



Calliope-Workshop

Tino Sperlich, Kids4IT

Das Schülerforschungszentrum Hamburg (SFZ) ist mittlerweile zu einer zweiten Heimat für Kids4IT geworden: In einer anregenden Umgebung haben wir wunderbare Räume und sogar einen Lagerraum.

Der letzte dreistündige Workshop stand ganz im Zeichen des Einplatinenrechners „Calliope“. Mit fünf Mentoren wurden vierzehn Schüler zwischen acht und vierzehn Jahren betreut. Die Vorstellungsrunde war sehr gut geeignet, das Eis zu brechen, und hat wirklich geholfen, sich (fast) alle Namen einzuprägen: Lisa liest, Tobias tippt, Sophie sagt, Charlotte chattet, Jakob mag Joghurt, Johann joggt, Hauke hilft, Patrick mag Peperoni, Magnus malt, Ilias isst keine süßen Sachen, Tino tanzt ...

Dann ging's gleich in die Vollen: WLAN-Passwort eingeben und den Online-Editor unter „<http://mini.pxt.io>“ aufrufen, das war schnell erledigt. Es gibt zwar einen groben, gedruckten Leitfaden, aber, wie immer, war es toll zu sehen, was den Schülern selbst einfiel: Spiele, Geheimnisse, Mathehilfe, Lichtshow. Der Autor als Calliope-Neuling war wirklich elektrisiert. Nachfolgend ein paar Beispiele, die man mit dem Code von GitHub (siehe „<https://github.com/kids4it/workshopresults/tree/master/20181027k4ithh/calliope>“) und dem Simulator auch ohne Hardware nachbauen kann.

Magnus und Ilias – Reaktionsspiel: Starte das Spiel mit beiden Knöpfen A und B. Die Lichtkette biegt zufällig nach links oder rechts

ab. Drücke rechtzeitig den richtigen Knopf (Knopf A links, Knopf B rechts), sonst ist das Spiel verloren und ein Ton ertönt.

Jakob – Binärzähler: Jakob hat mit den LEDs einen Zähler auf Zweier-Basis implementiert. Bei den ersten Versionen kam der Verdacht

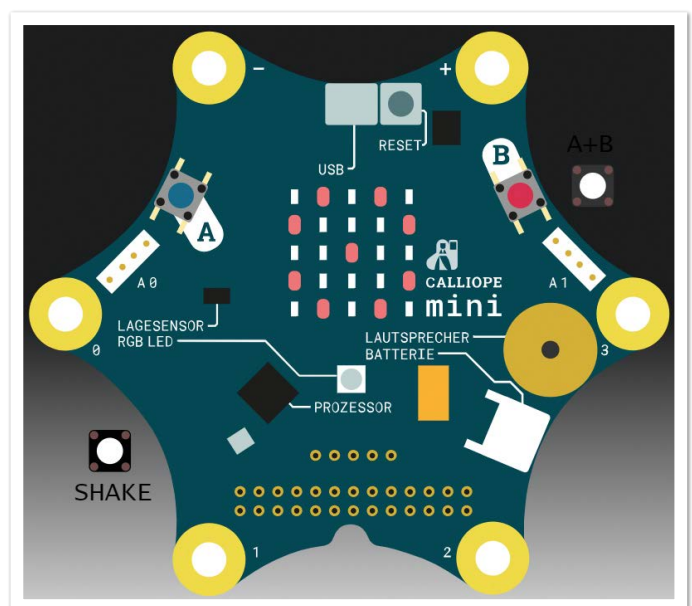


Abbildung 1: Der Calliope

auf, dass eine LED defekt ist. Was tun? Es wurde ein kleines Testprogramm erdacht, das auf Knopfdruck alle LEDs aufleuchten lässt. Die LEDs waren ok, also musste es ein Programmfehler sein. Daraufhin haben Jakob und Daniel debuggt und den Fehler gefunden.

Henry und Henri – elektronischer Würfel: Man drückt abwechselnd Knopf B, um einen neuen Wert zu erhalten. Wer eine „6“ würfelt, sieht und hört eine Belohnung.

Moritz und Kit – Pacman 2018: Man benutzt alle vier Eingabemöglichkeiten, damit sich der Pacman-Punkt über das Spielfeld bewegt: Knopf A nach links, Knopf B nach rechts, A+B nach oben, Schütteln nach unten, aber Vorsicht am Rand!

Johann und Eric – Verschlüsselung light: In bester Forschermanier wurden erst mal die Zahlenräume getestet und eine möglichst große Multiplikation gerechnet. Dann eine fast endlose Zeichenkette ausgegeben – wird der Calliope das aushalten? Und es musste eine Verschlüsselung her, um den Code zu schützen: Erst eine bestimmte Kombination von Knopf A und B gibt den gewünschten Text aus. Als Variante entstand noch ein „Keylogger“, der die Tastendrucke grafisch ausgab.

Weitere Informationen unter „<https://www.kids4it.de/calliope-workshop>“.



Tino Sperlich
tino.sperlich@gmx.de

Tino Sperlich arbeitet als Software-Entwickler und Requirements Engineer in Hamburg. Er unterstützt regelmäßig Kids4IT, eine gemeinnützige User Group, und veranstaltet monatliche Workshops in der Hansestadt, um Schülern die Faszination von Technik allgemein und Programmierung im Speziellen zu vermitteln.

Microservices mit dem Helidon Framework



Marcel Amende, Oracle Deutschland B.V. & Co. KG

Von moderner Java-Entwicklung wird heute viel Agilität verlangt. Man will schnell und einfach neue Projekte starten, diese von der ersten Minute an automatisiert testen und früh in Produktion bringen. Es braucht die richtigen Frameworks, damit man sich auf die funktionalen Anforderungen konzentrieren kann, ohne Abstriche bei Codequalität und -leistung zu machen. Das Spring-Ökosystem [1] mit dem im Jahr 2014 veröffentlichten Spring-Boot-Projekt [2] ist sicherlich ein Pionier der modularen Entwicklung in Java. Es eignet sich gut für die Entwicklung von Microservices, wobei es in Bezug auf Funktionsumfang und Größe am oberen Ende des Werkzeugspektrums liegt. Mit dem Open-Source-Projekt Helidon [3] gibt es nun eine leichtgewichtige Alternative, die insbesondere Enterprise-Java-Entwicklern einen leichten Einstieg ermöglicht.

Klein ist fein

Neue technische Möglichkeiten bringen Veränderung. Die Veränderungsgeschwindigkeit nimmt zudem immer weiter zu. Auch die „Kunst“ des Programmierens ist dadurch im steten Fluss. Java ist in dieser Hinsicht sicherlich keine Experimentierplattform, versucht allerdings konsequent, Trends aufzugreifen und diese zu professionalisieren, sobald sie sich bewähren. Für die Entwicklung von mehrschichtigen, monolithischen Webapplikationen ist mit Java EE eine beeindruckende Plattform entstanden. Heute ist jedoch die Entwicklung von (Cloud-) Applikationen in Form von kleinen, modularen und ereignisgetriebenen Diensten, sogenannten Microservices, gefragt. Die funktional reichhaltigen, aber schwergewichtigen Applikationsserver-Plattformen verlieren vor diesem Hintergrund trotz aller Optimierungen an Attraktivität. Zumal sich auch die Art der Ausbringung verändert. Statt eine reine Java-Applikation in ein Applikationsserver-Cluster auszubringen und alle weiteren Ressourcen zu referenzieren, packt man in Microservice-Architekturen in sich geschlossene Funktionalität mit allen benötigten Abhängigkeiten, Bibliotheken und Ressourcen in unabhängig lauffähige (Docker [4]-) Container. Diese werden wiederum auf einer verwalteten Containerplattform, zum Beispiel mit Kubernetes [5], betrieben.