



# Your Kids Play Games Mine Makes Games

**Katja Arrasz-Schepanski**

**@KatjaArrasz**

**Christian Mennerich**

**@cmennerich**



# Fahrplan/Agenda

- **Teil 1:** Devoxx4Kids - Was ist das eigentlich?
- **Teil 2:** Die Workshops für Kinder
- **Diskussion**



## Teil 1

**Was ist eigentlich die Devoxx4Kids?  
Und wie funktioniert sie?**



# Was ist D4K?

## Was ist die Devoxx?

- jährlich in Belgien stattfindende europäische Konferenz über Java, Android und HTML5
- die größte herstellerunabhängige Java-Konferenz der Welt
- mittlerweile Devoxx{Belgium|France|UK}



# Was ist D4K?



(Das belgische Devoxx4Kids Team.)

<http://www.devoxx4kids.org/deutschland/>

@Devoxx4KidsDE



# Was ist D4K?

- 2012 fand die erste Devoxx4Kids in Belgien statt
- Devoxx Team-Mitglieder:
  - Bildeten ihre eigenen Kinder in Programmierung aus
  - Brachten ihnen aktiven Umgang mit dem Computer bei
  - Suchten geeignetes Lehrmaterial
- Fazit: Lehrmaterial fast nur auf Englisch



# D4K Teams weltweit

- Weltweit gibt es zur Zeit 25 Teams
- Weitere 21 sind im Gespräch
- In Köln und München noch in Diskussion





# Was machen wir?

- Am 27.09.14 wird die erste Devoxx4Kids in Deutschland, Karlsruhe stattfinden
- Was werden die Kinder lernen?
  - Erste Begegnungen mit Programmierung und Computerlogik
  - Mit viel Spaß dabei!!
  - Programmierung von Spielen und Robotern
  - Einführung in Elektronik





# Wie kann man helfen?

- als Lehrer
- als Helfer vor Ort
- als Übersetzer der Workshops
- als Sponsor
- als Initiator einer eigenen Devovx4Kids in Deutschland





**Was ist bei der Durchführung einer  
Devoxx4Kids Veranstaltung zu beachten?**

# DEVOXX™ [4KIDS] Location

- Ausreichend Computer/Beamer
- Abgetrennter Bereiche, Sicherheit
- Genügend Parkplätze
- Beschilderung



# DEVOXX™ [4KIDS] Logistik

- Verpflegung (Essen und Trinken)
- Willkommens-Präsentation
- Goodies für die Kinder
- Sponsoren erwähnen
- Kopie der Workshops
- Fotos und Videos  
(Gruppenfoto, Einverständniserklärung)



# DEVOXX™ [4KIDS] Logistik

- Namensschilder/Gruppenschilder für die Kinder
- Plan der rotierenden Gruppen/Kopie für Lehrer
- 2-3 Lehrer pro Workshop
- Helfer: Organisation, Erste-Hilfe, Lehrer
- Versicherung





# DEVOXX™ [4KIDS] Logistik

- Devoxx4Kids T-Shirts  
<https://www.zazzle.com/devoxx4kids>
- TV/Radio/Zeitungen
- Online-Registrierung
- Statistiken



- Eltern sollen nicht in die Workshops rein
- ABER:
  - Kinder wollen ihre Ergebnisse zeigen
  - Zeit einplanen, damit Kinder ihren Eltern zeigen können, was sie gemacht haben
  - Eltern einbinden, wenn sie helfen möchten





# Workshops

- Programmiercode auf ein Minimum reduzieren
- Vergleiche aus dem wirklichen Leben finden
- Was motiviert die Kinder?
- Den Workshop spannend und lustig halten
- Vorhandene Workshops gibt es auf der Devovx4Kids Website frei zugänglich





## Teil 2

### Workshops für die Kinder!

Wie geht's, und was wurde schon gemacht?

# Workshops für Kinder

## Programmierung

- DrTechniko
- Scratch/Greenfoot
- Alice
- Minecraft Modding
- Sonic-Pi



## Robotik

- Lego Mindstorm
- NAO



## Internet of Things und Elektronik

- Raspberry Pi
- Arduino
- TinkerForge





# Workshops für die Kinder: Programmieren



# DrTechniko

## Eigenschaften

- Altersstufe: ab 2 bis 7
- Ein Elternteil ist der Roboter, den das Kind steuert
- Kein Computer notwendig
- Vorstellungskraft und Phantasie!

## Lehrkonzept

- Einfaches algorithmisches Denken
- Anweisungen (Bewegen, Drehen, Greifen etc.)
- Wiederholungen und Schleifen
- Ereignisbehandlung



<http://drtechniko.com/>

<http://www.devoxx4kids.org/deutschland/>

@Devoxx4KidsDE

**Dr TECHNO**  
www.facebook.com/drtechniko

ROBOT LANGUAGE DICTIONARY

Dr kid name

	LEFT	RIGHT
LEG FORWARD		
LEG BACKWARD		
BODY ROTATE		

GRAB

DROP

TALK "BIT BOT"

**empty space  
for inventing new  
commands**



Bilder von der DrTechniko Homepage:  
<http://drtechniko.com/>



# Scratch

## Eigenschaften

- Altersstufe: 5-15
- Puzzleartig, blockbasiert
- Veröffentlichen und 'Teilen' von Projekten
- Einbettung der Spiele in Webseiten



<http://scratch.mit.edu/>

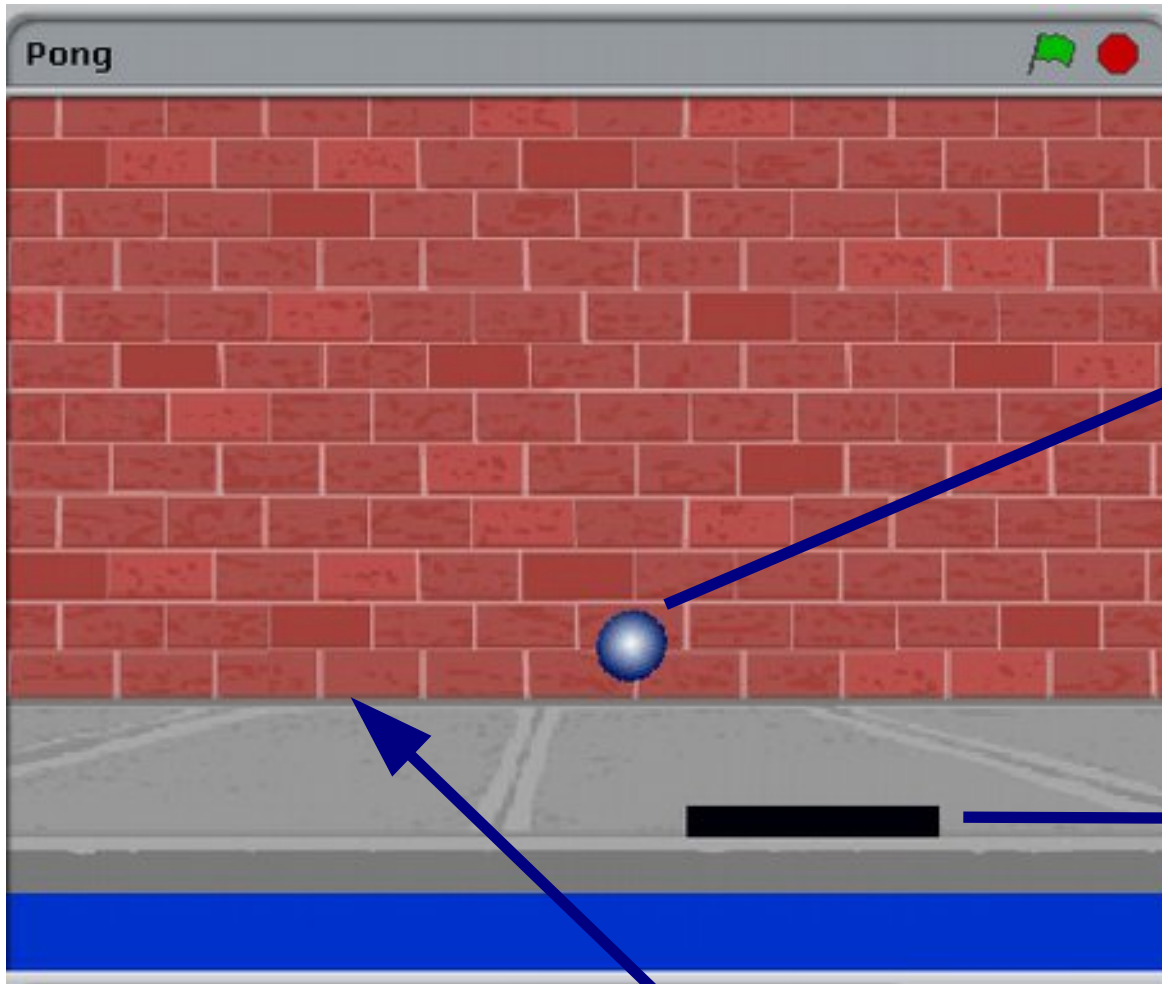
## Lehrkonzept

- Programmstruktur
- Sequentielle Abarbeitung
- Bedingungen und Verzweigungen
- Ereignisbehandlung
- Variablen und Strukturen
- Parallelität
- Synchronisation
- Benutzerinteraktionen





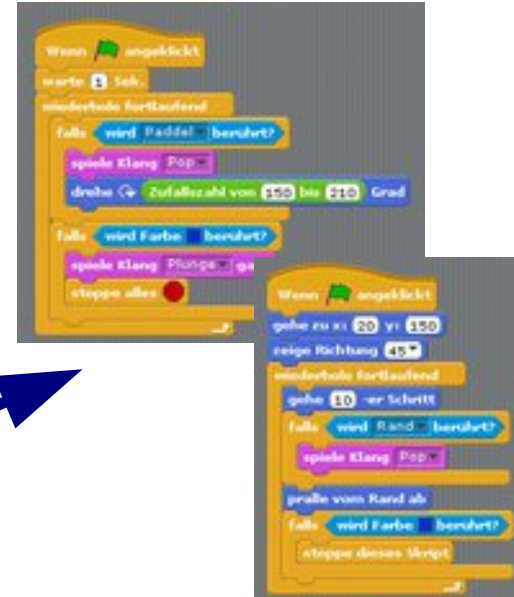
# Scratch - Pong



Kugel

Paddel

Hintergrund



## Eigenschaften

- Altersstufe : 10-15
- auch für Erwachsene geeignet
- Komplette 3D-Welten
- Geschichten erzählen mit Programmierung und Programmcode
- Basiert auf Java



## Lehrkonzept

- Objektorientierte Programmierkonzepte
- 3D Objekte
- Interaktion von Objekten
- Navigation von Kamera und Akteuren
- Geschichten erzählen
- Benutzung einer IDE

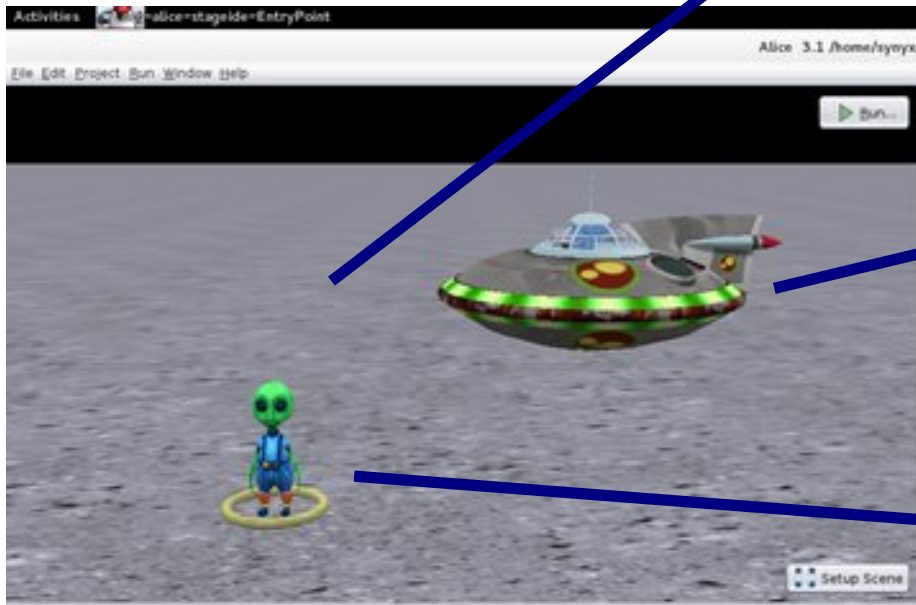




# Alice - Auf dem Mond

```
Scene initializeEventListeners myFirstMethod  
declare procedure myFirstMethod  
do in order  
  this.alien say "Hallo Welt!!", bubbleFillColor WHITE, duration 1.0 add detail  
  this.alien turnToFace this.myUFO, animationStyle BEGIN_AND_END_GENTLY add detail  
  this.alien moveToward this.myUFO getRightDoor, 2.0, animationStyle BEGIN_AND_END_GENTLY add detail  
  this.alien think "Ich denke ich haue nun ab...", duration 1.0, bubbleFillColor LIGHT_GRAY add detail  
  this.alien moveTo this.myUFO, pathStyle BEE_LINE, animationStyle BEGIN_AND_END_GENTLY add detail  
  this.alien setOpacity 0.0, animationStyle BEGIN_AND_END_ABRUPTLY add detail  
  this.myUFO moveToward this.camera, 30.0, animationStyle BEGIN_AND_END_GENTLY add detail
```

Szenerie



UFO



Alien

# Minecraft Modding

## Eigenschaften

- Altersstufe: ab 10
- Minecraft modding aus einer Eclipse Umgebung
- Details unter
  - <http://www.minecraftforge.net/wiki/Installation/Source/de>
  - <https://java4kids.java.net/minecraft-workshop/aug2013/index.html>

## Lehrkonzept

- Programmierung unter Verwendung einer IDE
- Programmierung: Code verändern und übersetzen (mit Eclipse)





# Minecraft Modding





# Workshops für die Kinder: Robotik



# Lego Mindstorm

## Eigenschaften

- Altersstufe: 10-15
- Software und Hardware
- Programmierbare Roboter
- Zubehör
  - CPU
  - Motoren, Sensoren und Kabel
  - GUI für Programmierung

## Lehrkonzept

- Ähnlich wie Scratch
- Roboter sind Computer
- Echtzeitsysteme
- Ereignisbehandlung:  
Auslesen von Sensordaten  
und Ausgleichsreaktionen







# LEGO Mindstorm

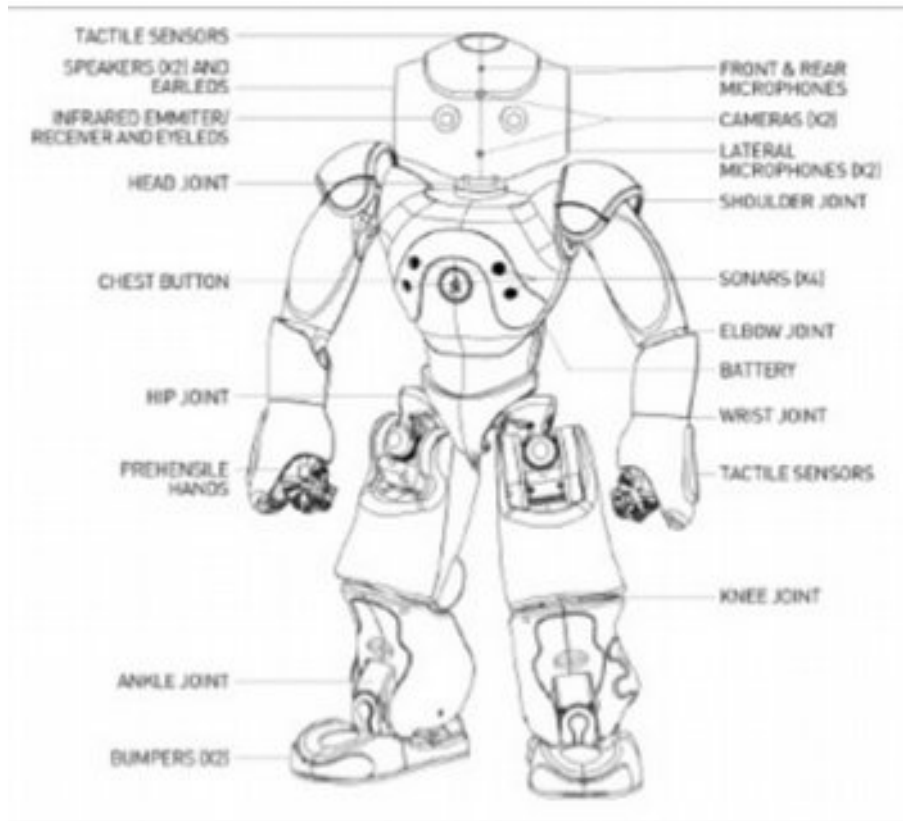




# NAO Robot

## Eigenschaften

- Altersstufe: ab 10
- sehr teuer: ca. 16000 \$



## Lehrkonzept

- Ähnlich zu Lego Mindstorm
- Bibliotheken für wiederverwendbare Komponenten
- Interaktion mit Webservices
- Vielfältige Funktionalität

# DEVOXX™ [4KIDS] NAO Robot







# Workshops für die Kinder: IoT und Elektronik



# Raspberry Pi und Java

## Eigenschaften

- Altersstufe: ab 10
- Zusammenbau des Computers
- Projekt: *Mary had a little Lambda* (Programmieren mit Java 8 und JavaFX)

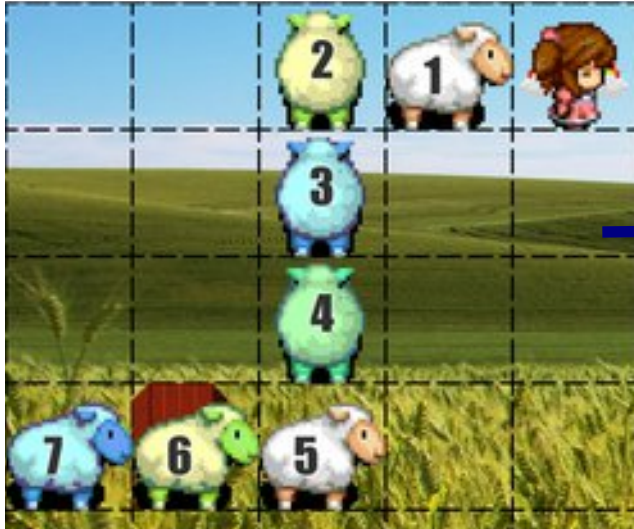
Spiel geschrieben von  
Stephen Chin (@steveonjava)  
[https://github.com/steveonjava/  
MaryHadALittleLambda](https://github.com/steveonjava/MaryHadALittleLambda)

## Lehrkonzept

- Wie Scratch und Arduino
- Programmierung: Code verändern und übersetzen
- Konzepte der Programmierung



# Mary had a little Lambda



Ströme, Filter und  
Lambda Ausdrücke

```
s.getAnimals().stream()
    .filter(a -> a.getNumber() % 4 == 2)
    .forEach(a -> a.setColor(Color.YELLOW));

s.getAnimals().stream()
    .filter(a -> a.getNumber() % 4 == 3)
    .forEach(a -> a.setColor(Color.CYAN));

s.getAnimals().stream()
    .filter(a -> a.getNumber() % 4 == 0)
    .forEach(a -> a.setColor(Color.GREEN));
```



Maps

```
Double mealSize = shepherd.getAnimals()
    .stream()
    .map(SpriteView::getScaleX)
    .reduce(0.0, Double::sum);

setScaleX(getScaleX() + mealSize * .2);
setScaleY(getScaleY() + mealSize * .2);

shepherd.getAnimals().clear();
```



# Sonic-Pi

## Eigenschaften

- Altersstufe: ab 10
- 'Audio-Programmierung'
- Computer reagiert sofort mit hörbaren Effekten
- Einfacher Programmcode

```
10.times do
  if rand < 0.5
    play 37
  else
    play 49
  end
  sleep 2
end
```

## Lehrkonzept

- Grundlagen der Programmierung
- Einfache Steuerstrukturen:
  - play, sleep, if-else, times, rand
  - einfache Zuweisungen und Variablenänderungen
- Programmänderungen geben sofort hörbare Reaktionen

```

play 60    play 65    play 63
sleep 0.3  sleep 0.3  sleep 0.3
play 61    play 64    play 63
sleep 0.3  sleep 0.9  sleep 0.3
play 62    play 65    play 62
sleep 0.3  sleep 0.3  sleep 0.6
play 63    play 65    play 62
sleep 0.3  sleep 0.3  sleep 0.5
play 64    play 65    play 64
sleep 0.6  sleep 0.3  sleep 0.3
play 64    play 65    play 64
sleep 0.6  sleep 0.3  sleep 0.3
play 65    play 64    play 64
sleep 0.3  sleep 0.6  sleep 0.3
play 65    play 63    play 64
sleep 0.3  sleep 0.3  sleep 0.3
play 65    play 63    play 60
sleep 0.3  sleep 0.3

```

(„Alle meine Entchen“)

```

10.times do
  play 60
  sleep 0.5
  play 65
  sleep 0.5
  play 72
  sleep 0.5
  if rand < 0.5
    play 37
  else
    play 49
  end
  sleep 1
end

```

```

i = 0
10.times do
  play 60+i
  sleep 0.3
  play 65+i
  sleep 0.3
  play 72+i
  sleep 0.3
  if rand < 0.5
    play 55
  else
    play 90
  end
  sleep 0.5
  i=i+1
end

```



## Eigenschaften

- Altersstufe: ab 10
- Open Source Electronics Prototyping Plattform
- Vielzahl an elektronischen Komponenten erhältlich

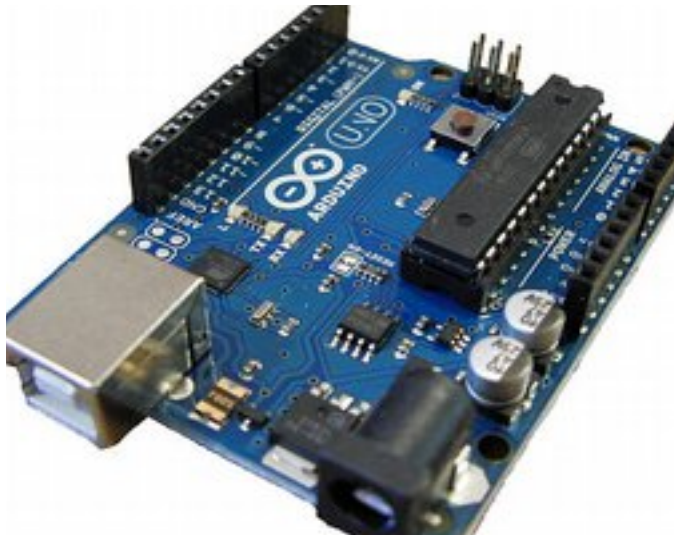


## Lehrkonzept

- Einführung in die Elektronik
- Computer bestehen aus elektronischen Komponenten
- Ohm'sches Gesetz
- Low-level Programmierung



# Arduino



Prozessorboard



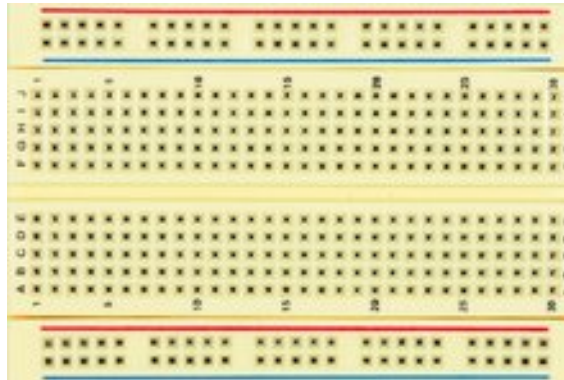
USB-Kabel



LEDs



Kabel



Steckplatine  
(Breadboard)



Widerstände



# DEVOXX™ [4KIDS]

# Diskussion

Fragen?

Bemerkungen?

Anregungen?





# Vielen Dank für´s Zuhören.

Katja Arrasz-Schepanski

@KatjaArrasz

Christian Mennerich

@cmennerich

**synyx GmbH & Co. KG**

Karlstraße 68

76761 Karlsruhe

[www.synyx.de](http://www.synyx.de)