

Using Coding Dojos to Support Learning in an Undergraduate Computing Programme

Stephen Naicken
African Leadership University
Pamplemousses, Mauritius
snaicken@alueducation.com

Mehdi Oulmakki
African Leadership University
Pamplemousses, Mauritius
moulmakki@alueducation.com

James H. Paterson
Glasgow Caledonian University
Glasgow, UK
James.Paterson@gcu.ac.uk

ABSTRACT

This paper describes the use of coding dojo techniques to support students in both developing programming skills and good professional habits through collaborative practice. Following the Randori dojo format, coding dojo sessions have been implemented with undergraduate students as early as their first programming course. The students have found the sessions engaging, enjoyable and productive in supporting their learning, and preparing them for interviews and internships.

CCS CONCEPTS

• **Social and professional topics** → **Computer science education**;

KEYWORDS

Coding dojos, pair programming, learning programming

ACM Reference Format:

Stephen Naicken, Mehdi Oulmakki, and James H. Paterson. 2018. Using Coding Dojos to Support Learning in an Undergraduate Computing Programme. In *Proceedings of ACM Woodstock conference (ITiCSE'18)*. ACM, New York, NY, USA, Article 4, 2 pages. https://doi.org/XX.XXX/XXX_X

1 INTRODUCTION

Learning in introductory programming courses can be assessed in a variety of ways, including exams and practical assignments. It has been reported that the likelihood of success in these assessments, even in exams, is enhanced by developing programming skill through extensive practice in programming [5]. There is also evidence that learning in programming may be enhanced by the use of collaborative activities such as pair programming [4].

One form of learning activity which can combine practice with collaboration is the coding dojo, introduced by Bossavit and Gaillot[3], which is based on the ideas used in martial art, where dojo is a term used to refer to a place of formal training. As noted by Rooksby et al. [7], the idea of the dojo is closely associated with kata, another martial arts term that refers to exercises repeated in order to gain mastery, but has a social and collaborative aspect.

Coding dojos have been widely discussed and promoted as a method of learning for both novice and professional developers.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

ITiCSE'18, July 2-4 2018, Larnaca, Cyprus

© 2018 Copyright held by the owner/author(s).

ACM ISBN XXX-XXX-XX-XXX/XX/XX...\$15.00

https://doi.org/XX.XXX/XXX_X

There is relatively little evidence of their use in formal education environments, although Lee et al.[6] report improvements in student performance in learning test-driven development (TDD) when students practiced in dojos - these students also enjoyed the dojos as non-competitive environments for learning. We are interested in the use of coding dojos in a way that is designed to integrate with the content and pedagogy of an undergraduate Computing programme, and have implemented dojos to support learning on a range of modules within our programme.

2 IMPLEMENTATION OF THE CODING DOJOS

The African Leadership University focuses on the development of skills such as leadership, critical thinking, problem solving, quantitative reasoning, communication and teamwork, and delivers a Computing degree programme, in partnership with Glasgow Caledonian University, at its Mauritius campus to students who come from over 40 countries in Africa. Peer instruction, collaborative activities and the flipped classroom form a key part of the learning approach. Within this context we decided from the start to adopt the idea of coding dojos to support learning on programming and related modules.

2.1 The format of the session

There are a number of formats that can be used for coding dojos [7]. The model that we have implemented is a variation on the Randori format, as described by Aniche and Silveira [1] and Bache [2]. The students work in small groups typically of around 3-6 members, trying to solve a problem together, in time-boxed rounds each lasting 5 minutes. There is a facilitator present who manages the timekeeping and is able to interact with participants to offer hints and advice where appropriate, although we have observed good participation and submissions in purely student-led sessions. The Randori format involves assigning students to one of three roles:

- **The pilot:** The pilot is the only person allowed to interact with the computer in use. The responsibility of the pilot is to move the team closer to a correct, well written solution. While this mostly involve coding, pilots are encourage to perform relevant searches, write pseudocode, as well as debug and test existing code.
- **The co-pilot:** The co-pilot supports the pilot's effort, following best practices of pair programming. The responsibility of the co-pilot is to double check the work being done by the pilot, challenge assumptions and offer suggestions that are relevant to solving the problem at hand. Moreover, the

co-pilot moderates interactions between the pilot and the remaining students.

- **The audience:** The remaining group members form an audience who may provide comments and suggestions at specific points. In the rules stated by Bache, it is assumed that a TDD process is being followed, and the audience are allowed to contribute only when tests are passing, although we do not necessarily follow this rule as TDD is often not the focus of the learning activity, and the students are not initially familiar with the technique.

The essence of the exercise lies in the students shifting roles at the end of each time boxed round. The pilot joins the audience, the co-pilot becomes the pilot, and a member of the audience steps up to be the new co-pilot.

Initially, dojos began with a single pilot, a co-pilot and the rest of the participants as the audience, as is typically the case when undertaken by professionals. However, in this setup, we noticed that the audience would fragment into small groups that would work on the problem. Eventually, they chose to self-organize into smaller groups, each executing the dojo pattern and solving the same problem independently. The benefit of this was more coding time (i.e. more time as the pilot) for the students, which given their stage of development is of importance.

2.2 The problem assigned to the participants

Aniche and Silveira state that the goal is not to solve the problem, but to share practice and knowledge. In that case the nature of the problem is not particularly important. However, our dojos are integrated with the programme of study, although not necessarily with a single module on the programme, so that problems are selected to make use of specific knowledge that the students have learned at that point in time, and to provide meaningful practice in applying that knowledge. To date problems have been related to programming in Java and web development with JavaScript. The dojo problems relate to the content of those modules in one of two ways: The first and most common is by directly reinforcing one of the key outcome of the week's material, for example tackling an inheritance-based problem set after the concept is introduced. Alternatively, dojos can be used to explore concepts or libraries that were mentioned briefly in the material. In this way, dojos allow us to add more depth or breath to the planned instruction where appropriate.

3 STUDENT ENGAGEMENT AND EXPERIENCE

The coding dojos are arranged in addition to the activities the students are expected to undertake on their class timetable. While there are no formal lectures, students attend scheduled classes which are typically flipped classroom sessions, in advance of which they study a range of learning materials. In programming and related modules there are also typically practical tasks and assignments, some of which are graded. Attendance at dojos is encouraged but is not mandatory, and there is no academic credit available for participating. The level of participation in the dojos is generally encouraging. It is clear that the students enjoy the sessions and see the value of

these for their learning. Feedback from student evaluations consistently identifies a desire for meaningful practical work, and the value of the dojos is noted, for example:

The most useful or enjoyable were the coding dojo sessions that we had. Were given coding challenges to solve in groups of two or more and then present our solutions for them. It was productive and encouraging for us to study and learn how to code.

We will do further evaluation of the coding dojos to determine whether there is an identifiable impact on learning evidenced through students' performance in the formal assessments, as well as their pursuit and performance during technical internships.

4 CONCLUSIONS

The idea of the coding dojo has been adapted to work within a formal educational environment to provide additional opportunities for students to develop skills in programming through practice in collaboration with peers. The dojo sessions are enjoyed by the students who engage willingly with these despite the fact that they are not a mandatory part of their study and it is clear that these are considered to be productive activities. The sessions we have implemented have been based on the Randori format, but it is worth noting that there are other formats for dojos which could be adopted in future. For example, in the Kata format [1] solutions to problems are prepared in advance by participants, and presented to and discussed by the audience during the session along with the steps taken to get to the solution. We conclude that the ideas advocated in the wider software development community for learning through coding dojos can make a valuable contribution to learning and engagement within formal education.

REFERENCES

- [1] Mauricio Finavaro Aniche and Guilherme de Azevedo Silveira. 2011. Increasing learning in an agile environment: Lessons learned in an agile team. In *Agile Conference (AGILE)*, 2011. IEEE, 289–295.
- [2] Emily Bache. 2013. *The Coding Dojo Handbook*. Emily Bache, Canada (2013).
- [3] Laurent Bossavit and Emmanuel Gaillot. 2005. The coder's dojo—a different way to teach and learn programming. In *International Conference on Extreme Programming and Agile Processes in Software Engineering*. Springer, 290–291.
- [4] Brian Hanks, Sue Fitzgerald, Renée McCauley, Laurie Murphy, and Carol Zander. 2011. Pair programming in education: A literature review. *Computer Science Education* 21, 2 (2011), 135–173.
- [5] Marko Hassinen and Hannu Mäyrä. 2006. Learning programming by programming: a case study. In *Proceedings of the 6th Baltic Sea conference on Computing education research: Koli Calling 2006*. ACM, 117–119.
- [6] Young Lee, Dhanunjaya Bhargavan Marepalli, and Jeong Yang. 2017. Teaching test-drive development using Dojo. *Journal of Computing Sciences in Colleges* 32, 4 (2017), 106–112.
- [7] John Rooksby, Johanna Hunt, and Xiaofeng Wang. 2014. The theory and practice of randori coding dojos. In *International Conference on Agile Software Development*. Springer, 251–259.