

Who are with us: MOOC learners on a FutureLearn course

Article

Accepted Version

Liyanagunawardena, T. R., Lundqvist, K. O. and Williams, S. A. (2015) Who are with us: MOOC learners on a FutureLearn course. *British Journal of Educational Technology*, 46 (3). pp. 557-569. ISSN 1467-8535 doi: <https://doi.org/10.1111/bjet.12261> (Special issue on MOOCs) Available at <https://centaur.reading.ac.uk/39204/>

It is advisable to refer to the publisher's version if you intend to cite from the work. See [Guidance on citing](#).

To link to this article DOI: <http://dx.doi.org/10.1111/bjet.12261>

Publisher: Wiley

All outputs in CentAUR are protected by Intellectual Property Rights law, including copyright law. Copyright and IPR is retained by the creators or other copyright holders. Terms and conditions for use of this material are defined in the [End User Agreement](#).

www.reading.ac.uk/centaur

CentAUR

Central Archive at the University of Reading

Reading's research outputs online



Who are with us: MOOC learners on a FutureLearn course

Abstract

Massive Open Online Courses (MOOCs) attract learners with a variety of backgrounds. Engaging them using game development was trialled in a beginner's programming course, "Begin programming: build your first mobile game", on FutureLearn platform. The course has completed two iterations: first in autumn 2013 and second in spring 2014 with thousands of participants. This paper explores the characteristics of learner groups attracted by these two consecutive runs of the course and their perceptions of the course using pre- and post-course surveys. Recommendations for practitioners are offered, including when the audience is different to the one expected. A MOOC is unlikely to please everyone, especially with such large cohorts. Nevertheless, this course, using game development as a vehicle to teach programming, seems to have offered a balanced learning experience to a diverse group of learners. However, MOOC creators and facilitators should accept that a course cannot be made to please everyone and try to communicate clearly who the intended audience for the course are.

Introduction

Teaching programming to beginners is challenging (Yan, 2009). When offering a Massive Open Online Course (MOOC) on computer programming, not only attracting students but also retaining those using innovative methods of teaching, is of great importance to establish a new generation of programmers. Though there are many MOOCs offered in programming, detailed statistics for those courses is not easily obtainable. The 'Computing For Teachers' (CFT) MOOC offered by the Department of Computer Science at the University of Warwick has published some of their course's statistics. According to Onah, Sinclair, & Boyatt (2014), there had been 552 enrolments in the course and initially 125 participants had taken part in the programming quizzes, which in 'session 5' had dropped to just 19. In this course they had also offered a tutor supported version for £100 fee. Out of the 15 participants who attempted the programming quizzes in 'Session 1' in this mode of study, only five remained in 'Session 5' showing that the attrition is high even in fee-paid option.

Although there is a growing popularity for eLearning courses, there is also a concern about the large dropout rates in comparison to traditional courses (Lee, Choi, & Kim, 2013). When it comes to MOOCs, the engagement contract is very different to that of conventional fee paying eLearning courses (Liyaganawardena, Parslow, & Williams, 2014) and makes it even difficult to retain students. Therefore the pedagogy adopted in programming courses, especially programming courses offered as MOOCs, requires careful consideration to facilitate effective learning. These courses have to consider providing additional incentives to keep learners motivated in these challenging courses. One such method could be to use games in programming courses.

Game-based learning has been used in teaching for years; for example as early as 1960s Logo was used to teach mathematical concepts to children (Feurzeig, 1969). Courses that teach introductory programming using games have reported positive outcomes; for example students highly regarding the use of games (Leutenegger and Edgington, 2007). Rajaravivarma (2005) claims that using games for teaching programming created "a passion to want to do more", which helps learning programming as it is a skill that needs developing through practice; while also helping to keep students interested and engaged

(Lorenzen and Hilman, 2002). Leutenegger, and Edgington (2007) show that their approach to teaching programming with game development had both increased enrolments and improved retention.

However, there is very little research on using games to teach programming in MOOCs. Unlike a college course, MOOCs attract a wide variety of learners from diverse backgrounds. Using a game to teach programming in such a course can be a challenge, especially due to computer system requirements and setting up development environments (Liyanagunawardena, Lundqvist, Micallef, and Williams, 2014). In this empirical case study, the authors examine two consecutive runs of the “Begin programming: build your first mobile game” (FLMobiGame) course to identify the learner groups attracted by this beginner level course and their perception of the course. The paper also offers recommendations to practitioners based on lessons learned, which will help in designing similar courses to an open audience.

Background

FLMobiGame was the first University of Reading MOOC. This was a seven week course expecting around three hours a week commitment from participants. The course provided an open source Android game framework developed by the second author, lead educator of the course, which was then used to teach basic concepts of programming by developing a game. In the first week participants downloaded and installed required software and set up the game framework on their computers. The game could then be run either on an emulator on the computer or on an Android mobile device. Weeks two to six introduced basic concepts of programming and each week the participants made changes in their games that built the concepts learnt up to that point. Week seven was dedicated to “consolidate, reflect and celebrate” the achievements. A demonstration of the FLMobiGame is available in Liyanagunawardena (2014).

When FLMobiGame was first offered in October 2013 (the first run), as one of the first courses on the FutureLearn platform (www.futurelearn.com), there was an initial cap of 10,000 placed on enrolments. The course proved popular and was fully subscribed within 24 hours of the launch. The second run of the course in February 2014 attracted some 38,000 enrolments.

The approach to teaching used in this course closely relates with the “maker culture” that “emphasises informal, networked, peer-led, and shared learning motivated by fun and self-fulfilment” (Sharples et al. 2013,p.33). The provided game framework gives learners a working mobile application (Liyanagunawardena, 2014). This framework allows the learners to make small changes in the program and “see” the effects on the screen. The course also encourages participants to experiment with the code. By providing updated versions of the program code required at the beginning of each week, participants are safe to experiment as they can always download the available working program code if things did not go as expected.

However, there were risks of using this approach as if one is unable to install and setup the software required, they would not be able to fully participate in the course. Due to the various hardware and software combinations this was considered a high risk. The course team minimized the risk by employing some seven upper year undergraduate mentors to support the course allowing extensive mentor presence in the course in the first week. Kay and McKlin (2014) in offering ‘Educational Robotics for Absolute Beginners’ MOOC have supported participants by offering extra help by way of an appendix “when something unexpected happens”. Similarly we developed a growing “Frequently Asked Questions”

document where we updated it every so often as required to include problems that were reported frequently.

Rationale for a MOOC

At the School of Systems Engineering, University of Reading, games are used in teaching with excellent outcomes (Lundqvist, 2013; McCrindle, 2013). The School wanted to showcase their excellent and innovative teaching to a global audience through a MOOC where anyone could participate at no cost (in terms of enrolment or tuition fees) as long as they can access the Internet and understand the language. Another compelling reason for employing a MOOC, which uses a game to introduce programming, was to inspire young learners to become “digital makers” possibly leading to knowledgeable learners joining degree courses at the University. The team also hoped to reach lifelong learners and leisure learners who may previously not have had an opportunity to explore programming. In designing the course, learner personas were used to clarify learners’ needs and expectations. The team identified the main target group for the MOOC as school pupils (13-18 years).

Methodology

All the authors were very much involved in the creation and facilitation of the course, spending tens of hours per week within the platform while the course was live. The first author was the lead facilitator for both instances of the course.

Data collection

FutureLearn administer a standard pre-course and a post-course survey (Appendix contains sample questions) for each course run on the platform. Course teams can add some four additional questions to these surveys. The pre and post course surveys are non-identical but some questions are posed in both surveys. The same set of surveys was used in both runs with slight changes to question responses (described in the text). Survey data are anonymised before being released to partner institutions. Anonymised full data sets for all four surveys were analysed. The number of responses for each survey is shown in Table 1.

<< Table 1 >>

Analysis

Pre-course Survey

Demographic details were captured only in the pre-course surveys. Analysing the two sets of pre-course survey data showed that the large majority of learners were males: 76% in the first run and 74% in the second run. The age distribution of participants is illustrated in Figure 1. Under 25s representation on the course has doubled from 14% to 28% from the first run to the second.

<<Figure 1 >>

The large majority who responded (72% in the first run and 67% in the second run) had a degree or higher level of education (Figure2).

<<Figure 2>>

Participants' perceived level of programming experience was captured under the categories: "complete beginner"; "have tried programming in the past"; "know some basics and have used one (or more) language(s)"; and "expert programmer". In the first run this question had a single answer but in the second run it was presented as a multiple answer question. Therefore, in the analysis, people who selected "expert programmer" together with other choices were considered expert programmers. In the first run 34% self-evaluated to be complete beginners as opposed to 32% in the second (Figure 3).

<<Figure 3>>

Post-course Survey

Only a small proportion of the participants had responded to the post-course surveys. The drop in the second run responses could be due to the post-course survey link being sent a week after the course had actually finished. As already mentioned, the post-course survey did not capture demographic data.

Question eight of the post-course survey asked "what previous experience, if any, do you have in this subject area?" with responses: "I studied it at school"; "I studied it at university"; "I work in a related field"; "I have taken other courses or classes in this subject"; "I have taken other MOOC(s) in this subject"; and "I have no previous experience". 19.5% in the first run and 21.9% in the second run have had no previous experience. At the time of writing, there was no means to link pre and post course surveys; therefore this question was used as a rough estimate to explore the progression of beginners in the course. Comparison of these results against pre-course question "what is your level of knowledge in computer programming?" is shown in Table 2.

<<Table 2>>

The participants were asked of their perception of the level of the course, the five options presented were: "much too basic"; "a bit too basic"; "about right"; "a bit too advanced"; and "much too advanced". For the purpose of analysis answers for first two options were combined into a category "basic" while the last two were combined to a category "advanced". Out of the responses, 51% in the first run and 32% in the second thought the level of the course was "just right".

<<Table 3>>

The results of cross tabulating perceived level of the course against previous experience of participants is given in Table 4.

<<Table 4>>

Answers to the question "do you consider you achieved your aims by participating in the Begin programming: Build your first mobile game course?" was presented in the first run post-course survey as multiple choice (with two options Yes and No) along with a free text field; however in the second run it

was presented only as free text. Free text for the answer was manually coded. In both runs, the large majority (70.0% and 71.8%) had achieved their aims in taking the course.

Cross tabulation of prior programming experience and the achievement of their aims in the course are presented in Table 5, which shows that both groups (learners with experience and without) have achieved their aims in the course.

<<Table 5>>

Table 6 presents cross tabulated data for perceived level of the course and the achievement of their aims in the course. The responses of the three learners who did not achieve their aims in February 2014 run and 22 learners who did not achieve their aims in October 2013 run, but thought the course was the right level were further examined. It was seen that the majority of the participants in this category had faced difficulties in running the game due to system requirements. Other reasons given were internet connectivity and download problems and expectations to have more in depth understanding of Android specific programming techniques, which could not be achieved in a beginner's programming course.

<<Table 6>>

Results and discussion

Participant demographics

In both runs of the course, approximately three quarters of the participants in the pre-course survey were males. This is not uncommon in MOOCs; for example, in the first 17 MITx and HarvardX courses only 29% of registrants were female (Ho et al., 2014). However, FutureLearn pre-course survey results (FutureLearn 2014) across 37 courses (as of May 2014) showed that the majority of learners (nearly 60%) on FutureLearn are females. Even in the courses categorised under "Science, Technology, Engineering and Maths", over half of the participants were female. Therefore, both runs of the course seem to be outliers with respect to FutureLearn courses. It would be interesting to see a comparison of demographics with other programming courses on FutureLearn (such as Creative Coding offered by Monash University) when data is available, as this would be a more like-for-like comparison.

The large majority of course participants (72% in the first and 67% in the second) had obtained a degree level or higher level of education; only 28% in the first and 33% in the second had secondary education and below. This is also in par with other observed MOOC demographics; for example only 33% of the registrants in MITx and HarvardX courses have had high school and below educational qualifications (Ho et al., 2014). On the other hand, on the FutureLearn platform as high as 78% of the participants have had a university degree or higher; that is only 22% did not have a degree level education. Both runs of FLMobiGame had attracted people with lower educational qualifications than the FutureLearn averages (likely to correlate with attracting younger participants).

Beginner course?

The course was pitched at beginner's level and the course description included the text:

<<Figure 4>>

However, only about a third of participants (pre-course survey) were beginners to programming. In the post-course survey, when rating the level of the course, people with experience showed a propensity to rate the course as “basic”.

The varying levels of skills in programming among participants had mixed outcomes in terms of course delivery. For example, when beginners were facing technical issues more experienced participants were able to offer help. However, when experienced programmers started modifying the game and sharing their program code, some beginners found it inspirational while some others felt intimidated. For example, comments below from beginners in the post-course survey suggested the frustrations they faced.

“.. the course was hijacked by experienced programmers and so beginners programming it wasn't...”

“I found I was reading other peoples comments, and found myself thinking ‘ why are they doing the course, as mostly they were capable and saying it was easy””

Some experienced programmers who participated in the course also felt that it was not for them, for example:

“I am already an experienced programmer in other languages. I thought I would learn more about Java, but a lot of the lessons were about basic programming concepts that apply to any programming language.”

“I am already an accomplished Java programmer and was looking more toward the Android game theory / development aspects of the course but the course did seem very[much] focused on the basics...”

This shows that it is challenging to fulfil the expectations of a massive audience (Liyanagunawardena et al., 2014), however it is worthwhile noting that this particular issue of prior programming experience of participants became a concern only in the second run, which was open for registration without a cap on the numbers as opposed to the first run that was a beta course on FutureLearn platform with a 10,000 cap placed.

Nevertheless, the course seemed to have retained a sizable proportion of the beginners (Table 2). Over two thirds of learners, 69.5% the first and 68.8% in the second, with prior programming experience had achieved their aims by participating in the course; for learners without prior experience achieving aims were 72.5% and 76.2%. A considerable proportion of participants who rated the course as too basic or too advanced had still achieved their aims in the course. Thus the course seemed to have offered learning opportunities to a large proportion of the diverse audience it attracted.

Different intentions

Participants had various goals that they wanted to achieve in the course. Some participants, mainly beginners, wanted a glimpse of programming while experienced programmers wanted to learn specific topics (for example, Android API) or refresh their programming skills. Some others joined the course to explore online learning. It should also be noted that in the pre-course surveys a sizable proportion (64% in the first and 44% in the second) suggested that they wanted to “try out FutureLearn or massive open

online courses (MOOCs) in general” as a reason for taking the course. A selected set of quotes from post-course survey is shown in Figure5.

<<Figure 5>>

Encouraging the target group

In the first run, out of the course participants, 18% were over 55years as opposed to only 14% under 25s. In the second run the numbers were 13% and 28% respectively. Comparing the numbers to participant demographics of the FutureLearn platform (Table 7) shows that the course in its second run had attracted a sizable group of under 25s above FutureLearn averages.

<<Table 7>>

A group of 6th form (secondary school) pupils who had taken the course with their ICT teachers on the first run of the courses were invited to the University to meet the FLMobiGame team (FutureLearn Blog 2014). On the second run of the course, the University of Reading outreach team distributed information to local secondary schools and information about the course were distributed using social media inviting school groups to participate. This may have helped recruiting more young participants.

Dip in and out?

As facilitators, the authors encountered participants registered in almost all the courses offered by FutureLearn (a participant’s profile shows the other courses s/he registered in). As each course requires about three hours a week time commitment, it was difficult to see how a participant could actually engage with all courses at the same time, unless performing an audit of the courses.

FLMobiGame gradually builds programming skills of the participant. For example, in the first week of the course participants setup their development environment, which they will be using throughout the course; second week introduces the variables and operators that will be essential to progress into the next week. Therefore, unless one has prior programming experience, it is difficult to dip in and out at different points of the course. On the other hand, some of the other courses may have distinct topics discussed each week, which could mean that participants missing a week in between could still be able to follow the course without catching up. The experience of facilitating the course to a massive audience in two consecutive runs showed that participants who joined the course with no experience, but wanted to dip in and out were disappointed due to the way the course increase the knowledge and skill base required for participating in the following weeks. The course, and similar skill-based courses, requires commitment from the participants; learners who are not willing or able to follow this progression will inevitably fall behind.

Conclusion

MOOCs are open for registration allowing anyone who wants to join. However, this can result in some courses attracting a sizable proportion of participants who are not the intended audience. MOOC facilitators are happy for anyone to join and learn in/from the course. However, while delivering a course if issues arise, it may be worthwhile considering the creation of sub groups within the MOOC (for example, different discussion forums: expert discussion forum and beginner discussion forum) to

facilitate the learning of the intended group(s). Nevertheless, it should also be noted that learning through scaffolding takes place when learners with different levels of skills work together, thus such decisions, if taken, should weigh the options fully.

On the other hand, if a course attracts only a small group of the intended audience and receives criticisms from the non-intended audience (for example, in this case experienced programmers rating the course as too basic) it may be worthwhile revisiting course descriptions to explore whether it could be made “clearer” who the intended audience is. However, it would not be wise to redesign the course to address such criticisms unless the course team makes a conscious decision to target a different learner group in a subsequent run. Conversely, it may be worthwhile considering giving pointers to other resources for learners who want more advanced level of understanding appropriately sign posting that they are extra activities.

Analysis presented in the paper is based on pre- and post-course surveys. However, using such data presents the issue of not capturing views of the participants who do not respond to surveys. Though post-course surveys provide useful insight into the experience of participants in the course, it is worthwhile reiterating that they are likely to reflect the views of participants who have completed the course. Another short coming of using pre and post course survey data in a MOOC is that the low rate of response. In some MOOCs, researchers have imposed conditions such as in order to receive a certificate the participant has to submit a completed survey (Kay & McKlin, 2014). Though this has proven to be a successful method to increase survey response rates, it is not without caveats. Thus there is opportunity for further investigations that will provide more insight about learner participation in courses. At the time of writing, authors were unable to map pre- and post-course surveys as the partner institutions are still in negotiation with FutureLearn over the issue. Thus there is opportunity for further investigations that will provide more insight about learner participation in courses once the pre post course survey mapping is in place. Nevertheless, this analysis does present useful insight to understand the difficulties of designing courses to massive audiences where the target learner group may not necessarily be the actual audience of a course.

MOOCs cater for massive and diverse group of learners. The learners in a MOOC have various intentions for taking the course. While some wants to learn the subject others may only want to audit the course. A MOOC is unlikely to please everyone, especially such large cohorts. Therefore, MOOC creators and facilitators should accept that a course cannot be made for everyone.

There is no potential conflict of interest in the work being described here.

References:

Feurzeig,W. et al. (1969). Programming-Languages as a Conceptual Framework for Teaching Mathematics: Final Report on the First Fifteen Months of the LOGO Project, Retrieved online 10/06/2014 at: <http://files.eric.ed.gov/fulltext/ED038034.pdf>

FutureLearn. (2014). Summary of Pre-Course Survey Data, Retrieved online 17/05/2014 at: <https://futurelearn-production-partners.s3.amazonaws.com/wp-content/uploads/2014/05/Pre-Course-Survey-Data-Summary.pdf>

FutureLearn Blog. (02/02/2014). Sixth form students visit University of Reading after taking a course on FutureLearn, Retrieved online 17/05/2014 at: <https://about.futurelearn.com/blog/sixth-form-students-visit-university-of-reading-after-taking-a-course-on-futurelearn/>

Ho, A. D., Reich, J., Nesterko, S., Seaton, D. T., Mullaney, T., Waldo, J., & Chuang, I. (2014). HarvardX and MITx: The First Year of Open Online Courses- Fall 2012-Summer 2013. Retrieved online 17/05/2014 at: http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2381263

Kay, J.S., & McLlin, T. (2014). The challenges of using a MOOC to introduce "absolute beginners" to programming on specialized hardware, In Proceedings of the first ACM conference on Learning at scale conference, 211-212, ACM, New York.

Lee, Y., Choi, J. & Kim, T. (2013). Discriminating factors between completers of and dropouts from online learning courses, 44(2), British Journal of Educational Technology, 328–337. doi:10.1111/j.1467-8535.2012.01306.x

Leutenegger, S. & Edgington, J. (2007). A Games First Approach to Teaching Introductory Programming, SIGCSE '07 Proceedings of the 38th SIGCSE technical symposium on Computer science education, ACM, New York, 115-118. doi: 10.1145/1227310.1227352

Liyanagunawardena, T.R. (2014). Begin Programming: Build your first mobile game, Retrieved online 17/05/2014 at: <http://youtu.be/mW-I3-q5FnQ>

Liyanagunawardena, T.R, Lundqvist, K.O, Micallef, L., & Williams, S.A. (2014). Teaching programming to beginners in a Massive Open Online Course, OER14: building communities of open practice, 28-29 April 2014, Newcastle.

Liyanagunawardena, T. R., Parslow, P. and Williams, S. (2014). Dropout: MOOC participants' perspective. In: EMOOCs 2014, the Second MOOC European Stakeholders Summit, 10-12 February, Lausanne, Switzerland, 95-100.

Lorenzen, T. & Hilman, W. (2002). CS1 and CS2: Write Computer Games in Java!, SIGCSE Bulletin, 34(4), 99-100.

Lundqvist, K. (2013). Teaching Android development using eLearning material in a traditional Java programming course, Innovative Practice in Higher Education, 1(3), Retrieved online 17/05/2014 at: <http://journals.staffs.ac.uk/index.php/ipihe/article/view/35>

McCrinkle, R. (2013). Educational Inclusiveness through Ludic Engagement and Digital Creativity. In C. Stephanidis & M. Antona (Eds.), Universal Access in Human-Computer Interaction. Applications and Services for Quality of Life - 7th International Conference, UAHCI 2013, Las Vegas, USA, July 21-26, 2013, Proceedings, Part III, 195-202, Springer.

Onah, D. F.O., Sinclair, J., & Boyatt, R. (2014). Dropout Rates of Massive Open Online Courses : Behavioural Patterns. In Proceedings of the 6th International Conference on Education and New Learning Technologies (EDULEARN14), 7th-9th July, 2014, Barcelona, Spain.

Rajaravivarma,R. (2005). A Game-Based Approach for Teaching the Introductory Programming Course, The SIGCSE Bulletin, 37(4), 98-102.

Sharples, M., McAndrew, P., Weller, M., Ferguson, R., FitzGerald, E., Hirst, T., & Gaved, M. (2013). Innovating Pedagogy 2013, The Open University, Retrieved online 17/05/ 2014 at:
http://www.open.ac.uk/personalpages/mike.sharples/Reports/Innovating_Pedagogy_report_2013.pdf

Yan, L. (2009). Teaching Object-Oriented Programming with Games, In Proceedings of the sixth International Conference on Information Technology: New Generations, Las Vegas, NV, 969-974.

Table 1 : Survey responses

	Pre-course Survey	Post-course Survey
First Run (Oct 2013)	3606	210
Second Run (Feb 2014)	2657	96

Table 2: Participants with no prior experience in course

	Oct 2013		Feb 2014	
	Pre (n = 3606)	Post (n = 210)	Pre (n = 2657)	Post (n = 96)
Complete beginner (level of programming knowledge)	34%		32%	
No previous experience in subject area		19.5%		27.3%

Table 3: Perceived level of the course

Level of the course	Oct 2013	Feb 2014
Basic	24%	44%
Right	51%	32%
Advanced	25%	24%

Table 4: Prior experience in the subject area and perceived level of the course

		Previous Experience			
		Feb 2014		Oct 2013	
		No	Yes	No	Yes
Perceived Level of Course	Basic	14.3%	58%	9.7%	28.3%
	Just right	38%	32%	61%	48%
	Advanced	47.7%	10%	29.3%	23.7%

Table 5: Experience of learner and the achievement of aims in the course

		Prior Experience			
		Feb 2014		Oct 2013	
		No	Yes	No	Yes
Achieved aims	No	23.8%	31.2%	27.5%	30.5%
	Yes	76.2%	68.8%	72.5%	69.5%

Table 6: Achieving aims and rating the level of the course

		Achieved aims			
		Feb 2014		Oct 2013	
		No	Yes	No	Yes
Level	Basic	50%	38.2%	31.6%	21%
	Just right	13.6%	41.8%	38.6%	56.4%
	Advanced	36.4%	20%	29.8%	22.6%

Table 7: Comparison of participant age

Age	FutureLearn Average (%)	First run Oct 2013 (%)	Second Run Feb2014 (%)
18 years or under	3.9	3	8
18-25	12.5	11	20
26-35	20.1	23	26
36-45	17.4	24	18
46-55	19.7	21	15
56-65	17.2	13	9
66 years or over	9.1	5	4

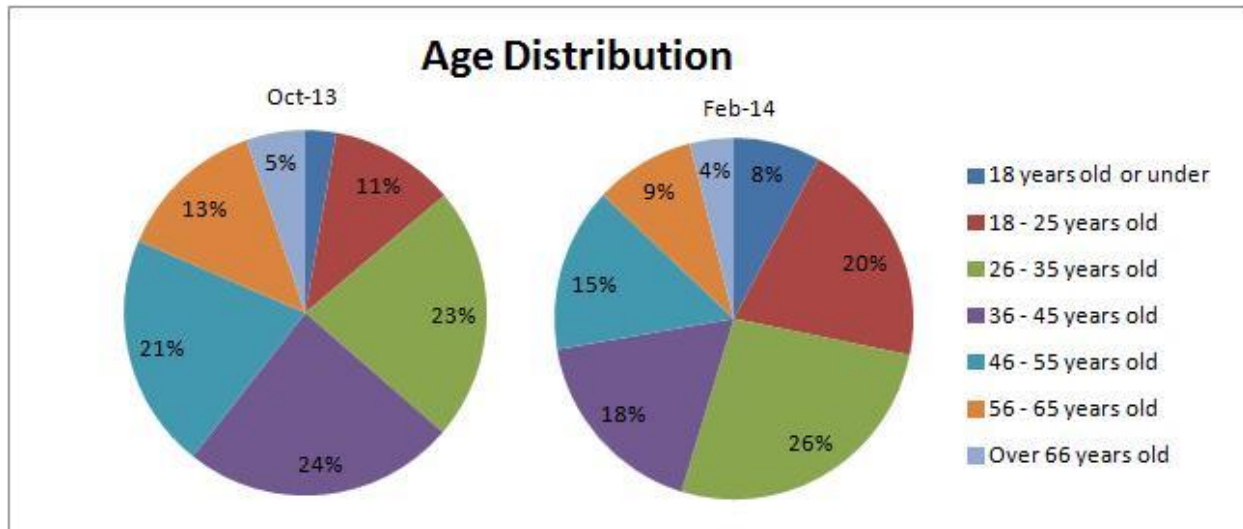


Figure 1: Age distribution

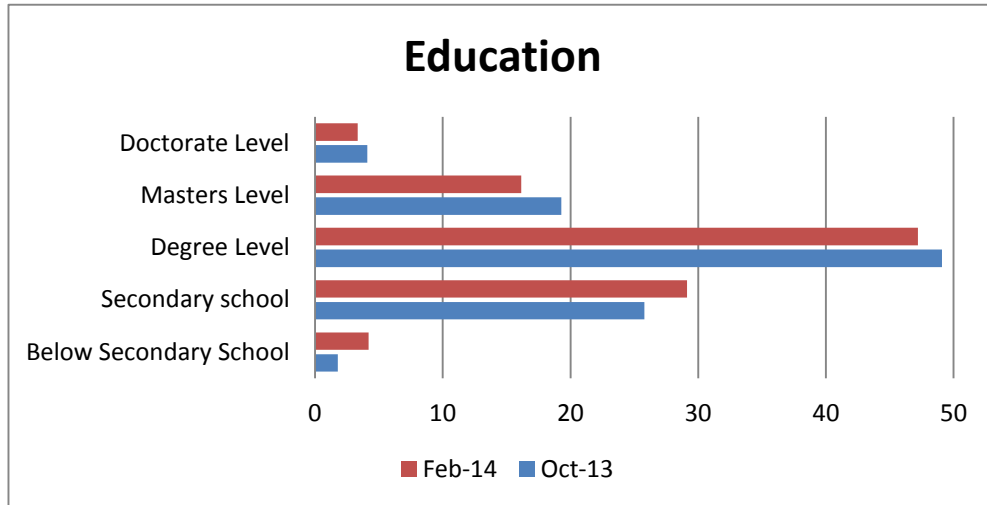


Figure2: Education level

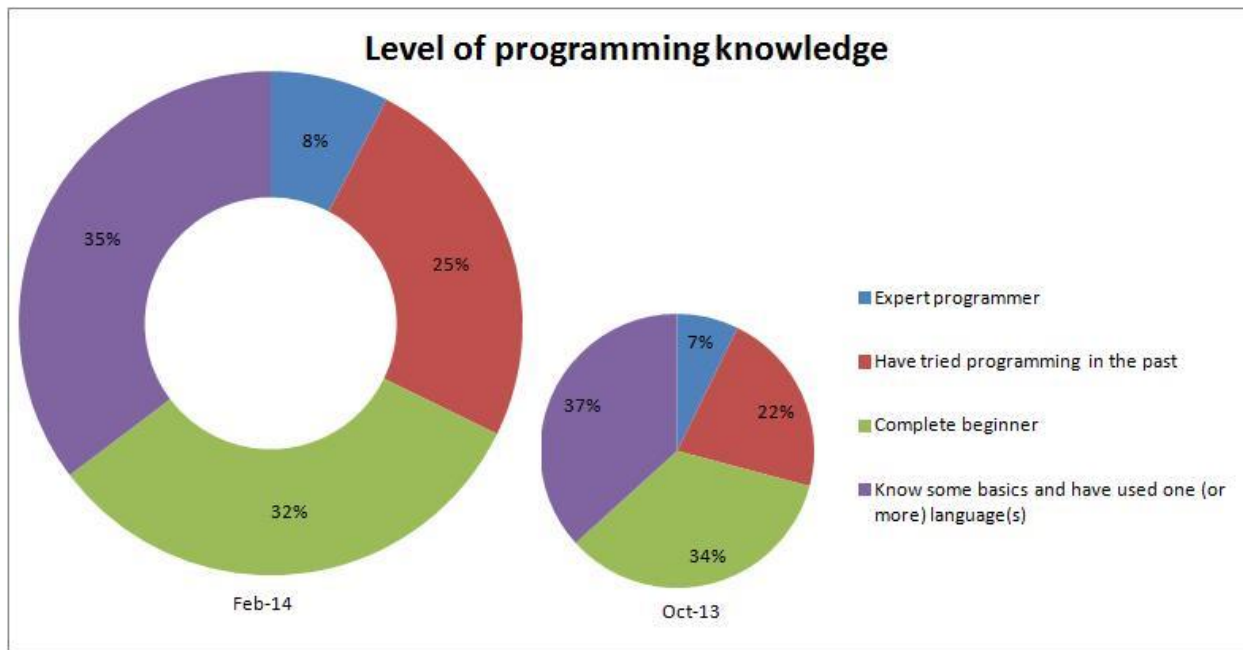


Figure3: Prior programming knowledge

.. this course provides a challenging but fun way to start programming in Java. Over seven weeks we will introduce the basic constructs that are used in many programming languages and help you to put this knowledge into practice by changing the game code we have provided..

Figure4: Course description

Challenge

I just wanted to challenge myself, and I feel that I did...

I felt that I achieved my aims as I got out of my 'comfort zone' and tried something completely new.

Refresh Skills/ Learn Skills

I wanted to refresh my rusty skills, and increase my confidence in my abilities

I've further advanced my programming skills.

[I] was interested in getting a quick intro into Java and the Android platform...

I wanted to gain an insight into computer programming...

I managed to complete my game and put it on the Google Play Store...The course has given me a new hobby!

Explore Online Learning

I wanted to learn a bit about coding and about online learning.

I also imagined that I could help others by writing comments - I did that too. I hoped that I could make friends with likeminded people - that was more difficult due to the constraints of the comment system (i.e. no private messaging)

Figure5: Quotes from post-course Survey (Feb 2014)