Visualisation and Gamification of e-Learning and Programming Education

Marie Olsson, Peter Mozelius and Jonas Collin
Department of Computer and Systems Sciences, Stockholm University, Sweden
marieols@dsv.su.se
mozelius@dsv.su.se
jcoll@dsv.su.se

Abstract: Courses in virtual learning environments can leave recently enrolled participants in a state of loneliness, confusion and boredom. . What course content is essential in the course, where can more information be found and which assignments are mandatory? Research has stated that learner control and motivation are crucial issues for successful online education. This paper presents and discusses visualisation as a channel to improve learner's control and understanding of programming concepts and gamification as a way to increase study motivation in virtual learning environments. Data has been collected by evaluation questionnaires and group discussions in two courses partly given in the Moodle virtual learning environment. One course is on Game based learning for Bachelor's programmes, the other is a course on e-learning for university teachers. Both the courses have used progress bars to visualise students' study paths and digital badges for gamification. Results have also been discussed with teachers and pedagogues at a department for computer and systems sciences. Furthermore, two visualisation prototypes have been designed, developed and evaluated in programming lectures. Findings indicate that visualisation by progress bars is a good way to improve course participants' overview in online environments with rich and multifaceted content. To what degree the visualisation facilitates the course completion is hard to estimate, and like students have different learning styles, they also seem to have different visualisation needs. Gamification by digital badges seems to have various motivational impacts in different study groups and in traditional university programmes the traditional grades seem to be the main carrots. Finally, it seems that software visualisation might be a promising path to enhance programming education in the 21st century.

Keywords: Visualisation, Gamification, Programming education, Virtual learning environments, E-learning

1 Introduction

A general trend in the 21st century is that traditional face-to-face rostrum teaching and learning at university level is transformed to courses including online activities reaching various percentages of blended learning and dual modality (Graham, 2006; Lim & Morris, 2009; Park & Choi, 2009). If these new online virtual learning environments are not designed carefully, learners can get stuck in a state of confusion (Hara & Kling, 2000) and loneliness (Brown, 1996). Online learning must, like traditional education, be designed with a human touch otherwise there is a risk for low motivation (Keller & Suzuki, 2004) and high drop-out rates (Park & Choi, 2009).

A critical phase for every online course is the learner's first login. Common question that often arise are: What course content is essential in the course? Where can I find more information? Which exercises and assignments are mandatory to complete the course? The identified problem with high drop-out rates in the inception phase of online courses might be caused by the students' cognitive overload. Learning of new content or skills, for which a schema in long term memory is lacking, can cause (the) learner's limited working memory (to) overload. This overload can result in unconfident and even anxious learners, which can lead to a failing learning process and course drop-out (Tyler-Smith, 2006).

Results in a report on online education at the Open University in UK indicate that 38% of e-learners leave courses before submitting their first assignment (Simpson, 2004) which suggests that a learner's initial interaction with an online platform may have a significant impact on further studies (Tyler-Smith, 2006). Research has also stated the importance of learner control (Chou & Liu, 2005) and learners' motivation (Keller & Suzuki, 2004). This study will have a focus on how the concepts of visualisation and gamification might increase learners' self-control in online learning environment.

Teaching programming to novice students is classified as hard, especially to weaker students. It is perhaps not surprising since computer programs and algorithms are abstract and complex entities, that involve concepts and processes, which are often found difficult to teach and learn ((Guzdial & Soloway, 2002; Lahtinen, Ala-Mutka & Jarvinen, 2005; Eckerdal, 2009). Research on multimodal teaching has shown that adding more

channels for the knowledge transfer can facilitate learning in general. This could then be of particular use in programming education, which is a relatively abstract form of both knowledge and skills. Multiple forms of representation or modes, each performing its semiotic labour, also benefit the understanding (Kress, 2010). Further, the way knowledge is presented is relevant to the content and the form can be seen as part of the content, instead of separate from the content (Rostvall & Selander, 2008). Learning is thereon focused upon as communication and sign-making activities; how the information is processed and transformed into knowledge within a certain institutional setting (Selander & Kress, 2010).

In this paper, a design theoretic multimodal perspective to teaching and learning programming has been applied and the approach builds on adding an extra channel of communication between teacher and student. The channel selected is a visual display of the dynamic parts of program execution, usually called software visualisation. Software visualisation aims to facilitate learning and understanding of programming and simplified it can be seen as the utilization of graphical representations, in form of text, pictures and animations, e.g. a rapid display of a sequence of pictures. Two main underlying concepts of software visualisation are program visualisation and algorithm visualisation. Program visualisation is a graphical representation of a running program, i.e. software code that is executing and algorithm visualisation is a graphical representation of an algorithm, i.e. a set of instructions that are carrying out a specific task.

1.1 Aim of the study

The aim of the study is to describe and analyse techniques for visualisation and gamification in virtual learning environment, and to discuss how they might increase learners' control and motivation.

2 Extended background

The two techniques to increase learner control and motivation that are described and discussed in this study are visualisation and gamification. Both concepts have been hyped and extended to various definitions during the last decade but in this study the terms are used as they are defined below.

2.1 Visualisation

Students of today need more skills to be able to work efficiently and manage their education than students a few decades ago, as the integration of technology and e-learning has exploded. One of these skills is visual literacy, which is also one of the most critical competencies (Mnguni, 2014). Selander and Kress (Selander & Kress, 2010) are significantly taking into account that learning is largely shifted from local to global learning environment and describes learning from a multimodal perspective, where resources, in many cases visualisations, has a central role.

For decades it has been common to use graphical visualisations as a learning aid. And static visualisations are exquisitely capable of showing the appearance or the constitution of something. However, visualisations such as animations are able to provide for a more exhaustive understanding of abstract and dynamic entities, since changes and states can be animated step by step.

Selander and Kress focus learning as sign-making activities and communication. The common base for the sign-making activities consists of multimodality and didactic design. Multimodality is based upon the different resources (visualisations included) that are used for interpretation and to create meaning. However, words, symbols, visualisations etc. do not mean anything itself, rather they are assigned its meaning from the context in which they are created and used.

Didactic design is used as a term for how individuals constantly recreate (re-design) information in their own meaning-making processes. It is also used as a term for how to form social processes and create learning conditions. Further, Selander and Kress mean that all the arrangements produced to facilitate learning, such as visualisations, are all design for learning. Selander and Kress have developed a model that summarizes a formal learning design sequence (Figure 1), where the information is first being processed and then transformed into knowledge.

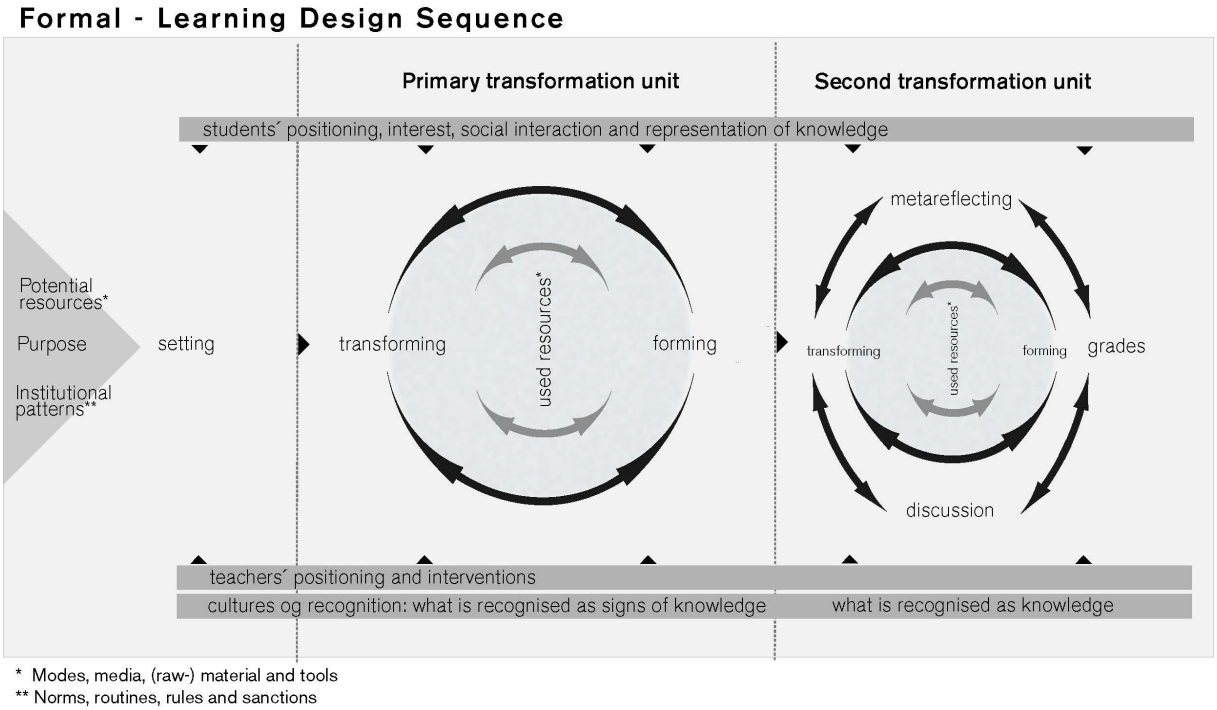

Figure 1: Selander 2008, processed by Anna Åkerfeldt 2014

Some potential resources, like visualisations, are available in a learning situation. There is a purpose of the activities and the engaging activities also relates to various degrees of clear institutional patterns, routines and habits. Selander and Kress believe that those are important aspects where an activity or a learning sequence begins. The form of arrangement created for certain activities is set. The setting also includes individuals´ interpretation and framing of the situation and the way the individual position itself. Resources used, i.e. visualisations are transformed and formed in a primary transformation unit. It can be seen as a transformation cycle, in which information is selected, processed and combined in a new way, to form a representation. To reach a coherent new representation, the processes are neither linear nor circular, but about exploratory, sketchy and hypothetical processes. Selander and Kress believes that interaction, dialogue and positioning are important factors to understand the direction that the activities can take. The choices made in a learning process can be seen as fixation points where a meaning and a basis for a possible continuation temporary is locked. What one chooses depends on what catches the attention at a particular time, and what is perceived as an interesting challenge, according to Selander and Kress. Further, the resources that are used to mould and shape ones understanding is connected with what is perceived as an obvious, reasonable or possible resource in a situation.

In the transformation, one uses, combines and creates new representations by using the resources available, i.e. visualisations, in a situation. Those representations are more sketchy representations processed along the way. The representation which forms the transition to the second transformation unit in fig 1, is referred to a more finished fixation of an individual's understanding of a phenomenon, by Selander and Kress. In the second transformation unit individuals' knowledge and understanding is represented. And a new transformation cycle of discussion, assessment and meta-reflection on the more finished representation, and on the learning process as a whole starts.

2.2 Gamification

According to Deterding et al (2011) *"'Gamification' is the use of game design elements in non-game contexts"*. Unpacking their definition, Deterding et al, argues that gamification is more about gaming than playing – there are rules to follow. Gamification use elements of game design but can't be considered as full-fledged games. There are several levels of game elements where a gamified system can borrow its design; Game interface design patterns, Game design patterns and mechanics, Game design principles and heuristics, Game models and Game design methods. In other words a gamified system is designed to look and/or feel like a game but it

does not go all the way. These game elements are used for other purposes than being part of a game, thereof the non-game context.

Karl M Kapp defines gamification as *"using game-based mechanics, aesthetics, and game-thinking to engage people, motivate action, promote learning, and solve problems."* (2012, p. 10) This definition considers the purpose of gamification and points at the pedagogical use of the concept; *"to gain a person's attention and to involve him or her in the process you have created."* (p.11).

In gamification, motivational theory is really important, where for example the Self-Determination Theory (SDT) (Ryan & Deci, 2000), ARCS (Keller, 2010), The Taxonomy of Intrinsic Motivation (Malone & Lepper, 1987), divides motivation into extrinsic and intrinsic motivation. Extrinsic motivation is motivation due to external rewards and intrinsic motivation refers to rewards that come from within a person. Keller (2010) state that *"it is more common to find that there are elements of both [intrinsic and extrinsic motivation] that are intertwined in any particular situation"*. As an example he mentions that even if you like your work, you probably wouldn't do it without the salary.

On the one hand extrinsic motivators are very useful if they lead to intrinsically motivated learning activities. But on the other hand if the learners already are intrinsically motivated, extrinsic motivators can undermine the intrinsic motivation. Kapp especially states *"extrinsic reward structures of engagement-contingent rewards, completion-contingent rewards, and performance-contingent rewards all undermine intrinsic motivation in most cases"* (Kapp, 2012).

In his argumentation around satisfaction, Keller (2010 p.53) states *"one of the most rewarding results of performance-oriented instruction is to use the newly acquired skills or knowledge"*. But there is not always possible to put new knowledge into use and the learner can also lack intrinsic motivation. Here Keller suggests the use of extrinsic reinforcement. He also states the importance of equity, to ensure that the course outcomes are consistent with what has been communicated.

It's easy to think that gamification is all about badges, scores and leader boards, which would be a significant part of a behaviourist concept. Bíró (2013) states that gamification has more in common with the behaviourist concept than with the cognitivist, constructivist and connectivist approaches. In behaviourism though, the learner is considered to be a reactive and instinct-driven individual, motivated extrinsically by an active teacher to learn the external knowledge that the teacher has defined as the right one. But within gamification theory the learner is thought of as a conscious proactive individual with an intrinsic motivation. Knowledge seems to be both internal and external.

Bíró (2013) points out that *"gamification considers the visual dimension of the learning process very important, especially the visualization of the advancement in the learning process and the chosen learning path."* In other words visualisation theory can bring a lot of important aspects to gamification theory.

2.3 Software visualisation

Software visualisation is either a static or a dynamic and animated visual representations of information about software systems. In this study, software visualisation is referred to as graphical representations, in form of animations of programs and algorithms. Further, software visualisation can be seen as means of communication between the designer of the visualisation and the user of the visualisation. Moreover, it is associated with principles from cognitive psychology and perception (Petre, 1995). Traditional programming education has focused on the transmission of a complete mental model to the students. With visualisation the students are offered additional ways to realise how the dynamics of the programs work and several paths to gain knowledge are offered by different potential learning sequences. Two main underlying concepts of software visualisation are program visualisation and algorithm visualisation. Their definitions are outlined as follow:

- Algorithm visualisation – a high-level description of a piece of software is visualised (Price, Baecker, & Small, 1993), not related to the program's source code and without details (Myller, 2004), often used for a better understanding of the procedural behaviours of the algorithms (Hundhausen, Douglas & Stasko, 2002).

- Program visualisation – programs are mapped to be represented graphically (Roman & Cox, 1993). Thus, different states of the program or a running program are illustrated graphically, while the program is defined textually, for example in form of source code (Myers, 1986).

Both these concepts are employed in the experiment, in trying to convey knowledge about the execution of computer programs, but mainly program visualisation is being used in the two animations: iteration visualisation and object visualisation. They differ in their aims, in what learning hurdles they are trying to bridge. While iteration visualisation mainly deals with problems of understanding the dynamic behaviour of computer code, object visualisation also deals with conceptual problems of understanding objects in addition to dynamic properties.

2.4 Software visualisation with a design theoretic multimodal perspective

In the design theoretic multimodal perspective, the learning is focused as a signs-interpretive and signs-creative activity, where information is transformed and new representations of knowledge are created. Further, (Selander & Kress, 2010) view learning as meaning-making communication in the context of specific situations in an institutional setting. Design shapes objects as well as conditions for communication and in the context of didactic design, design creates new meaning. Multimodality is based on the various kinds of resources that can be used to interpret the world and create meaning.

The teaching and learning of programming with the help of code visualisation can be seen as formalised learning design sequences. In this study, the first setting arranged is the animation of an executing computer program and the second setting examined is an animation of an executing algorithm. Presenting animations for the students represented the learning sequences, where the aim was to study whether an extra added visualisation channel could offer extra possibilities of meaning-making and learning. The added channel consisted of a visual display of the dynamic parts of program execution; one about imperative programming and one about object-orientation. Imperative programming and object-orientation are two typically hard concepts to teach and learn, therefore these concepts were tested. To guide the student's attention to the key aspects in the context of the moment, and for the students to more easily orient themselves, e.g; see patterns, interpret, frame, position themselves in the context and create meaning, each learning sequence is supplemented with narration. Additionally, the program code is also continuously visible. Each learning sequence is framed by the fact that a sequence begins when the animation begins and ends when the animation ends.

Transforming and forming are two central sequences in Selander's model of a formally framed learning design sequence. In this study, transforming and forming are the processes where the students need to interpret, process and transform the information in the settings to representations of their own. It can be looked upon as a transformation cycle, where information is selected, processed and combined in a new way, to form a representation. The processes are neither linear nor circular to gain a new coherent representation. Instead, it is about exploratory, sketchy and hypothetical processes. In the attempts of transforming information, aspects as iteration of processing information, interaction and dialogue are of importance. In this study, the settings actually consist of the classical model: sender – message – channel – receiver and the transforming and forming of information refers to individual reflections and processing in the representation phase.

3 Methods for data collection

Data has been mainly been collected from course evaluation questionnaires and from discussions and comments at course seminars. Students study patterns were analysed by their individual progress bars where completed activities are marked. Gamification and visualisation ideas have also been discussed with colleagues at the department where the authors are employed.

The overall approach is a case study, with the case study strategy defined as an investigation of a real world process or phenomenon (Yin, 1989). To explore the chosen processes in depth a mix of data collection methods are often used (Creswell, 2009) where the combination of different data sources should support a deeper understanding of the investigated process (Remenyi, 2012).

Evaluation questionnaires were given in Swedish and all quotations in the article have been translated to English by the authors. For ethical reasons all informants are kept anonymous and screen shots have been

edited in *Photoshop* to remove names and other details that might be associated with individual course participants.

To study how the software visualisations worked, a learning experiment was devised. Two typically hard concepts for beginners were selected, one in imperative programming and one in object-orientation. For the imperative part, the dynamics of a for-loop was chosen. For the object-orientation part, the dynamics of an object in the form of instantiation, inheritance and method calls was chosen. These two prototypes were shown to first year undergraduate programming students on two separate occasions (one for each prototype).

Students were divided randomly into two groups, one experiment group and one control group. There was no announcement of the lectures being different and the students were not given a choice of which lecture to participate in. For the iteration visualisation prototype, the experiment group consisted of 85 students and the control group of 72 students. For the object visualisation prototype, the experiment group consisted of 59 students and the control group of 57 students. In each case, the control group was given an ordinary lecture, consisting of a teacher talking the students through the program code and at the same time drawing on a whiteboard. Thus, they were offered the lecture through the ordinary channels of text and voice. For the experimental group, the visualisation was added to the content shown to the control group. Thus, they were offered an additional modal channel of dynamically seeing the code being executed in animated form.

4 Courses and software visualisation experiments

Both courses have been given twice during the autumn semesters of 2013 and 2014 at a department for Computer and Systems sciences. The courses have large variations regarding content and activities but a common denominator is the tendency to test and evaluate new blended learning techniques.

4.1 The course on game-based learning

A syllabus consisting of 9 lectures, 2 workshops and 5 assignments builds a foundation for the final project where students should design, implement and present an educational game. Course content as well as examination is a mix of theoretical and practical work where reading and code analysis is combined with essay writing and programming. Various guest lecturers present their niches in Game-based learning, design and programming. Literature and teaching sessions also includes gamification, pedagogy and accessibility aspects. Practical programming assignments where the students build games are interwoven with theoretical exercises where short essays are posted and discussed online.

Discussion fora and course content with recorded lectures and workshops are available in the *Moodle* virtual learning platform. Almost all students take the course in a blended learning mode but some participants have completed the full course, with an exception for the final seminar, in distance mode. According to the Bologna model, 7.5 credits (ECTS) correspond to 5 weeks of fulltime work but this course is given in parallel with another course where both courses have a 10 week frame. Closed assignments with strictly given instructions are combined with open tasks where students are free to design and create their own solutions. The more closed assignments are exercises in HTML5, CSS, JavaScript, jQuery and game creation techniques. The aim with these more technical assignments is to provide the skills needed to implement the game in the final project. For the higher grades the project submission should consist of a unique, playable learning game, tested and with the development process documented.

Moreover, game design should, for the higher grades, also be aligned to a pedagogical or motivational theory/model. The game idea should also be presented in the students' ePortfolio that has been created as one of the mandatory assignments as well as in the Moodle platform with the basic game idea demonstrated in a video. Finally at the examination seminar the created games are presented and discussed to get feed forward for further game development.

At the examination seminars there is also a discussion on course activities and course structure. In one seminar at the end of the first version of the course there was a discussion on how the course outline might be presented more clearly. As a result of this debate the individual progress bars, that are evaluated in this study, were added to the course and mapped to the course activities. The progress bars in the second version of the course had the standard design that is depicted below in Figure 2.

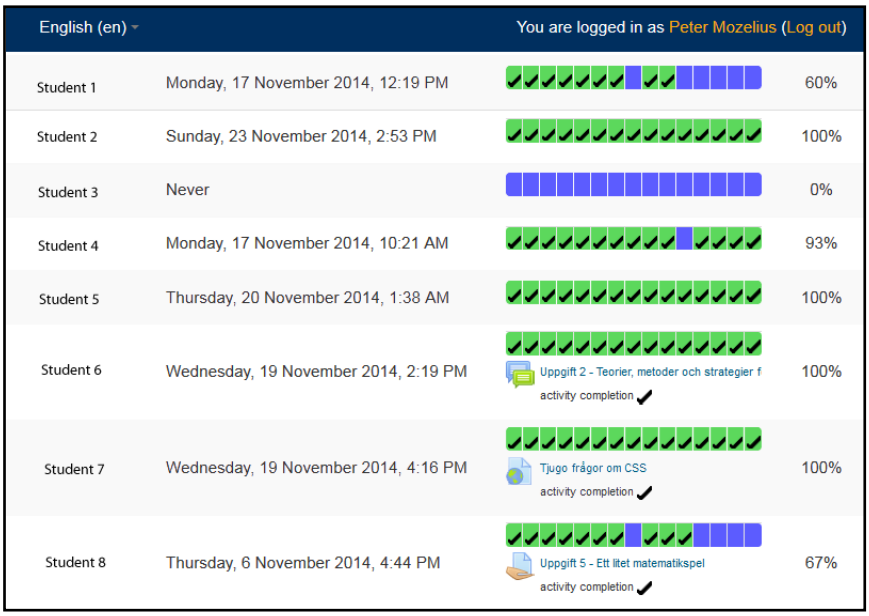

Figure 2: Progress bars in the Moodle virtual learning environment

4.1 The course on e-learning for university teachers

The aim of this course is to teach university teachers about the development of digital learning resources. It has been conducted two times with some enhancements the second time. The courses were mainly held online in the Moodle learning platform with a lot of pre recorded video films, articles and URLs to web resources. The topics discussed were about interaction design, gamification, media production, pedagogy and motivation.

The first course, conducted in late fall 2013, was worth 2.5 ECTS at 25% pace for six weeks. There where three assignments; 1) Course analyses, 2) Concept development/Prototyping, 3) Production. In the production phase the participants could choose between creating a fully working learning resource or a prototype for demonstrating a more advanced concept. The assignments were posted in online forums and peer-reviewed by the participants.

There were two online seminars in Acrobat Connect and an all-day event on location at the department. At the event the participants were introduced to production software like PowerPoint, Camtasia and Articulate Storyline. They also planned and performed talk show interviews in the department's professional video studio.

The course evaluation was positive. All participants thought that as a result of the course they were going to change their way of teaching and that it had been really useful for them. The most negative feedback was that the workload was too heavy for the credits. The analytics from Moodle revealed that the participants only watched a few of the videos before the online seminars and a lot of the activities remained unseen after the course had come to an end.

The second course (fig 3) was conducted late fall 2014 with some enhancements. The credits were raised to 3 ECTS and the time stretched to eight weeks. A progress bar was implemented to indicate the mandatory activities coupled with a course badge as an end goal. To complete an activity and mark it as completed in the progress bar the participants had to answer one to three easy multiple-choice questions about the key concepts. These small quizzes were only used to activate the learners and for diagnostics but not for examination.

The online forums were also connected to the progress bar. First they were marked as completed after the participant had started one discussion and answered two discussions. But this was obvious a design flaw because they could start discussions without submitting any assignment, for example to ask a question and in

the same way they could answer a discussion without posting any peer-review. This was changed to completion by pass/fail instead. The course format was changed to *Weekly format* were the modules are divided into weeks which are marked with a bright colours also indicating progress.

In the first edition of the course the first real world meeting took place the second week during the all-day-event. The two online seminars were conducted week one and three. In the second edition the group didn't meet face to face until the fourth week. Both of the online seminars took place the first and third week. To compensate for the lack of contact with the teacher and the other participants a new introduction video was recorded with the purpose of simulating a sense of presence. The course leader was filmed like he was standing inside of Moodle presenting the course and the graphical user interface talking directly to the user.

Figure 3: Progress bar, completion checkboxes, the badge and the introductory video where the teacher interacts with the graphical user interface

4.2 Software visualisation prototypes

The iteration visualisation prototype and the object visualisation prototype examined in this study are designed and developed by the first author of this paper. Both prototypes will be explained in detail below in section 4.3.1 and 4.3.2.

4.3 Iteration visualisation prototype

The iteration visualisation prototype begins with the display of the class, in which the algorithm is found. The algorithm used is a for-loop that represents iteration. A variable x is declared in the **class For** and later used in the visualisation of the for-loop. This visualisation proceeds with the visualised execution of the for-loop, depicting the operations of the for-loop and the order in which the operations are performed. An illustration of the gradually emerging result of the executing for-loop in the Command Prompt is also illustrated. The

algorithm is gradually visualised from plain program code, to the illustration of the code, to the transformation of the illustrated code, into the illustration of the result of the algorithm.

The window consists of two frames. First, there is one minor frame in the upper left corner. In this frame, the program code, which contains the for-loop is displayed. Second, there is one large frame in the centre, aligned to the right. In this frame, the illustration of the for-loop being processed is presented. The gradually emerging result of the for-loop is displayed in the lower left corner. A coloured square, which corresponds to the UML notation is used, to illustrate the class, in which the for-loop is found.Variables are represented as boxes that hold the values of the variables. The boxes have the same colours as the objects, in which they are found. The names of the variables are placed on the left side of the boxes. The iteration visualisation prototype is complemented with textual clarifications of different states in the iteration visualisation. The textual clarifications are presented in form of small commentary text areas, being visible as the visualisation proceeds. The contours of the text areas and the clarification text written inside the text areas are red. To follow each round (iteration) of the for-loop, the number of each round, is displayed.

The colour of the display, i.e. the numbers of each round is red. Program code in the upper left frame is highlighted with a yellow translucent colour. The highlighting starts from the beginning of the code, row by row, until the end of the code. The parts of the program code that is highlighted for the moment, is the part that is visualized in the large frame in the centre. This is to facilitate the focus of what in the code is being visualised at a certain moment. The iterative process of the actual algorithm is highlighted as well. The highlighting illustrates the operations of the for-loop and the order, in which the operations are performed, when executing the for-loop.

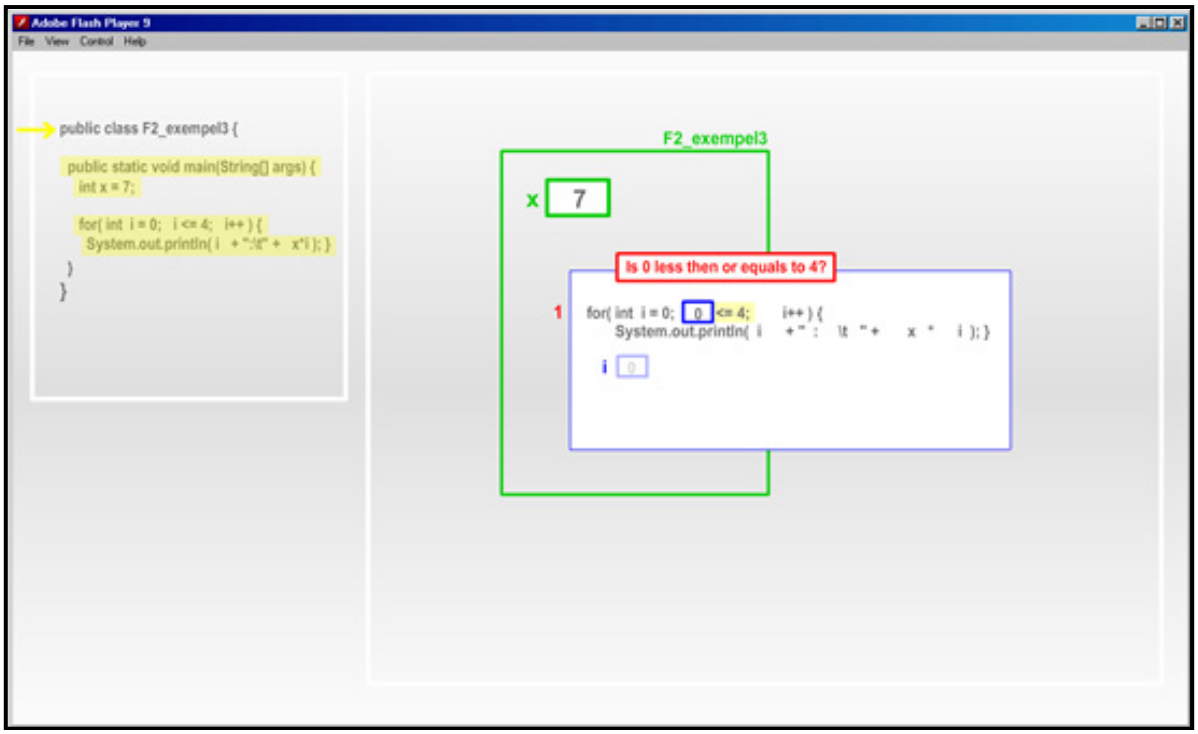

Figure 4: A snapshot of the iteration visualisation prototype at the beginning of its process

4.3.1 Object visualisation prototype

To illustrate the object-oriented view, the object visualisation prototype begins with a display of a core outline of the classes. This is to scope the overall picture of the program's complexity. The object visualisation then proceeds to illustrate the dynamics of the running program. In addition, an illustration of the gradually emerging result of the running program on the screen, i.e. the Graphical User Interface, is illustrated.
Thus, the program is visualized from plain program code, to the visualisation of the code, to the transformation of the visualisation of the code, into the visualisation of the outcome of the program on the screen. The window is composed of two frames. First, there is one minor frame in the upper left corner. In this

frame, the program code is displayed. Second, there is one large frame in the centre, aligned to the right. In this frame, the illustration of the running program is presented. The gradually emerging result of the program is displayed in the lower left corner.

Coloured squares corresponding to the UML notation is used, to illustrate classes in Java. Different tones of the colours are used to indicate correlations between classes and objects. The names of the classes and the objects are placed above the squares, representing classes and objects, in the upper right corner. The representations of the standard classes are coloured grey. Further, they are combined with unfilled arrows, to illustrate Java's inheritance structure. This is also consistent with the UML standard concerning inheritance. Reference variables are represented as boxes holding the references. The boxes have the same colours as the objects in which they are found. The names of the reference variables are placed on the left side of the boxes.

Declarations of reference variables are illustrated as growing arrows. The arrows have the same colours as the classes which contain the reference variables. Further, the arrows are expanding from the inside of the variable boxes, until they reach the instantiated objects. When an object is instantiated, the colour of the origin square, representing the class of which the instantiated object belongs to, becomes intensified. This illustrates that the square is now representing the instantiated object. Methods and constructors of classes and objects, which are not standard classes in Java, are illustrated as solid lined ovals on the left border of those classes and objects. In addition, constructors are filled with yellow colour. Methods of classes and objects, which are standard classes in Java, are illustrated as broken lined ovals.

When a method is called, the contour of the oval, representing the method, becomes coloured like the object in which the method is found. The contour of the oval also becomes solid if the method is located in a standard class. When a method call finished executing, the line of the oval representing the method, remains solid and the colour of the line is diminished, to illustrate the executed method call. The program code in the upper left frame is highlighted with a yellow translucent colour. The highlighting starts from the beginning of the code and proceeds until the end of the code. The parts of the program code that is highlighted for the moment, is the part that is visualized in the large frame in the centre. This is to facilitate the focus of what in the code is being visualised at a certain moment.

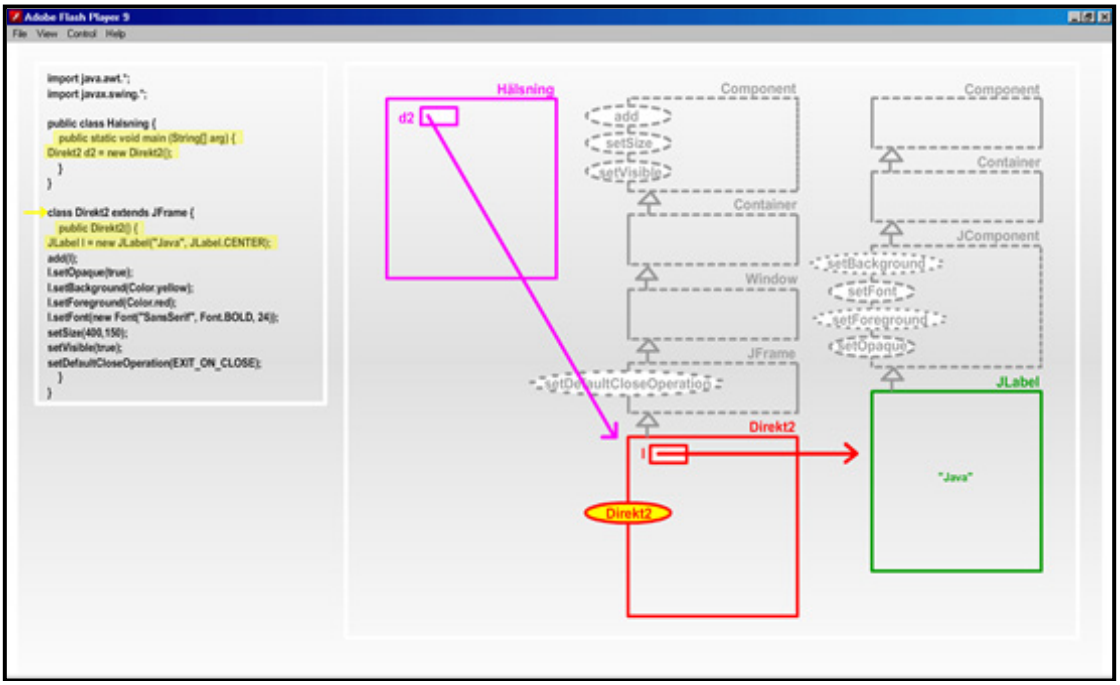

Figure 5: A snapshot of the object visualisation prototype at the middle of its process

5 Results and discussions

Visualisation by progress bars was appreciated in both courses even if all participants did not use them. Digital badges seem to be a more interesting concept in the course where standard grades are absent. In the course on e-learning for university teachers digital badges were a course component that stimulated participants, but in the course on game-based learning the traditional university grades and credits is the main carrot and some students even see the digital badges as redundant and disturbing.

Most of the students thought that the iteration visualisation prototype was a clear way to learn the phenomenon of the loop and it was relatively easy for them to keep the focus all through the prototype. The object orientation prototype was not that clear for the students in general. The students lost more easily focus and the explanation is probably because the object orientation prototype contains more focus points.

5.1 The course on game-based learning

Not all course participants used the progress bars and a majority was indifferent to the opportunity to achieve a digital badge. Considering the progress bars one student answered that *"To be honest, I didn't recognise it before I should answer this question"*. However, most respondents seem to have been aware of their individual progress bar and almost all see them as a positive aid for increased overview. Two answers here were that the progress bar,
"Kept a good control of what activities that remained …" and "Good, you get an overview of your work in the course".

The current standard progress bar design in Moodle seem to work for most students but one respondent thinks that "The idea is good but the problem is that you don't get any feedback on what is missing or what the progress bar represents ...". This is the first version of progress bars in the Moodle virtual learning environment and even if most students understood the feedback idea the design might be improved. From a teachers' perspective all that were involved in the second version of the course posited that it is a useful tool that can facilitate grading and discussions with the students. Two examples when they are of help for teachers are that the progress bars indicates if students have read the grading criteria and show if all postings have been submitted in discussion assignments.

Progress bars were tested in this course with the aim to visualise and not for gamification but in a course on game-based learning students are aware of gamification and one student answered that: *"It gave a good overview and it was easy to check which assignment I'd passed and what was remaining. I don't know if the purpose was to create a game feeling, if this was the case it failed ..."*. In a course on game-based learning where the students play and construct games there are already so many gamified activities that it seemed that even the most competitive gamers were indifferent to digital badges. Some participants had a negative attitude and one student explained in his/her answer that: *"It adds absolutely nothing when the main achievement of a grade is so much more important and stimulating"*.

5.2 The course on e-learning for university teachers

The analytics in Moodle shows that some of the participants, of the first course, didn't even click on the mandatory activities, while all of the participants of the second course not only clicked, they completed the activities on time by answering multiple choice questions.

The course evaluation questionnaire did not reveal much. Most of the five participants thought that the progress bar was good, very good and excellent which can be interpreted as they thought it was useful and motivating. One participant even wrote, *"Good – It really gets me going! You want the bar to be green!"* which indicate that it can enhance students' drive to complete tasks. One participant wrote *"I did not look that much at it"* and one wrote *"Good, a simple way to spur"*.

The progress bar can also be a good way to spur the teachers, as they can see the progress of the participants' paths through out the course content. It can be used to see if the engagement is dropping making it possible to early catch upcoming problems. It can be useful in the flipped classroom concept to check if the participants are preparing themselves before the seminars.

The badge seems less meaningful to the participants but still gets positive remarks according to the course evaluation questionnaire. One participant wrote *"Always nice with medals and swimming badges, but I really don't know what to use it for"*. Another participant wrote "*Fun, but not the reason for me to attend this course :-)"*. The course is voluntary. The credits can be used in promotions but there are other courses that give more credits for less work and there are also mandatory courses for the university teachers. It might be concluded that the participants mainly are intrinsically motivated. The badge can be seen as something new and interesting connected to the gamification discourse but not as a main goal of the course.

5.3 The software visualisations

At the end of each lecture, both the experimental group and the control group answered questionnaires. The questions dealt with how the learning situation was perceived, if it was helpful and if it was focused. Since the difference between the experimental group and the control group only was the additional modal channel, the replies indicate differences in the learning outcome from the different settings.

For the imperative for-loop (iteration visualisation prototype), 61% of the respondents in the experimental group replied that the lecture was a clear and effective way of transferring knowledge. For the control group, only 41% replied the same. Thus, there is a 20% difference in perception of the lecture in that respect. Further, 68% in the experimental group and 39% in the control group responded that they did actually receive help in their perceived understanding of the dynamics by attending the lecture. Finally, 85% in the experimental group and 62% in the control group stated that they understood where the focus of the code executing was all through the animation. These results constitute a significant indication that adding the visualisation channel improved the lecture in terms of student learning outcome. That can be explained in a multimodal perspective as a result of offering differing paths for the student's own design of knowledge, i.e. differing learning design sequences.

For the object-oriented code (object visualisation prototype), 24% of the respondents in the experimental group replied that the lecture was a clear and effective way of transferring knowledge. For the control group, 20% replied the same. Further, 39% in the experimental group and 29% in the control group responded that they did actually receive help in their perceived understanding of the dynamics by attending the lecture. Finally, 60% in the experimental group and 54% in the control group stated that they understood where the focus of the code executing was all through the lecture. These results do not show any significant results implying that adding the visualisation channel improved the lecture in terms of student learning outcome as much as in the first case.

According to multimodal design theory, this can be explained by the differences in the design of the prototypes, where the design of the iteration prototype is of a much simpler graphical kind. The object-oriented prototype contains several focus points, leading to a perceived focusing overload. This overload manifests itself in lower improvements by visualisation assisted learning compared to the for-loop. This is not surprising, since the dynamics of loops, while traditionally being hard to comprehend for novice programmers, is not conceptually hard to understand. Objects, on the other hand, are hard both conceptually and dynamically, leading to their animation being multi-focused, i.e. two kinds of learning complications in the same visualisation.

6 Conclusions

Teachers have for many years noticed that students have different study patterns and the idea that students have different learning styles is old and something that probably was discussed in Ancient Greece (Wratcher et al. 1997). The study patterns are also different in different environments with variations between distance education groups and face-to-face teaching and learning (Diaz & Cartnal, 1999). There is definitely a need for new techniques in e-learning to address the issues of boredom and loneliness in online platforms. Visualisation and gamification seem both to be promising concepts in e-learning worth further exploration, even if neither of the techniques will cure all problems or reach all students.

Visualisation by progress bars seems to be a way to increase students' self-control and facilitate the overview in online environments. Gamification is a trendy phenomenon that, even with careful implementations, probably never will attract all. Since students have different learning styles the solution might be to overload and overlap in courses with the idea that everything not necessarily has to be for everyone.

Even though the badge doesn't seem to be an ultimate goal for anyone in any of the courses it can be used as a fun gimmick creating a positive feeling in the course. Connected to the progress bar you can speculate that it can be a sort of climax to receive the badge when the last sector of the progress bar turns green.

The amendment of programming lecturing with dynamic visual explanations is a promising path. Guided by multimodal design theory, real improvements in novice student's understanding of program dynamics can be obtained. Selander and Kress mean that the choices that each individual is making in a learning process can be looked upon as fixation points. A meaning and a basis for a potential continuation is then temporary locked. What is perceives as an interesting challenge and what catches the attention at a particular time, are two factors which, according to Selander and Kress depends upon what one chooses.

7 Future work

The analytics from the course for university teachers indicates that something in the second course made all of the participants engage earlier and deeper than in the previous course. Was it because of the progress bars, the badge or something else? This spring of 2015 there will be an interview study conducted with the course participants focusing on what makes them engage early in a course.

References

Bíró, G. I. (2014) Didactics 2.0: A Pedagogical Analysis of Gamification Theory from a Comparative Perspective with a Special View to the Components of Learning, In proceedings of *Social and Behavioral Sciences, Volume 141, 25 August 2014, Pages 148-151*
http://www.sciencedirect.com/science/article/pii/S187704281403451X

Brown, K.M. (1996) The role of internal and external factors in the discontinuation of off-campus students. *Distance Education 17*, 44–71.

Chou, S. W., & Liu, C. H. (2005) Learning effectiveness in a Web-based virtual learning environment: a learner control perspective. *Journal of Computer Assisted Learning*, *21*(1), 65-76.

Creswell, J. W. *(*2009*) Research design*: *Qualitative*, *quantitative*, and *mixed methods approaches* (3rd ed.). SAGE ... Thousand Oaks, CA

Deterding S, Dixon D, Khaled R & Nacke L (2011) From game design elements to gamefulness: defining "gamification". In Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments (MindTrek '11). ACM, New York, NY, USA, 9-15. http://doi.acm.org/10.1145/2181037.2181040

Diaz, D. P., & Cartnal, R. B. (1999) Students' learning styles in two classes: Online distance learning and equivalent on-campus. *College teaching*, *47*(4), 130-135.

Eckerdal A. (2009) Novice Programming Students' Learning of Concepts and Practise, PhD Thesis Uppsala University, available at http://www.avhandlingar.se/avhandling/6809751ebf/

Graham, C. R. (2006) Blended learning systems, *CJ Bonk & CR Graham, The handbook of blended learning: Global perspectives, local designs. Pfeiffer*.

Guzdial, M. & Soloway, E. (2002) Log on education: teaching the Nintendo generation to program. Communications of the ACM, 45(4), pp. 17-21.

Hara N. & Kling R. (2000) Students' distress with a webbased distance education course: an ethnographic study of participants' experiences. Information, *Communication, and Society 3*, 557–579.

Huang, H. M. (2002) Toward constructivism for adult learners in online learning environments. *British Journal of Educational Technology*, 33(1), 27-37.

Keller, J., & Suzuki, K. (2004) Learner motivation and e-learning design: A multinationally validated process. *Journal of Educational Media, 29(3*), 229-239.

Keller, J. (2010) Motivational Design for Learning and Performance: The ARCS Model Approach. *Springer*.

Lahtinen, E., Ala-Mutka, K. & Jarvinen, H. (2005) A Study of Difficulties of Novice Programmers, Innovation and Technology in Computer Science Education 2005, pp. 14–18

Lim, D. H., & Morris, M. L. (2009) Learner and Instructional Factors Influencing Learning Outcomes within a Blended Learning Environment. *Educational Technology & Society*, *12*(4), 282-293.

Malone, T. & Lepper (1987) Making Learning Fun: A Taxonomy of Intrinsic Motivations for Learning, In *Snow, R. & Farr, M. J. (Ed), Aptitude, Learning, and Instruction Volume 3: Conative and Affective Process Analyses*. Hillsdale, NJ

Mnguni, L. E. (2014) The theoretical cognitive process of visualization for science education, *SpringerPlus*, *3*(1), 1-9.

Myers, B. A. (1986) Visual Programming, Programming by Example, and Program Visualization: A Taxonomy. CHI'86 Conference Proceedings (59-66). ACM, New York.

Park, J. H., & Choi, H. J. (2009) Factors Influencing Adult Learners' Decision to Drop Out or Persist in Online Learning. *Educational Technology & Society*, *12*(4), 207-217.

Petre, M. (1995) Why Looking Isn't Always Seeing: Readership Skills and Graphical Programming, ACM, New York.

Price, B. A., Baecker, R. M., & Small, I. S. (1993) A Pincipled Taxonomy of Software Visualization. Journal of Visual Languages and Computing 4(3): 211-266.

Remenyi, D. (2012) *Case Study Research*, Academic Publishing International, Reading
Roman, G-C., & Cox, K. C. (1993) A Taxonomy of Program Visualization Systems. IEEE Computer.
Rostvall, A-L., & Selander, S. (2008) Design för lärande, Norstedts Akademiska Förlag, Sweden
Ryan, R. M., & Deci, E. L. (2000) Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being, *American Psychologist, 55*, 68-78.
Selander, S., & Kress, G. (2010) *Design För Lärande – ett multimodalt perspektiv*. Norsteds.
Simpson, O. (2004) The impact on retention of interventions to support distance learning students. *Open Learning: The Journal of Open, Distance and e-Learning*, *19*(1), 79-95.
Tyler-Smith, K. (2006) Early attrition among first time eLearners: A review of factors that contribute to drop-out, withdrawal and non-completion rates of adult learners undertaking eLearning programmes. *Journal of Online learning and Teaching*, *2*(2), 73-85.
Wratcher, M. A., Morrison, E. E., Riley, V L.& Scheirton, L. S. (1997) Curriculum and
program planning: *A study guide for the core seminar*. Fort Lauderdale, Fla.: Nova
Southeastern University. Programs for higher education.
Yin, R. K. (1989 - 2008) *Case study research: Design and Methods*, Thousand Oaks: Sage