

# CodingDojo: Interactive Slides with Real-time Feedback

Felix Raab<sup>1</sup>, Markus Fuchs<sup>2</sup>, Christian Wolff<sup>1</sup>

Lehrstuhl für Medieninformatik, Universität Regensburg<sup>1</sup>

Lehrstuhl für Informationswissenschaft, Universität Regensburg<sup>2</sup>

## Abstract

We present "*CodingDojo*", a presentation platform with interactive tools and real-time feedback elements that are controlled by a smartphone to be used in typical academic teaching contexts. The system provides tools for source code presentation, discussion and different types of coding challenges. *CodingDojo*<sup>1</sup> was evaluated in a University course about good coding practices where traditional presentation platforms would be limited in functionality. We received positive user experience rankings and feedback from participants, suggesting that the system should be further developed and tested in other courses.

## 1 Introduction

In a University course about good coding practices, we introduced our new presentation platform "*CodingDojo*". As traditional presentation software like *Powerpoint* is missing important features for presenting and discussing source code, we have created a system that provides interactive tools for that purpose. We considered it important that course participants could freely express their opinion about the readability of code examples. Therefore we have included elements of student feedback systems which have proved to enhance the quantity and quality of discussions in courses (Kay & LeSage 2009).

Relevant literature about clean code and refactoring, such as *Code Complete* (McConnell 2004), *Refactoring* (Fowler 1999) or *Clean Code* (Martin 2008), served as guidelines for course contents and code examples. Using the *code smell metaphor* (Fowler 1999) as a common theme throughout the course, we have designed custom icons and playing cards for each smell (see fig. 2). Students could use their personal card for authentication and gave short presentations about their code smell during the course. The tag on the back of each card can be recognized by appropriate devices for future tangible interaction applications.

---

<sup>1</sup> „Dojo“ (jap. 道場) is a Japanese term for (martial arts) training halls, cf. Wikipedia (2012).

## 2 The CodingDojo Presentation Platform

*CodingDojo* is an integrated presentation platform with real-time feedback visualisations, interactive coding challenges as *gamification* elements and a smartphone controller. Slide synchronisation between teacher and students is one of the key features. Students can either control the deck on their own or synchronize slide changes and free-hand annotations with the lecturer. Users can also take notes on a slide or anonymously ask questions that are communicated via a smartphone. The mobile application enables the lecturer to: (1) switch slides with simple slide gestures, (2) start *challenges* (i. e. tasks the students have to master), (3) switch through different challenge states and (4) look at all student questions. In addition, the system logs various events (e. g. slide changes, browser tab switches) that can be further explored with visual analysis tools.

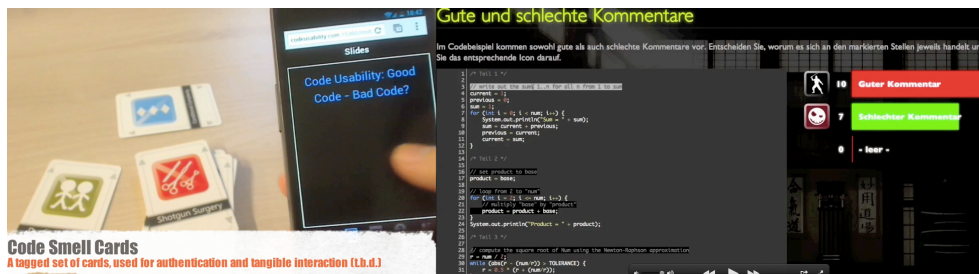


Figure 1: Left: Code smell cards and smartphone controller. Right: Interactive result visualisation of a challenge

The system provides three different types of challenges, aiming at engaging students with interactive exercises and collecting data about their performance. In our course, most challenges involved identification of good code, bad code, specific code smells, or refactorings. Students can answer either by selecting a number of given choices (*single- and multiple-choice challenge*) or by using drag and drop-operations to mark different parts of a code example with code smells or refactorings (*magnets challenge*). Moreover, students could improve existing code by writing into an embedded editor (*refactoring challenge*).

## 3 Evaluation

The system was used in the context of a "Code Usability" course given during the winter term 2011/2012. Participating students came from applied computer science curricula and had basic knowledge of object-oriented programming (Java) as well as some experience in application development. During the course, both quantitative as well as qualitative methods were applied in order to evaluate different aspects of *CodingDojo*. On the one hand, we were interested in how participants experienced the interaction with the system (see ch. 3.1 below). On the other hand, we collected data related to actual course contents and student performance concerning *code usability* (ch. 3.2).

### 3.1 User Experience

We used the *User Experience Questionnaire* (UEQ, Laugwitz et al. 2008), to measure the overall user experience. The UEQ takes six different factors of interactive software products into account. Figure 2 shows accumulated results for 15 participants. The reason for the low dependability value may be largely due to bugs that sometimes forced students to restart the application or repeat their previous action. The high rankings for attractiveness, stimulation and novelty confirm informal feedback during the course to the effect that students perceived the system as interesting and stimulating.



Figure 2: Left: Results of the User Experience Questionnaire (UEQ). Right: Examples from the CodingDojo Card Deck with Code Smell Icons (front) and explanations and 2D codes (back)

### 3.2 Recognition of Clean Code

Analysis of the initial challenges gives some indication to what extent students are able to recognise good and bad coding practices *without prior training*. We examined the results of 21 challenges that students were given on the first day of the course: (1) Most participants did not recognise the smell “Duplicate Code”; (2) Participants could not clearly identify good and bad names; (3) Only few participants could clearly recognise the “Comments” smell; (4) Most participants correctly recognised bad code formatting and aesthetics.

### 3.3 Student Questions

In total, this feature was used 20 times over the course of four days. The questions asked by the students can be classified under the following topics: System features, course structure, course contents, and housekeeping. Students also reported corrections and once, it was used for an ad-hoc poll. Sometimes we overlooked notifications on the smartphone and therefore could not address all questions in time.

### 3.4 Questionnaire

At the end of the course, all 15 participants were asked to fill out a short questionnaire: In our questionnaire, participants either had to choose from a list of options or assess how strongly they agreed or disagreed with a set of statements. All students agreed with the

statement that the challenges helped them to better comprehend the course contents. More than 60% agreed that the system provided valuable feedback about their comprehension rate and that the system helped them to stay more attentive throughout the lecture. More than 90% would like to use the system in other courses. 60% of the students liked the magnets challenge best and 80% perceived the refactoring challenge as the most difficult. 60% of the participants regarded the challenge frequency as appropriate.

As far as suggestions for improvements are concerned, participants requested most frequently that they could look at their challenge answers again. Furthermore, they requested high score lists where they could see their performance in relation to others. Other participants wished that they could access challenges outside of the lecture context or see questions asked from other course participants.

## 4 Conclusion

*CodingDojo* is an interactive presentation platform that we evaluated during a University seminar called “Code Usability”. The system provides tasks as gamification elements and can be controlled by a smartphone. Although we received positive user experience rankings and feedback from participants, we identified some limitations and areas for improvements: For example, notifications for new questions should be augmented with haptic feedback on the smartphone. Furthermore, students should be provided with features to access all challenge results and respond to questions from other participants. As for future versions, we would like to test new smartphone applications in lecture rooms and to evaluate tools for teaching good coding practices on interactive devices such as interactive tabletops or tablets.

### Acknowledgements

We would like to thank Andreas Preis for designing the code smell icons.

### References

- Fowler, M. (1999). *Refactoring: improving the design of existing code*. Reading, MA: Addison-Wesley.
- Kay, R. H., & LeSage, A. (2009). Examining the benefits and challenges of using audience response systems: A review of the literature. *Computers & Education*, 53(3), 819-827. Elsevier Ltd.
- Laugwitz, B., Held, T., Schrepp, M. (2008). Construction and Evaluation of a User Experience Questionnaire. In Holzinger, A. (Hrsg.): *Proceedings of the 4th Symposium of the Workgroup Human-Computer Interaction and Usability Engineering of the Austrian Computer Society on HCI and Usability for Education and Work (USAB '08)*. Berlin: Springer-Verlag, S. 63-76.
- Martin, R. C. (2008). *Clean Code: A Handbook of Agile Software Craftsmanship*. Upper Saddle River: Prentice Hall.
- McConnell, S. (2006). *Code Complete: A Practical Handbook of Software Construction*. Redmond, Wash.: Microsoft Press.
- Zichermann, G.; Cunningham, Ch. (2011). *Gamification by Design. Implementing Game Mechanics in Web and Mobile Apps*. Sebastopol, CA: O'Reilly.