
# Visually Explaining Source Code in CS Education

**Benjamin Biegel, Sebastian Baltes, Bob Prevos, and Stephan Diehl**

**University of Trier, Germany**

@s_baltes

s.baltes@uni-trier.de

IEEE Symposium on
Visual Languages and Human-Centric Computing
October, 18-22 – Atlanta, USA

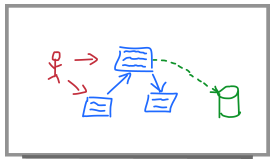
VL/HCC

# Motivation

- In CS education, teachers have to present concrete source code examples and related abstract concepts
- Traditional presentation tools have shortcomings:

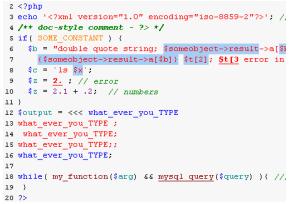
**PowerPoint**

- Easy-to-use drawing features (shapes, colors, etc.)
- But: Linear, predefined presentation
- Interaction with audience?
- Source code formatting?



**Whiteboards**

- Flexible, but depending on drawing skills of teacher
- Difficult to prepare or modify content
- Not suitable for longer source code examples



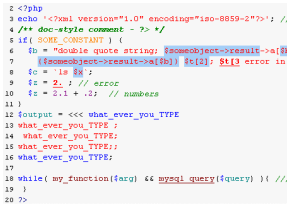
**Code Editors / IDEs**

- Exploration and modification of source code possible
- But: Not designed for presenting source code (e.g. step-by-step revealing)
- Visualization difficult

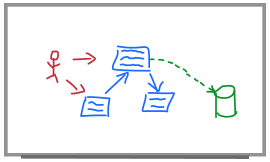
# Motivation

**Tradeoff:**

visually appealing but static ⟷ flexible but text-oriented



flexible but cumbersome



**Common scenario:**

Switching between PowerPoint and source code editor

**Our goal:** New tool for presenting source code combining:

- Drawing features of PowerPoint
- Flexibility of source code editors
- Interactivity of whiteboards

# Our Approach

# Requirements

**Preparation** 

- Create content in advance
- Prepared content can be reused

**Presentation** 

- Step-by-step revealing of visuals and code
- Highlighting during lecture

**Follow-up** 


- Content available to students immediately after lecture

**Flexibility** 

- Modifying and adding content during the lecture

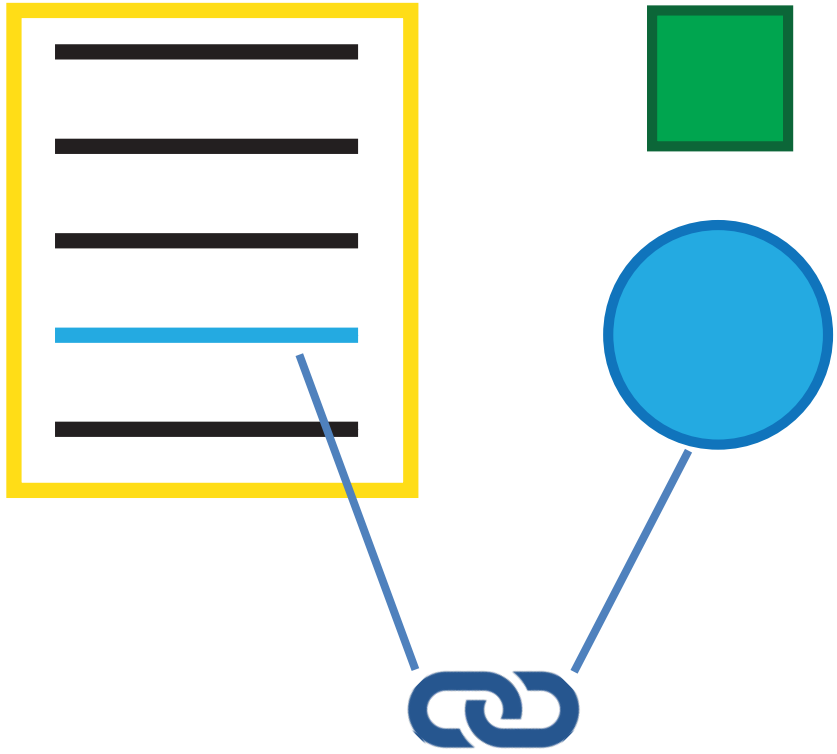
# Vision

- Digital canvas with source code editor
- Teacher defines beforehand in which order source code is revealed
- Drawing features similar to PowerPoint, but intended to be used also during lecture
- Drawings and code can be linked
  → Position is updated if code is inserted
- Highlighting features for both source code and drawings
- Presentation can be saved and shared any time during the lecture
- Students can branch documents for own annotations

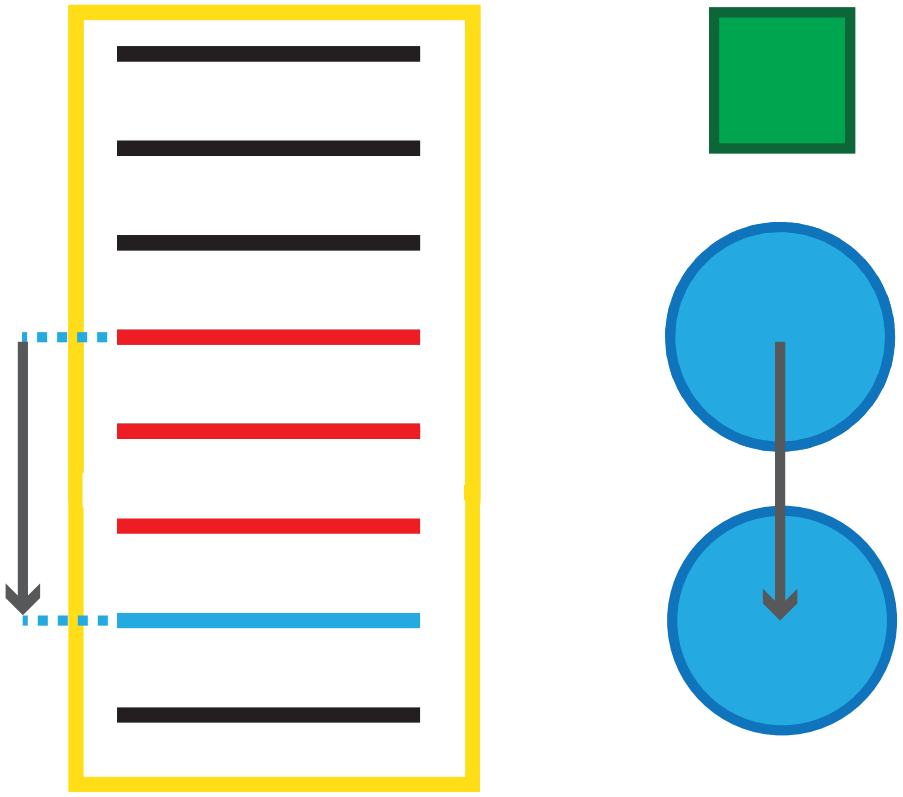
# Vision

- Digital canvas with source code editor
- Teacher defines beforehand in which order source code is revealed
- Drawing features similar to PowerPoint, but intended to be used also during lecture
- **Drawings and code can be linked**
  **→ Position is updated if code is inserted**
- Highlighting features for both source code and drawings
- Presentation can be saved and shared any time during the lecture
- Students can branch documents for own annotations

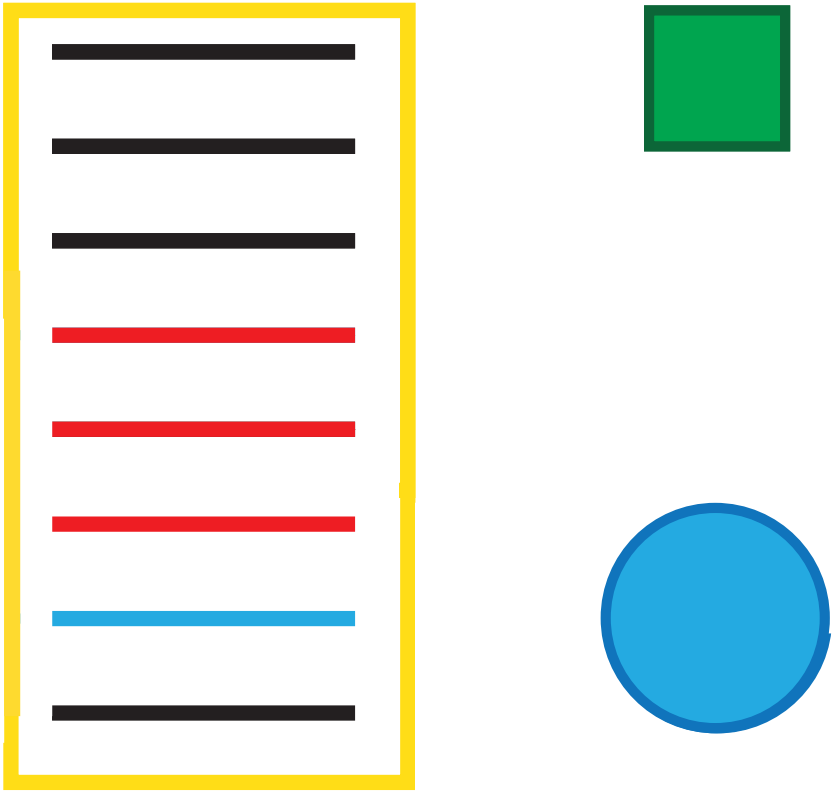
# Linking of Code and Drawings



# Linking of Code and Drawings



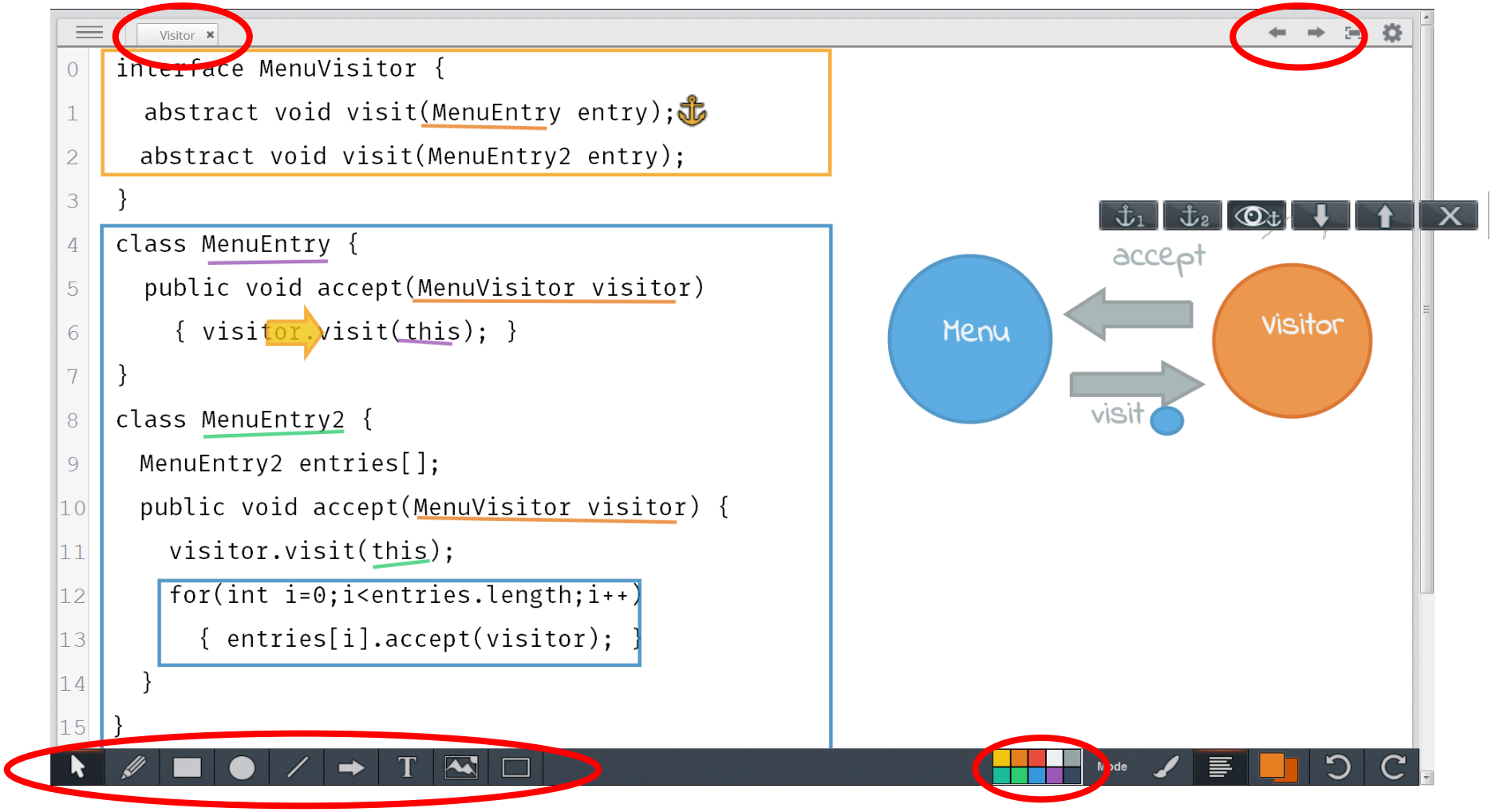
# Linking of Code and Drawings



# Prototype Implementation

# Prototype Implementation

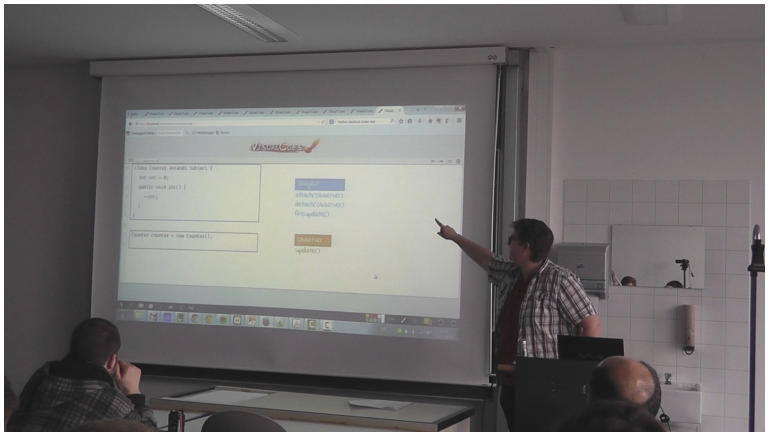
- Web-based
- Runs on both desktop and tablet browsers (Firefox and Safari)
- Source code “animation” using XML file



# Practical Experiences

# Practical Experiences

**Evaluation in two undergraduate lectures:**

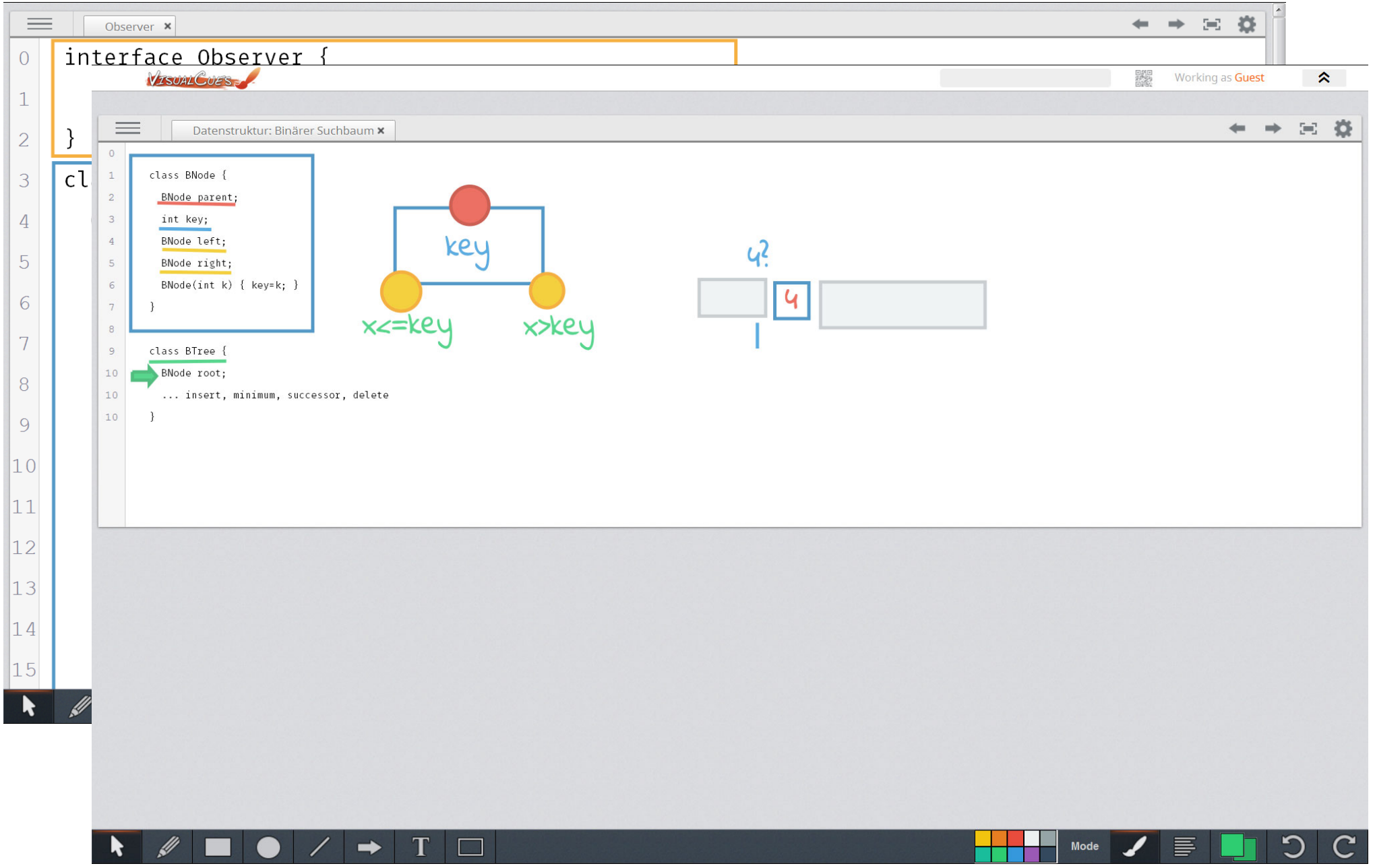


1. Software engineering (design patterns)
   - Interviewed three students and two teachers

2. Programming concepts (binary search tree implementation)
   - Interviewed three students and two teachers
   - Questionnaire

We used an ad-hoc visual notation, often utilizing color to indicate relation:

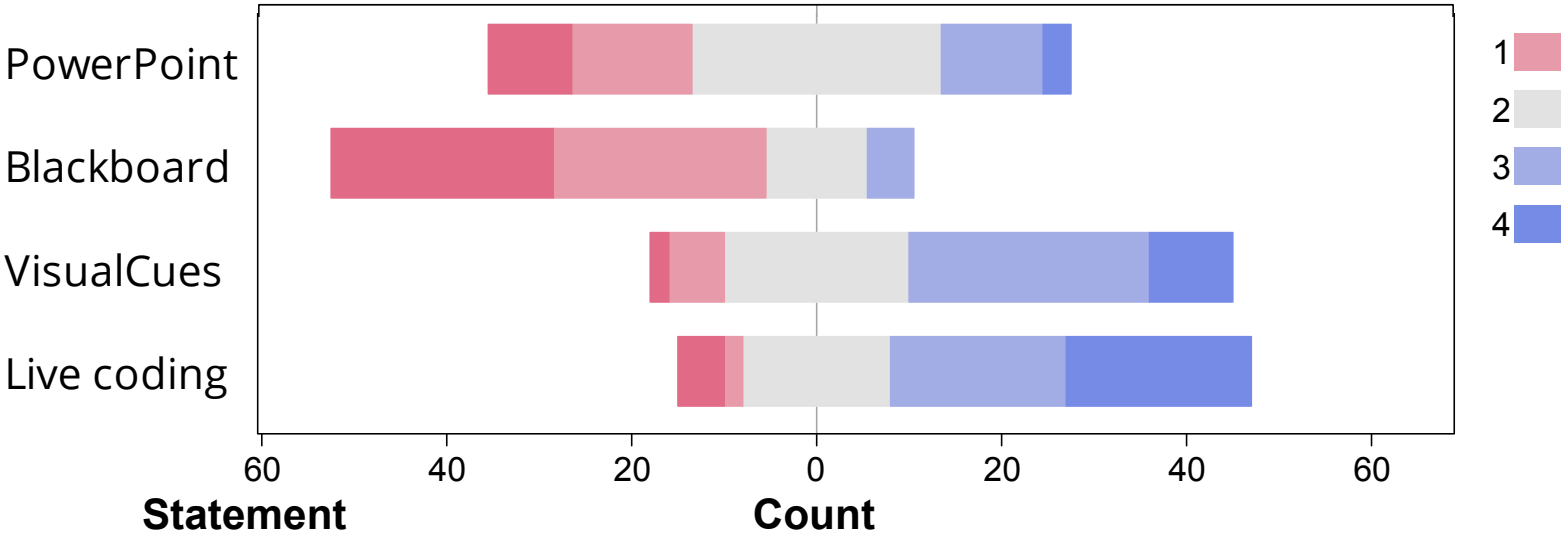
# Practical Experiences



# Students' View

- Ad-hoc notation understandable
- Appreciated step-by-step revealing of source code and linking
- Liked that teacher can immediately respond to questions
- *"VisualCues prevents the teacher from doing PowerPoint karaoke."*
- *Questionnaire:*

"For future lectures, I would like source code examples to be presented mainly using..."




n=63
Likert scale items ranging from 0="I do not agree" to 4="I agree"

# Teachers' View

- 50% of drawings prepared, 50% created ad-hoc during lecture
- Often framed code with colored rectangle and used same color for corresponding drawings
- Tried to used shapes consistently
- Better support for planning the lecture requested

# Conclusion and Future Work

- Students not happy with status quo
  (presenting source code with PowerPoint)
- VisualCues was well received
- But also strong preference for live coding

- **Future work:**
  - Combine VisualCues and live coding
  - Evaluation in larger context
  - Make VisualCues also available for students during lecture?

Demo video and supplementary material:

**`http://st.uni-trier.de/visualcues`**

@s_baltes

s.baltes@uni-trier.de