der „Langen Nacht der Wissenschaft“ in Berlin, wurde die Idee des Kids Programming geboren. An einzelnen Plätzen konnten so die Kinder bereits ab 4 Jahren in die Welt des Programmierens eingeführt werden und eine Idee entwickeln, was ein Software Entwickler eigentlich macht.

Die grundlegende Idee

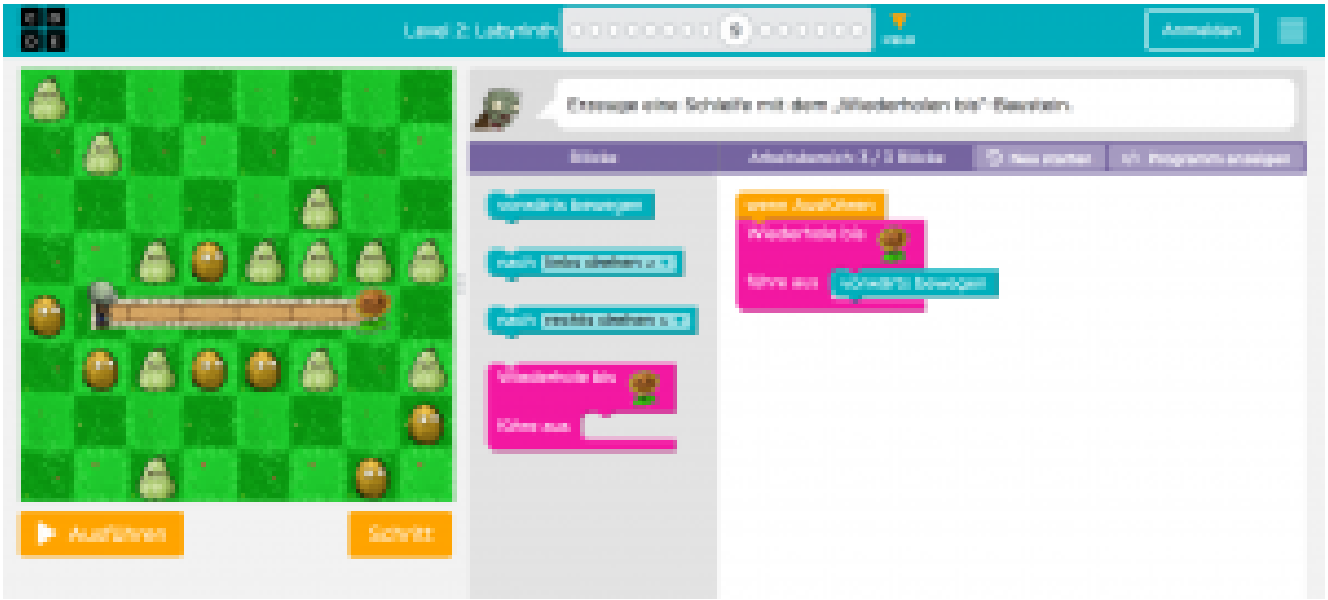
Wir haben dafür mit Hilfe von code.org einige Programme für die Altersgruppe 4-6 und 6-12 rausgesucht und unterschiedliche Themen gewählt. Dadurch konnten wir später nicht nur Jungen sondern auch einige Mädchen für die Kurse begeistern. Für die fortgeschrittene Programmierung stellten wir darüber hinaus noch Scratch vom MIT zur Verfügung und konnten das Ganze durch einige, speziell für Kinder gestaltete Lehrbücher, unterstützen.

Darüber hinaus wollten wir bei den Kindern aber auch das Interesse für Softwareentwicklung wecken und sie neugierig auf mehr machen. Deshalb haben wir unsere Raspberry Farm ebenfalls zur Scratch Spielwiese umfunktioniert und

mit einigen Hardware Projekten ergänzt. Das wurde dann zum echten Blickfang und sorgte für die Aufmerksamkeit von jung und alt. So bekamen wir am Ende mindestens genauso viel interessierte Eltern wie Kinder in den Kurs.

Programmieren mit code.org und Scratch

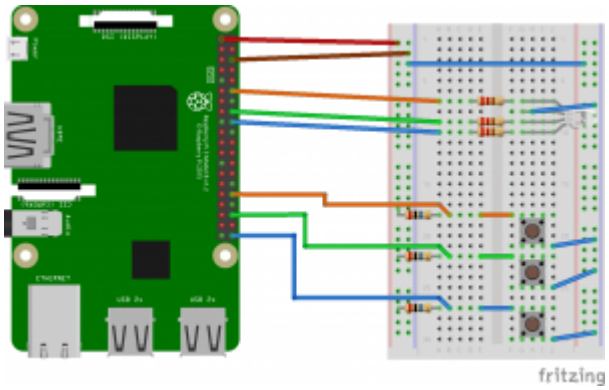
Die die Idee dahinter ist nicht eine echte Programmiersprache zu erlernen, sondern ein Verständnis von strukturierten Abläufen und Funktionsbausteinen zu schaffen. So können die Kinder mit einzelnen Code-Blöcken, im Lego Baukasten Prinzip, ihre Programme zusammenklicken. Dabei geht es vom Erkennen von Wiederholungen / Schleifen bis zum Erstellen eigener Funktionen. Während Scratch eine reine Entwicklungsumgebung bietet, unterstütz code.org das mit stuckierten, auf einander aufbauenden Kursen.



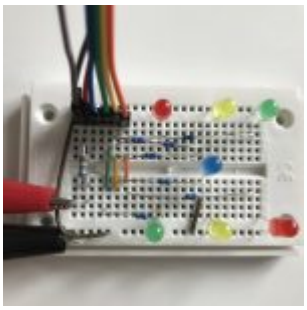
Programmieren von Hardware

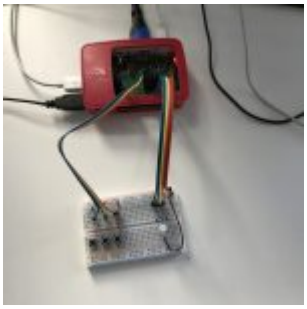
Als letzte Stufe haben wir den Kinder gezeigt, dass mit einfachen Programmen sogar elektronische Schaltungen gebaut und programmiert werden können. Zur Auswahl standen drei Beispiele:

1. Steuerung einer RGB LED mittels dreier Taster,
2. Steuerung einer Fussgängerampel mit dazugehöriger Fahrzeug Ampel und
3. Steuerung eines RGB Würfels, der durch zwei Knetkontakte gesteuert wird.



Die Programmierung erfolgt dabei auch mit Hilfe von Scratch in einer speziellen Version für Raspberry Pi. In dieser Version können die Ein- & Ausgänge als Funktion in Scratch verwendet und gesteuert werden. Inspirirt wurden diese Beispiele durch das Buch „Der kleine Hacker – Programmieren für Einsteiger“. In diesem sind die Schaltungen und Programme sehr schön auch für unerfahrene Einsteiger beschrieben. Bis auf den Raspberry wird auch alles benötigte Zubehör mitgeliefert.





Wer Interesse bekommen hat, das Ganze mal selber auszuprobieren, wird hier fündig:

Christian Immler

Der kleine Hacker: Programmieren für Einsteiger

Mit Scratch schnell und effektiv programmieren lernen

Der kleine Hacker

eBook ab 14,99 €...

Taschenbuch 25,99 €

Programmieren ist langweilig und trocken? Nicht mit dem kleinen Hacker! Die grafische Programmiersprache Scratch macht's möglich. Programmieren lernen mit Spaß. Egal, ob du nur die Katze tanzen lassen oder ein richtiges Spiel programmieren willst - Scratch eignet sich sowohl zum Einstieg in die Programmierung als auch für anspruchsvollere Projekte.

Links zu den Seiten: code.org, scratch.mit.edu

Software Delivery ist keine Abteilung!

„Software Delivery“ geht weiter als Analyse und Entwicklung. Sie hört auch nicht nach der Qualitätssicherung auf. Nach dem Deployment ist vor dem Deployment. Und der Betrieb ist sowieso inklusive, das ist mal klar.

Dieses Ziel erreichen wir nicht mit ausschließlich Analysten und Softwareentwicklern. Ohne QA-Spezialisten, Operations-Experten und Product

Ownern in diesen Teams ist die Herausforderung nicht zu schaffen.

Daher besteht „Software Delivery“ bei uns aus selbstorganisierten, cross-funktionalen Produktteams mit dem Ziel, autonom und selbstorganisiert Software zu entwickeln, zu deployen und vor allem Mehrwerte für unsere Kunden zu generieren.

Sind wir da schon am Ziel? Nein. Wollen wir da hin? Definitiv.

Tägliche Besprechung im crossfunktionalen Team. Im Hintergrund das Monitoring der Systeme in Produktion.

Um im Bereich der agilen Softwareentwicklung richtig durchstarten zu können, haben wir uns in den Produktteams für Scrum als Vorgehensmodell entschieden. Was uns dabei sehr wichtig ist:

- Verantwortung für den Gesamterfolg
- Erzielen von gemeinsamen Ergebnissen
- Regelmäßige, kritische Überprüfung der Qualität der Zusammenarbeit & der Prozesse
- Messen der eigenen Produktivität
- Transparenz im Tun und in der Kommunikation
- Kundenorientierung
- Mut haben: zur Einfachheit und Offenheit

Sind wir bei all dem schon Experten? Nein. Wollen wir das werden?

Definitiv.

„You build it, you run it“ – aus dem klassischen Legacy-Umfeld kommend bauen wir unsere Umgebung und unsere Prozesse aktuell radikal um und haben eine Menge Zukunftsbilder im Blick:

- Umbau vom ‚Release-Train‘ hin zur ‚Release-Subway‘
- Wissensaufbau im Team vom Datenbankindex bis zum Frontend-Skript
- Umbau von Infrastrukturanforderungen hin zu automatisierter Servicebereitstellung
- Erweiterung automatisierter Qualitätssicherung vom JUnit-Test zum Regressionstestautomaten
- u.v.m.

Haben wir das alles schon erreicht? Nein. Wollen wir das? Definitiv.

Wir befinden uns gerade in einer der spannendsten Phasen einer Transformation. Die Wege sind nicht ausgetreten, vieles ist neu, muss definiert oder überhaupt erst erschaffen werden. Alle müssen dazulernen.

Unsere produktbezogene Softwareentwicklung bedeutet ständige Veränderung in

Bezug auf den Kunden, den Markt und auf uns. Keine kleine Herausforderung, aber eine sehr motivierende.

Also: Software Delivery ist keine Abteilung – es ist eine ganze Menge mehr.

Martin Ernst – Teamleiter Software Delivery

Claudia Landmesser – Teamleiterin Software Delivery

Interesse & Lust bekommen, uns auf diesem Weg zu begleiten und Dich aktiv einzubringen? Dann freuen wir uns auf Deinen Kontakt.