



# **DENKEN VERSTEHEN LERNEN**

## **Computational Thinking in der Grundschule**

Grundschulpraktikum (B.Ed. und B.Sc.)

26./27.10.2016



# Einführung

Hintergrund, Computational Thinking, Unser Projekt



## Hintergrund

- aktuelle Situation: geringes Interesse an Informatik
  - wenig Studierende in derartigen Fächern
  - großes Missverhältnis zur Nachfrage an Absolventen
- mögliche Ursache: falsches Bild von der Informatik
  - sehr technologiezentriert
  - Vorstellung vom „Hacker“
  - fehlendes Wissen um grundlegende Inhalte



# Computational Thinking

- „Computational Thinking“ = für die Informatik wesentliche Denkprozesse
  - Begriff geprägt von Seymour Papert und Jeanette Wing
  - Kontrast: technologische Realisierung

## Computational Thinking

„[Die] Fähigkeit, Probleme durch Zerlegung, Modellierung, Abstraktion, Generalisierung, ... zu lösen“ (J. Wing)



# Computational Thinking an Schulen

- im Ausland gibt es viele Bestrebungen, Informatikunterricht attraktiver zu machen
  - Computing at School (UK)
  - code.org (US)
  - ...
- dabei wesentliche Rolle des Computational Thinking-Ansatzes
- **nicht** die **Artefakte** (Quellcode etc.) liegen im Vordergrund, ...
- ..., sondern **wie** Probleme gelöst werden können
- Ziel: Transformation des Informatikunterrichts hin zur Vermittlung von Kompetenzen zur Problemlösung



## Unser Projekt

- unser Plan: einen Computational Thinking-Kurs an einigen Grundschulen in der Umgebung anbieten
  - Curriculum auf Basis von [code.org](https://code.org)'s Course 2
  - gerichtet an SchülerInnen der Klassen 3-4
  - unterrichten werden Teilnehmer des Praktikums (= Sie)
  - wir haben 2 Grundschulen für die Teilnahme gewinnen können
- unsere Ziele:
  1. den SchülerInnen einen interessanten Kurs bieten, an dem sie Spaß und auch im Alltag einen Nutzen haben
  2. einen kleinen Beitrag für ein besseres Verständnis von Informatik in der Gesellschaft leisten
  3. langfristig: Überzeugungsarbeit für die Veränderung des bestehenden Informatikunterrichts in D betreiben



# Organisatorisches

Anrechnung, Ablauf, Prüfungsleistungen



# Anrechnung

- das Praktikum kann als Teamprojekt angerechnet werden (Modul INF2110)
- 9 ECTS für das Modul
- **wichtig:** zum Bestehen des Moduls ist zusätzlich notwendig, die Vorlesung „Software Engineering“ zu hören und zu bestehen





## Ablauf

- diese Woche: Einteilung auf die Schulen, in 2er/3er-Teams
- Oktober - Dezember 2016: Workshops
  - 6 Wochen (+2 Wochen Puffer), 2 Stunden pro Woche
  - Vorstellung des Curriculums
  - Diskussion von Möglichkeiten zur praktischen Umsetzung im Unterricht
  - Einbringen eigener Ideen ins Curriculum
- während dieser Zeit: Treffen mit der jeweiligen Schulleitung
- danach: Unterricht an den Schulen, Zeitrahmen schulabhängig
  - wöchentliches Reflektionstreffen
- Anfertigung eines Abschlussberichts



## Zeitlicher Rahmen des Unterrichts

- GS Wanne: Januar - März 2017 (KW 2-8 & 10-11),  
1 Doppelstunde (2\*45 Min.) pro Woche
- Hügelschule: KW 5-8 & 10-29/30,  
1 Stunde (45 Min.) pro Woche



# Prüfungsleistungen

- Einbringen eigener Ideen während der Workshops (25%)
- Beteiligung an den wöchentlichen Reflektionstreffen während der Unterrichtsphase (25%)
- Abschlussbericht (50%)
  - diesem kann auch ein Artefakt wie z.B. der Quellcode eines Lernspiels beigefügt sein, das im Rahmen des Praktikums entstanden ist



## Nächste Schritte

- Finden Sie sich in 2er/3er-Teams zusammen.
- Entscheiden Sie sich für eine der Schulen.
- Lesen Sie das Lehrerskript für die ersten beiden Unterrichtseinheiten, um sich auf das nächste Workshop-Treffen in 2 Wochen vorzubereiten.



# Fragen...?



# Danke.

Kontakt:

**Julian Jabs**

B221

Sand 13, 72076 Tübingen

[julian.jabs@uni-tuebingen.de](mailto:julian.jabs@uni-tuebingen.de)