MAXIMIZING AND PERSONALIZING E-LEARNING SUPPORT FOR STUDENTS WITH DIFFERENT BACKGROUNDS AND PREFERENCES

Olga Mironova, Irina Amitan, Jelena Vendelin, Jüri Vilipõld and Merike Saar
Faculty of Information Technology, Department of Informatics,
Chair of Software Engineering, Tallinn University of Technology, Akadeemia tee St. 15A, Tallinn 12618, Estonia

ABSTRACT

The present paper introduces some strategies used in teaching an Informatics course for non-IT first year university students. The strategies were developed during several years, using the main principles of flexible and blended learning. The methodology has been based on the results of experiments with students from social sciences, economics and technical faculties. Numerous tests were carried out to find out the level of prior knowledge and the preferred learning styles of each student. According to the results, students were divided into separate groups within an e-learning environment.

The authors considered three main aspects: student's background, the level of knowledge and the most suitable style of learning. The success of the experiment presented in this paper is demonstrated by comparing the academic results of the test group who was taught using the new strategy with the reference group, taught in a common way.

KEYWORDS

E-Learning, Flexible Learning, Felder-Silverman model.

1. INTRODUCTION

High level higher education is one of the most important aspects in the social development of a country. How can a learner get it? What can an educator do to provide it? First of all, both parties should be aware and using modern techniques that help raise learning effectiveness.

At present there are so many opportunities to deliver information and knowledge, especially with the help of the World Wide Web. The Internet has quickly spread into all areas of human life around the world. In our country it is impossible to imagine an everyday activity without some e-components: e-government, e-school, e-health, etc. Furthermore, e-learning is also a rapidly developing possibility with various platforms.

The main goal of e-education is to provide knowledge for each learner in the most suitable form. However, huge amounts of information from different sources do not guarantee perfect knowledge. Therefore, educational e-materials have to be carefully selected and structured to provide for the needs of the learners. It is a new paradigm of the present-day education.

In modern times, all information and, as a result, all knowledge is rapidly changing, making it impossible for educators to predict exactly what the contemporary student will need to know tomorrow – on the next level of his study or in their future work. Instead, teachers have to be helping students to develop the necessary habits and skills which would enable students to learn whatever they might need in their future.

At the moment, Informatics is a continuously and rapidly changing subject in the university curriculum. The main purpose of computing courses, therefore, should be to develop computational thinking and skills needed to use the standard PC equipment on an advanced level, as well as to help users understand and follow the latest trends in the field of IT.

The contents of an Informatics course often seem to be inextricable for the first year non-IT students. There are several reasons for it: from low motivation to the absence of the real IT-projects during the first year at the university. Gradually, it has become obvious that IT-courses need to be innovated and the needs of the target audience have to be taken into account: we need more flexible learning possibilities and as much individualization in the teaching process as possible.

2. THE COURSE OUTLINE

The Informatics course under discussion belongs to the curriculum of all specialities and it has been designed for the first year non-IT students. It lasts for two semesters and is taught in three languages.

The main purpose of this course is to develop computational thinking by means of creating applications (using spreadsheets and some programming environments). The learning process starts with processing and analysing data in MS Excel: writing formulas, using built-in functions and drawing charts. The amount and topics of practical tasks and the theoretical material depend on the students' faculty: social sciences, economics, chemistry, civil engineering or mechanics.

In the second part of the course students learn the basics of modelling and programming in practice. Python and Visual Basic for Application (VBA) are the programming languages used.

It should be mentioned that this part of the Informatics course used to be rather complicated for most of the students, especially students from the economics and social sciences departments. Difficulties at the start of programming tasks were solved by implementing graphical programming as an introduction to the course. After several years of experiments we chose the graphical programming environment Scratch as the main tool to introduce the main programming steps (MIT Media Lab, 2015). Scratch is a very intuitive language and greatly helps learners to take on board the main concepts and terms of modelling and programming such as the variable, subroutine, process, branching or cycle.

During the course, we try to use and combine different styles of teaching and learning: the orthodox face-to-face classroom method, pair or group work and, especially, students' independent learning in the e-environment. A learning environment such as Moodle (Moodle Trust, 2015) provides us with a great variety of different additional opportunities which enable making the learning process more flexible and more individual. A major way is the learning pace adjustment and variety in the number of learning assignments, which are based and matched with students' level of knowledge and their learning styles.

3. THE COURSE EVOLUTION

3.1 The Background

Unfortunately, teachers cannot provide all learners with one-to-one tutoring at all educational institutions. However, this fact has not affected the main aim of the educational process – to guarantee high-quality knowledge and modern skills. During the experiments with our course structure and content, we considered the differences in students' characteristics, especially their background (the faculty and the level of prior knowledge) and their preferred learning styles.

Since 2010, a group of lecturers started applying a new approach to the design of the computer science courses. Year by year this approach has become more flexible and adaptive to the nature of every student.

At the beginning, we randomly divide all the students into equal reference and test groups. The division is not linked to the students' specialization. The average number of members in each group is about 100-150; it varies depending on the general number of students. It should be noted that students are not aware of the research. The reference group is taught using the same course materials but these students are not supported with any additional systems. The students of the test group are directed in choosing their e-learning materials based on the data obtained through the tests in the e-environment.

Our intention here is to compare the results of these two groups at the beginning and at the end of the course.

At the beginning of the course the students are tested to find out their level of knowledge in the Informatics subject. Experience has shown that such testing is necessary for the development of the course content. The purpose is to keep track with new times and main trends, as the computer science is one of the fastest-developing sciences (College Board, 2015), (CSTA, 2015).

The nature of the tests for both groups is similar and based on the concepts defined in the European Computer Driving License (ECDL) (ECDL Foundation, 2015). The assignments focus on some principles of the work with the PC, like creating text documents and presentations, handling information using the spreadsheets, and elementary programming knowledge. Tests include both practical and theoretical tasks.

The programming category of the questions was added to the test some years ago and is currently developing rapidly. We bear in mind that a new elective course "Basics of Application Development and Programming" was recently included in the secondary school curriculum in our country and assume that this is already showing the first results.

The current situation of teaching computer sciences at schools in our country is quite discrepant. Some secondary schools do not have informatics lessons at all, in some schools it is taught only for two or three years, which is a very short period to prepare students for the university level. This drawback is associated with two main reasons. The first one being that there is no nationwide Informatics curriculum in our country. The second one is that Informatics subjects are not mandatory in our primary schools. A logical consequence of these reasons is the situation where each primary school teacher introduces learners to the material at his own discretion: certain pupils draw in Paint, others learn the computer hardware in theory, etc.

In connection with this, the level of PC skills among non-IT learners falls every year and reduces to commonplace Facebook usage. The follow diagram (Fig. 1) shows the last year statistics about non-IT testing at the beginning of the Informatics course. The feasible maximum number of points is 100. As can be seen, the level of computer skills is quite low and steadily decreasing.

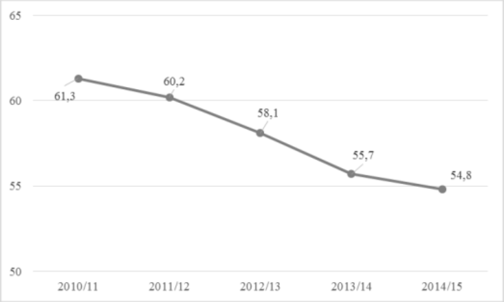


Figure 1. The averages of the beginning test

We are also faced with another problem: school pupils who have obtained sufficient informatics-knowledge at their schools or additional courses often become IT students at university level. Unfortunately, they are not our audience.

In the current work for innovating the Informatics course and individualization of the learning process the authors can name the main strategy – a maximally effective use of the e-environment with its possibilities. E-learning allows modern students (with a personal computer, connected to the internet) to attend the course anywhere and at any time. It is very important to get the material or submit the homework on time regardless of whether the student is or is not at the university.

3.2 E-course

The first phase, which we named "e-course", includes the adaptation of educational materials for the e-environment in Moodle.

Uploading a set of lecture materials and exercises into a learning environment does not ensure that students comprehend it and obtain necessary knowledge. Therefore, in order to make study materials suitable for an e-course all teaching materials (theory as well as materials for practice) were thoroughly revised. Our aim was to provide an effective delivery of the online content. To achieve this goal we aimed at working out a new pedagogical and didactic policy as well as strategies for the new e-course.

Theoretical materials were innovated and supported with learning videos. It should be noted that in our course we use not only video lectures but also short screen-captures, which explain the most complicated tasks. Creating these videos, the authors adhered to the principles of Khan Academy (2015). Practical tasks of the course were reconsidered and supplemented with various group and pair work tasks and also self-tests. These innovations made the Informatics course more dynamic and attractive for our students. Both groups, the test and the reference group got access to this renewed course.

At the same time, we have not abandoned the standard lessons in computer classes. They were held as usual but now we got many advantages. Due to the e-lectures and visual explanations in Moodle we had more time for practical training in contact lessons. It is necessary to mention that students have no access to practical exercises unless they solve the tests which are based on the theoretical material of each topic. Thereby students come to the lessons already prepared for the practical tasks. Often they get a few small practical tasks in the e-environment and afterwards, in class, they use already ready solutions to solve the bigger tasks.

It is generally known that effective computing is impossible without practice. During face-to-face lessons students work in pairs or groups and are given an opportunity to try the obtained knowledge in practice. Moreover, such kind of work develops teamwork skills, which is very important, especially for the first-year students during their first semester.

In such case, the role of the classical educator is slightly different – the lecturer becomes more of a supporter in the students´ team work of their learning assignments. During contact lessons learners have an opportunity to ask questions related to their homework and share their practical skills and experience with the rest of the group. Practical knowledge transferred in such a way is obtained much faster than in standard lectures. This fact was confirmed by the results of student tests and feedback.

In their feedback students named another advantage of such practical lessons: they have an opportunity to get support or to ask something not only from their teacher but from other students, too. It should be mentioned that this form of support is equally important and useful for both sides: the one who gets it and, especially, for the one who gives it. To detect, explain and, afterwards, to correct a mistake in calculations or in the program code is a substantial practical skill in computer science subjects.

In addition, in an e-environment students get their practical assignments in accordance with their specialities but still united under a common subject topic. For example, students from the economics department get more tasks related to table calculations; social sciences students implement the information filtration, statistics calculations and various requests.

It should be noted that the course materials are organized sequentially and it is not possible to get a new portion of theoretical materials and practical tasks without solving the previous ones. Thanks to checking opportunities in the e-environment, like tests or self-tests, teachers do not have to spend time on routine inspection of the assignments at all. Using the automated checking tools gives students an opportunity to learn within their own pace. They do not have to wait for the feedback on the assignments from the lecturer and his/her manual permission to proceed onto the next level. E-checking systems do it faster and as many times as is needed.

In addition to the aspects mentioned above we and our learners use forums very actively in our course. This way the students get an opportunity for online communication and fast online help. Our aim was to show that they can get support and advice any time, and can also share their ideas, problems and solutions. Through these forums we often get perfect brainchildren for group work and individual assignments.

Another advantage that non-IT students have when they participate in the e-course is that their technical skills improve.

Transition to the Moodle e-environment gave our educators an opportunity to follow students´ progress and it became fairly easy to get the statistical data of different samples for analysis, development and improvement of the implemented learning methods.

It makes no sense now to recount all the advantages that we achieved during the first stage of our modification of the course. Such benefits have already been systematized and described in detail in different sources (Broadbent, 2002). However, we would like to mention the first positive results of our work: students' feedback and increase in academic achievements. The data was extracted from an e-environment, where students and teachers get information about the courses and curricula, students declare courses, keep results and give anonymous feedback on their educational process. The authors do not present specific data in numbers in this paper because the transition of the course into e-learning took place a long time ago.

3.3 Prior Knowledge

Starting from the second stage, which the authors entitle "prior knowledge", experimental work takes place only with students from the test group. The students of the reference group are taught as usual.

At the beginning of the course we start with dividing the students into three e-streams based on their readiness for Informatics subjects. This division was realized through an e-test and implemented in the Moodle environment. The students, however, were not aware of the experiment.

In the described division we proceeded from the level of students' knowledge required to start the course. Those students whose e-test results are more than 85 points we named "experts"; "advanced users'" result is between 60 and 80 points, other students are called "beginners".

The named groups of students receive different amounts of practical and theoretical tasks in Moodle, with different levels of difficulty. To move to the next topic the mandatory set of exercises has to be solved. In the e-environment, the "beginners" and "advanced users'" have to solve their sets of tasks before they get access to "experts'" exercises – the main material of the course curriculum. "Advanced users" solve their tasks and can proceed to the main topics.

These additional sets of tasks are catered for exactly what learners need to know for the current Informatics course. Students do not need to pay for any additional IT-courses and they get all the materials and assignments centrally, in one place – the Moodle e-course, in parallel with their main studies.

To automate and speed up the checking of the increased number of tasks we have developed a special e-tests system.

Thereby, we could increase the amount of practical assignments for students with different levels of readiness without increasing the subject hours and students' load. This stage of the course innovation gave us an appropriate level of the students' readiness for face-to-face lessons.

The above mentioned method provided our Informatics teachers with actual and important information about what the learners knew before starting the course. Every year we get an overview of the current situation of Informatics subjects at secondary schools in our country. Moreover, according to the results, we are able to provide students with the necessary learning materials.

3.4 Learning Style

The third part of our innovation experiment with the course modification is the "learning style" phase and it was also realised in Moodle. This stage is a continuation of the test group's students division into groups. There is a lot of research about the individualization of learning depending on students' abilities (Kolb, 1984), (Palmer, 2011), (George Lucas Educational Foundation, 2014). In our course Felder-Silverman model was picked out as the basis for the distribution (Felder & Spurlin, 2005), (Felder & Brent, 2005). In this stage we could maximize the use of the Moodle e-opportunities.

Learning styles are characteristic cognitive, affective, and psychological behaviours that serve as relatively stable indicators of how learners perceive, interact with, and respond to the learning environment. Students learn best when instruction and learning context match their learning style.

Depending on their learning style, Felder differentiates between the following groups of learners:

- active and reflective
- sensing and intuitive
- visual and verbal
- sequential and global.

For better comprehension of the topic we give a brief description of the learners' types.

Active learners acquire new knowledge best by doing, discussing and explaining it to others in a group. At the same time reflective learners first think about it alone.

Sensing learners like learning facts and solving problems by well-known methods. Intuitive learners prefer discovering new possibilities and relationships and they are more innovative.

Visual learners remember pictures, diagrams, charts and videos best. Verbal learners prefer written and spoken explanations.

Sequential learners like step by step studying, where each step follows logically from the previous one. Global learners prefer to get information by large portions and randomly.

Through a test, we learned that the majority of our course participants in the test group were active and visual learners and they had very strong preferences for their learning process. These preferences were detected according to the Felder test, which was held at the beginning of the course. It should be noted that each year the number of active and, especially, visual students increases (Fig. 2).

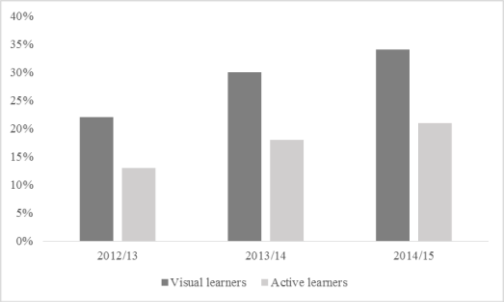


Figure 2. The increase of the number of the visual and active learners

Throughout the educational process, students were provided with necessary learning materials and activities in accordance with Felder's instructions.

For example, active learners automatically received more group work and opportunities to help others – they could check and correct other students´ work and assignments in Moodle; they answered questions in the e-course forums and took the role of a tutor in face-to-face classes. It should be mentioned that they did it with pleasure. For visual learners, a great variety of visual representation of the educational materials (that was already mentioned above) was provided. Interests and preferences of the other types of learners were also taken into account.

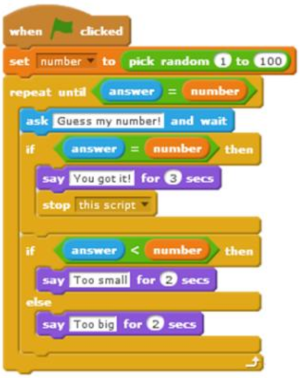
4. PROGRAMMING BASIS FOR VISUAL LEARNERS

As mentioned above, the majority of the course participants are visual and active learners. For visual learners it is very important to see "how it works" and for active – "to try it out". In the most complicated module of the Informatics course, programming, we found a solution – maximum visualization.

Our introductory module, with Scratch as the main tool with its graphical elements, is already perfectly visualized. Scratch´s dynamics and attractiveness helps visual students to understand the main ideas of the created applications. Working in this programming environment, active learners can manually modify some program elements and immediately see the result. Moreover, there are no syntax mistakes in Scratch that greatly facilitates the programming process. Therefore, introduction to programming in our course has already been visualized (Fig. 3).

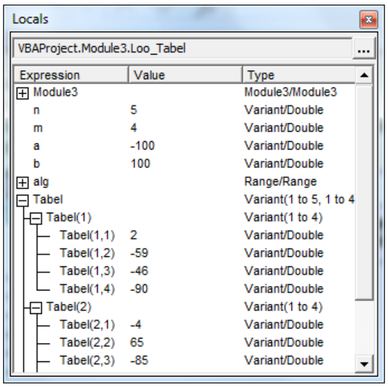
The next module, VBA or Python, is rather complicated for non-IT students. What can be visualized there?

VBA already has a built-in visualising tool: students can follow the code execution using the Locals Window (Fig. 4). The named window automatically displays all the declared variables in the current procedure, their names, types and values in the real time. When the Locals Window is active and visible, it automatically updates during the time the program is running – visual students can follow the code and check each step and its result in their applications, especially if they run programs in step-by-step mode.



```
when 🏴 clicked
set number ▾ to pick random 1 to 100
repeat until answer = number
    ask Guess my number! and wait
    if answer = number then
        say You got it! for 3 secs
        stop this script ▾
    if answer < number then
        say Too small for 2 secs
    else
        say Too big for 2 secs
```

Figure 3. Scratch program



Locals

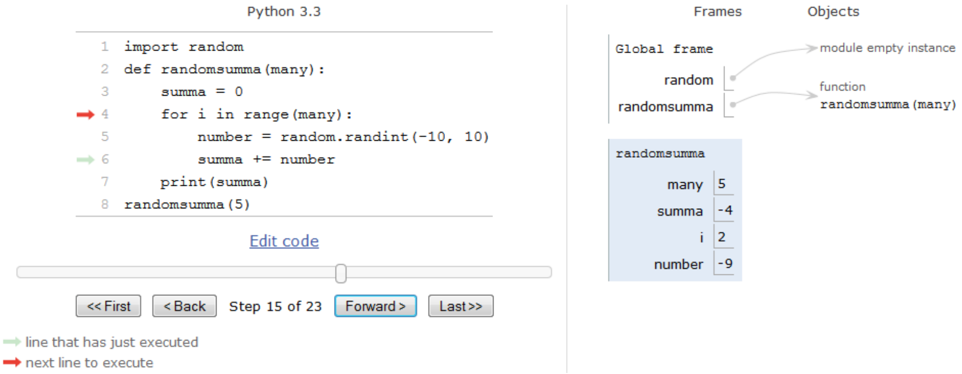
VBAProject.Module3.Loo_Tabel

Expression	Value	Type
⊞ Module3		Module3/Module3
n	5	Variant/Double
m	4	Variant/Double
a	-100	Variant/Double
b	100	Variant/Double
⊞ alg		Range/Range
⊟ Tabel		Variant(1 to 5, 1 to 4
⊟ Tabel(1)		Variant(1 to 4)
Tabel(1,1)	2	Variant/Double
Tabel(1,2)	-59	Variant/Double
Tabel(1,3)	-46	Variant/Double
Tabel(1,4)	-90	Variant/Double
⊟ Tabel(2)		Variant(1 to 4)
Tabel(2,1)	-4	Variant/Double
Tabel(2,2)	65	Variant/Double
Tabel(2,3)	-85	Variant/Double

Figure 4. VBA visualizing tool – Locals Window

Besides the Locals Window it is possible to use the Immediate Window to reach the same goal: check and correct mistakes and understand the program structure.

Unfortunately, Python does not have such an opportunity, but in spite of this, it still needs to be visualised for the beginners in programming. Therefore, we introduce the Online Python Tutor to our students. Using this internet application they can follow each step of their program code and check the variables' values, types and the order of the operators during the execution. It has to be mentioned, though, that the online tool has some drawbacks, as it does not support the Python graphics, time functions and files processing. However, for the beginners in coding the application gives an understanding of the main principles of program construction and code execution (Fig. 5). There have been cases when students started to write the program not in Python environment but directly there, on the internet.



```
import random
def randomsumma(many):
    summa = 0
    for i in range(many):
        number = random.randint(-10, 10)
        summa += number
    print(summa)
randomsumma(5)
```

many	5
summa	-4
i	2
number	-9

Figure 5. Python visualizing tool

Using these online and built-in tools we could raise the interest to programming among the non-IT students in the second part of the Informatics course.

5. COMPARING THE RESULTS

Figure 6 shows the results of both groups´ tests at the beginning and at the end of the course. The possible maximum number of points is 100.

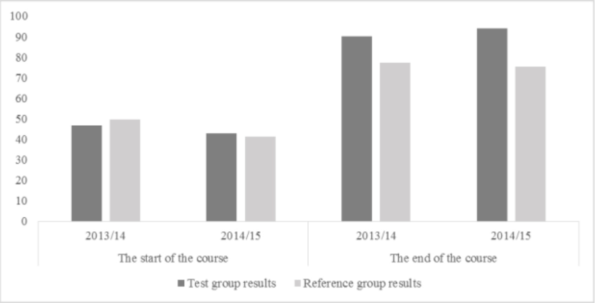


Figure 6. The testing results of the both groups

As can be seen, the average test results of the reference group students are high enough - approximately 76 points, but test group students' average results are 16 points higher – about 92. The given outcomes confirm the topicality of our research and the suitability of the chosen method of the course innovation.

6. CONCLUSIONS

During the process of upgrading the Informatics course, we were able to create a model of individualization of the educational process in an e-environment, which considers the level of students´ prior knowledge and their preferences in the learning process (Fig. 7).

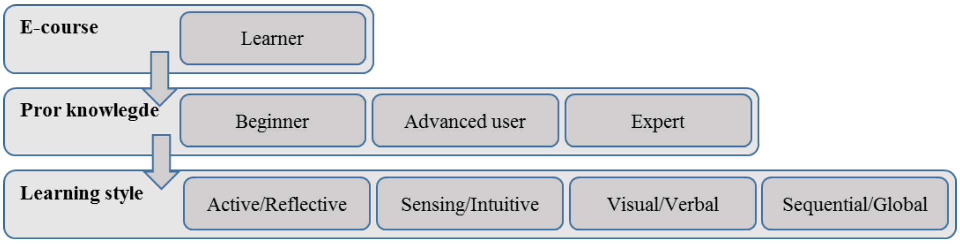


Figure 7. The model of the course

Applying this model, we try to find an individual approach to each student in our e-course and make it more flexible.

The experiment results, presented in the article, show positive outcomes of the strategy and suggest that there are no unreachable aims in an educational process. The authors intend to continue with the created model and develop the Informatics e-course for non-IT students in the chosen direction, trying to maximally adapt it to students with different preferences.

ACKNOWLEDGEMENT

The authors thank anonymous EL2015 reviewers for their comments and suggestions.

REFERENCES

Broadbent, B., 2002. *ABCs of e-Learning: Reaping the Benefits and Avoiding the Pitfalls.* 1st ed. NY: John Wiley & Sons, Inc.

College Board, 2015. *AP Computer Science Principles.* [Online] Available at: https://advancesinap.collegeboard.org/stem/computer-science-principles [Accessed 4 5 2015].

Computer Science Teachers Association, 2015. *Computational Thinking Task Force.* [Online] Available at: http://csta.acm.org/Curriculum/sub/CompThinking.html [Accessed 2 5 2015].

ECDL Foundation, 2015. *ECDL Foundation.* [Online] Available at: http://www.ecdl.com [Accessed 1 5 2015].

Felder, R. & Brent, R., 2005. Understanding Students Differences. *Journal of Engineering Education,* 94(1), pp. 57-72.

Felder, R. M. & Spurlin, J. E., 2005. Applications, Reliability, and Validity of the Index of Learning Styles. *Intl. Journal of Engineering Education,* 21(1), pp. 103-112.

George Lucas Educational Foundation, 2014. *Project-Based Learning.* [Online] Available at: http://www.edutopia.org/project-based-learning [Accessed 24 11 2014].

Khan Academy, 2015. *Khan Academy.* [Online] Available at: https://www.khanacademy.org/ [Accessed 3 5 2015].

Kolb, D., 1984. *Experiental Learning.* Engle Cliffs: Prentice Hall.

MIT Media Lab, 2015. *Scratch - Imagine, Program, Share.* [Online] Available at: http://scratch.mit.edu/ [Accessed 6 5 2015].

Moodle Trust, 2015. *Open-source learning platform | Moodle.org.* [Online] Available at: https://moodle.org/ [Accessed 6 5 2015].

Palmer, S. R., 2011. The Lived Experience of Flexible Education – Theory, Policy and Practice. *Journal of University Teaching & Learning Practice,* 8(3).