UNIVERSITÉ DU LUXEMBOURG

University of Luxembourg

Faculty of Science, Technology and Communication

Incorporating Learning Material and Personalised Feedback into a Serious Mobile Game

Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of *Master in Information and Computer Sciences*

Author
Jo SCHIMBERG

Student Number
0121282714

Date
August 2017

Supervisor
Prof. Dr. Steffen ROTHKUGEL

Reviewer
Prof. Dr. Denis ZAMPUNIERIS

First Advisor
Christian GRÉVISSE

Second Advisor
Dr. Jean BOTEV

Abstract

Game-based student response systems become more and more commonplace in the classroom. They aim at fostering student engagement, group collaboration and participation. However, these systems are usually limited to collecting data from the students and displaying their results in front of the class. They often neglect providing a continuous learning experience, where the learner continues to improve himself after the lecture and to understand his personal difficulties. This constitutes an especially big problem for large lectures, where the individual opinion risks to get lost.

Fortunately, there are personalised response systems that provide individual learning feedback and knowledge growth to learners outside of the classroom.

This thesis presents a personalised mobile learning application that provides performance feedback to the individual learner, as well as learning material support and suggestions once his underlying issues are understood. The application works hand in hand with a classroom tool to enable a continuous learning experience. It benefits from the fact that mobile devices are not restricted to a specific time or location. Its integrated spaced repetition algorithm differs from traditional approaches by focusing on the teaching of learning categories, which can potentially be composed of multiple, interrelated tasks, instead of individual, separated items. The included spaced repetition algorithm represents a trade-off between the best possible learning intervals, which aim to minimise the time spent learning for a maximum learning outcome and the user freedom to decide which topics he wishes to repeat. Moreover, game-based elements are applied to motivate students to use the application and to create an engaging working environment.

Keywords: Game-based learning, student response systems, personalised response systems, learning material support, individual learning feedback, spaced repetition, learning categories, mobile learning

Declaration of Honor

I hereby declare on my honor that I am the sole author of the present thesis. I have conducted all work connected with the thesis on my own.
I only used those resources that are referenced in the work. All formulations and concepts adopted literally or in their essential content from printed, unprinted or Internet sources have been cited according to the rules for academic work and identified by means of precise indications of source.
This thesis has not been presented to any other examination authority. The work is submitted in printed and electronic form.

Luxembourg, August 2017

Jo Schimberg

Acknowledgments

Firstly, I want to sincerely thank my supervisor Prof. Dr. Steffen Rothkugel for his guidance and help throughout my Master thesis, as well as for offering me a student job and being my supervisor of my Bachelor thesis. Secondly, I want to express my gratitude to my first advisor Christian Grévisse for his help and advices throughout this work, as well as for providing me with a LaTeX template. Without both of their support, this thesis work would not have been possible.

Moreover, I want to thank my second advisor Dr. Jean Botev for his guidance, as well as Prof. Dr. Denis Zampunieris for being my reviewer and taking part in my defence board.

Furthermore, I want to express my gratitude to my sister Zoé Schimberg and my two friends Laurent Philippe Hentges and Patrick Zeimes for proofreading my thesis.

Contents

List of Figures

List of Tables

List of Listings

1 | Introduction

A common challenge in education consists in creating an engaging working environment. Often, students either fear or are not motivated to ask questions and to participate in the classroom. Moreover, in large classes, as it is often the case in higher education, this problem becomes even more obvious. Big lectures often implicate large auditoriums with a significant distance between the front and the back of the room, as well as a large number of participants, which results in an impersonal environment. Students might be afraid to actively participate in such an intimidating ambiance (Trees and Jackson, 2007).

As a consequence, teachers often have difficulties to estimate if their students understood a topic or not. Moreover, it is hard for them to figure out which aspects cause the most difficulties to their learners.

Active participation is therefore important in the classroom, first so that students can understand where their problems are and second for the teacher, so that he can better estimate the overall performance level of his learners. However, to find the correct approaches to engage students in classroom discussions is not a trivial task.

The fast evolution of technology offers new possibilities for teachers to foster active participation of their students. Classroom Response Systems (CRSs) become more and more commonplace in colleges and universities. They are often used in quizzes to collect assessment data and to estimate student's attitude in the classroom (Fies and Marshall, 2006). CRSs help teachers to better assess the knowledge base of the class, as well as to improve learners' commitment.

Furthermore, Digital Game-Base Learning (DGBL) is often more effective and motivational than non-gaming approaches and is thus useful for achieving instructional goals (Papastergiou, 2009; Garris et al., 2002). Therefore, modern CRSs often integrate game-based elements.

Overall, a lot of young learners have grown up playing video games, so including what they like to do in their free time in education can help stimulate their motivation. As a consequence, numerous attempts have been made to integrate game-based learning in various areas of education with promising results (Ebner and Holzinger, 2007; Sung and Hwang, 2013; Erhel and Jamet, 2013; Burguillo, 2010; Schmitz et al., 2011). Especially over the past years, the number of published articles on DGBL has significantly increased (Hwang and Wu, 2012).

However, traditional CRSs are solely used in the classroom and often lack an individual, continuous learning experience to support learners at home after the lectures. In the classroom, feedback is usually granted by the teacher, based on the overall performance of the attendees. Especially in larger rooms with many students, like it is often the case for university courses, it is close to impossible to always satisfy the needs of every single learner. Therefore, providing students with an option to repeat the content presented during lectures afterwards at home can be very useful for them to improve their skills and to foster a continuous learning experience. Such personalised response systems may also be useful for people that are not in an educational institution and want to learn for themselves. Moreover, game-based elements can further support them to promote lifelong learning and motivation. This is especially true in the context of personal mobile technologies which support studying anywhere at anytime (Sharples, 2000).

There are multiple personalised response systems that try to achieve this very goal, namely to help individual learners to improve themselves. However, they often offer only limited functionality to help users to enhance their knowledge, as well as to provide them with the ability to understand complex learning topics. This will, amongst other things, be discussed in greater detail in chapter 2.

This work incorporates learning material and personalised feedback into a serious mobile game to help the individual student extend his knowledge base. The applied main use case of the prototypical application is the domain of university computer programming courses. The proposed mobile quizzing application includes spaced repetition to help learners to optimise their time spent learning, while maximising the achieved results. In contrast to traditional spaced repetition algorithms of personalised response systems, which focus solely on the teaching of atomic topics, the presented approach aims at understanding complex, interrelated categories (Schimanke et al., 2013), while leaving a certain degree of freedom to the learner to decide which topics he wishes to revise. Moreover, in order to motivate students to use the application, it includes game-based elements and a large variety of question types.

The mobile application is part of a larger system, called Yactul (Grévisse et al., 2017). The entire system enables a continuous learning experience inside as well as outside of the classroom. However, this work focuses on the enhancement of the Android mobile application component, which targets the personal use and individual knowledge growth. As a consequence, it focuses amongst others on the different types of individuals that exist, in order to meet the needs and requests of most people (Felder and Silverman, 1988).

Chapter Overview

The present work will proceed to explain the spaced repetition learning technique, as well as popular methods and approaches for scheduling repetitions in chapter 2. Moreover, it presents some currently popular CRSs and personalised response systems. It points out the restricted usage possibilities of the latter ones. Chapter 3 presents the main concepts of the developed personalised mobile learning application, as well as its collaboration and partial integration into a broader system. Moreover, it shows challenges and limitations that might generally be encountered in designing personalised mobile learning applications. Chapter 4 demonstrates the implementation and provides a brief overview of the most important aspects and possible future enhancements. Chapter 5 constitutes a summary of the achieved results and presents several possible improvements and extensions to be taken into consideration for future prospects.

2 | State of the Art and Background

Contents

The first part of this chapter presents the spaced repetition learning technique, as well as popular methods and approaches for scheduling repetitions. In the second part, it introduces two of the most widely used CRSs, which intend to foster student engagement and motivation in the classroom. The third part presents some personalised response systems that take benefit of the spaced repetition technique to help individual learners to improve themselves. Finally, the last section highlights the findings from the chapter.

2.1 Spaced Repetition

2.1.1 Forgetting Effect

In 1885, Ebbinghaus (Ebbinghaus, 1913) analysed the existence of the human memory, as well as its effects on forgetting. He memorised a series of nonsense syllables and discovered that the information that the memory can correctly retrieve decreases very quickly at the beginning and slows down over time. More precisely, the decrease has an exponential form with a negative exponent and is represented by the forgetting curve. A typical forgetting curve can be seen in figure 2.1. Its exact form depends on the learner's memory, as well as on the material he wants to study. However, the curve is always based on an exponential function with a negative exponent. The one depicted in the figure is based on Ebbinghaus' collected data. Large losses of content retention for short retention intervals and smaller, but incremental ones for larger retention intervals can be noted. For example: if the studied material is first revised after 20 minutes, approximately 58% of the content is remembered, whereas if it is reviewed only after a full day, still more than 30% of the material is retained in memory (Custers, 2010).

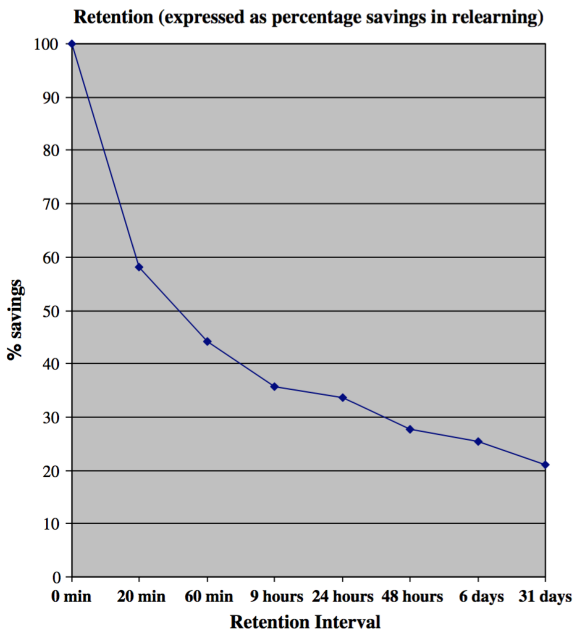


Figure 2.1: Example of a typical forgetting curve based on Ebbinghaus' original data (Source: (Custers, 2010))

In 1985, Loftus (Loftus, 1985) evaluated the forgetting effect for different degrees of initial learning. He analysed whether an intensive initial learning session results in a slower decrease of the forgetting effect, compared to a less intensive one. He evaluated multiple forgetting curves on a variety of data and came to the observation that forgetting is indeed slower for a higher intensive primary learning session.

Moreover, Rohrer et al. (Rohrer et al., 2005) analysed the effect of overlearning on the memory. They explain that it consists in continuing to repeat learning content in one and the same study session even after one perfect trial. They performed practical experiments on college students to find out the effect of overlearning on long-term retention. Their results showed that students that used overlearning in their study sessions were able to recall the learnt material better after one week,

but that this difference compared to the other learners decreased very fast for longer time intervals. As a consequence, learners can really benefit from over-learning if they need to remember learning content for several days, but do not benefit from a significant advantage for longer retention periods. Therefore, this strategy is only viable for students that need to study for example for an exam taking place within a couple of days.
However, applying repeated spaced repetitions over multiple learning sessions increases the strength of the memory (Edge et al., 2012), as depicted in figure 2.2. The figure shows that repeated spaced repetitions of an item result in a more gradual decrease of the forgetting effect and in an increase of the chance to correctly recall an item. Over time, as the strength of the memory increases, the repetition intervals become larger and larger. Spreading studying over multiple learning sessions is therefore more efficient than repeating the same items over and over again in one and the same session.

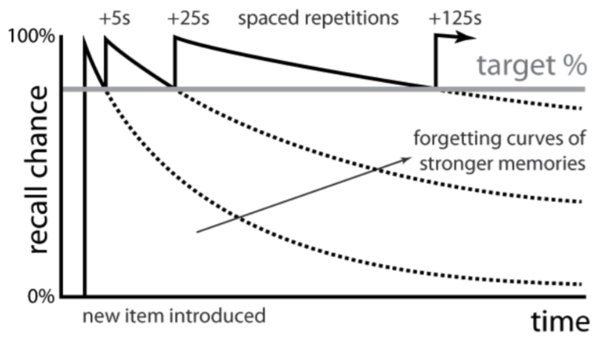


Figure 2.2: Applying repeated spaced repetitions to the forgetting curve (Source: (Edge et al., 2012))

As everyone's time is valuable and limited, minimising the time spent studying while still remembering a maximum of learning content can be helpful to a lot of people. Bahrick and Phelps (Bahrick and Phelps, 1987) defined the concept of optimum repetition intervals as being the longest possible intervals that do not result in retrieval failures. These optimum intervals must be determined empirically and depend on the person, as well as on the studied material. Spaced repetition can be useful to find these optimum intervals.

2.1.2 Popular Methods and Approaches

Pimsleur Method

In 1967, Pimsleur (Pimsleur, 1967) presented the Pimsleur method which is based on repeated spaced repetitions. It is primarily targeted to studying foreign languages, with a special focus on learning and remembering a large amount of vocabulary and structures. This method is solely based on oral recalls, and can for example be used by teachers. In the classroom, they can orally teach vocabulary of a foreign language. The Pimsleur spaced repetition procedure is called "graduated interval recall" and is based on the fact that if a student gets reminded of a word before he completely forgets it, his chances of remembering it the next time will augment. Moreover, each time the learner gets reminded of a word, the subsequent forgetting interval will increase in time.

The method uses a schedule which takes an exponential form and is based on Pimsleur's programming experience and the findings from experimental psychology. This exponential schedule has to be adapted to the words that need to be studied, as well as to the learner himself. The speciality of this spaced repetition approach is that it is characterised by very short repetition intervals, in the order of seconds, for the first repetitions. Nonetheless, it is important that these recalls are separated by other activities to distract the student's mind from the word. For example if the first repetition interval is $4^1 = 4$ seconds, then the second interval is $4^2 = 16$ seconds, the third one $4^3 = 64$ seconds and so on. The first interval is determined by an arbitrary probability value that the student still remembers the word. This value can for instance be defined as a 70% chance that the learner retains the item in memory.

A disadvantage of this spaced repetition method is that the base of the used exponential function is kept constant and relies solely on the forgetting time of the first interval. Its value stays the same for subsequent repetition intervals and only the value of its exponent changes. It is therefore not a very flexible approach. Moreover, there is no distinction in time between repetitions if the item is correctly remembered or not. Finally, as the Pimsleur method is solely based on oral recalls, the total amount of available time is often very limited. If the method is for example applied by the teacher during lectures, the overall time, as well as the moments where he can use it, is constrained by the time slots of the lectures (Edge et al., 2012).

Leitner System

Developed in the 1970s, the Leitner system (Leitner, 1995) also uses spaced repetition. Leitner originally designed his system for paper flashcards. His goal was to decide when each card should be repeated. The schedule that he defined is based on several boxes of flashcards. Each box has a different size, which represents how well the learner knows its content. The first box has the smallest size. In the beginning, when the learner starts making new cards, he puts them all in the first box. As soon as it is entirely filled with cards, he starts repeating most of them. If he recalls the content of a card correctly, he moves it forward to the second box, which is larger in size. If he does not recall what is written on a card, he puts it to the back of the first box.

As soon as the second box is completely filled with cards, the student repeats some of them. If he remembers them, he moves them to the third one, which is again a little bit larger in size. If he does not remember them, he moves them to the back of the first box. The same counts for the third box. As soon as it is entirely filled, the learner repeats some of its contained cards. If he does not recall them, he moves them to the back of the first box and so forth. As the first box is the smallest one and contains new flashcards, as well as cards that the learner did not remember, they will be the ones that will be repeated most frequently. An example of such a paper-based flashcard system is depicted in figure 2.3.

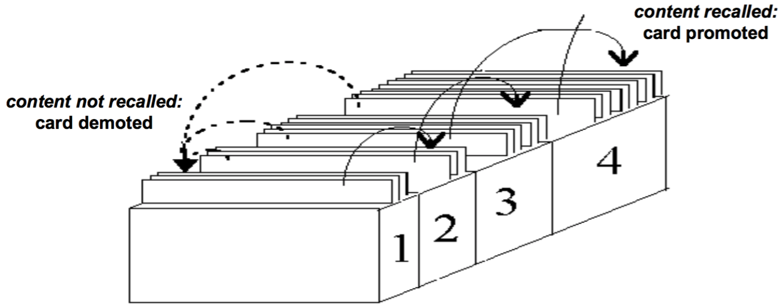


Figure 2.3: Paper-based Leitner system with different box partitioning sizes (Source: (Mubarak and Smith, 2008))

Compared to the Pimsleur method, the Leitner system is more flexible, as the learner can always decide when he wants to study and is not limited to, for instance, the time spent in the classroom in the presence of his teacher. Moreover, the Leitner system differentiates between correct and wrong recalls, which is not the case for the Pimsleur method. However, it does not take into account other factors, such as the amount of time that the student needed to recall a card.
Other versions of the Leitner system exist as well, where flashcards are not moved to the first box if they are not correctly remembered, but only one box backwards. Yet an additional version, which can also be used for electronic flashcards, is described in (Edge et al., 2012). It uses numbered flashcard piles P_1, P_2,... P_N, where each subsequent pile contains flashcards of an increased inter-repetition interval. As illustration, on day D, all flashcards of piles P_x need to be repeated, where D must be a multiple of x. For example on day 3, the student needs to repeat all flashcards from piles 1 and 3. If the cards are not correctly remembered, they are returned to the first pile, else they advance to the subsequent one.

Neural Networks

Piech et al. (Piech et al., 2015) used a machine to model the estimated knowledge state of a student while he was studying for a course. They performed a utility study to analyse student learning with recurrent neural networks (RNNs). They used two different types of RNNs, a simple and a more complex one, to predict student responses to exercises, based on their past results. Given the estimated knowledge state of a student at some point in time, their RNN can potentially determine which exercise should be repeated next. They fed the network with each

possible next exercise and calculated what the resulting expected knowledge state of the learner would be, given that choice. The next exercise that was chosen was the one that was expected to help the student the most to improve himself.
The advantages of their model are that it does not require expert annotations, as it can learn patterns in the data on its own. Moreover, it can operate on any input data that can be vectorized. Being able to tell which exercise is the most valuable for a student to repeat next, can be very useful for spaced repetition. Moreover, Piech et al. explained that RNNs could be used to demonstrate how students forget. However, they need large amounts of training data and are therefore not suited to be used in a classroom. They fit better into large online education environments, such as the Khan Academy (Khan Academy, 2017), with whom Piech et al. worked together and from which they received a large dataset.
In general, neural networks require a large amount of training data and are not suited to be used on a small scale, such as for example in a classroom. Additionally, the training of a neural network requires a lot of computational power and is therefore not always realisable.

2.1.3 SuperMemo Method

The SuperMemo method (Wozniak and Gorzelanczyk, 1994) relies on spaced repetition as well and is especially designed for computer-assisted implementations. It targets mainly the scheduling of optimum repetition intervals for digital flashcards. Wozniak and Gorzelanczyk defined an optimum interval $I(EF, R)$ for the repetition R, as being the product between the previous optimum interval $I(EF, R-1)$ and an optimal factor $OF(EF, R)$. The optimum interval is expressed in days and EF represents the "easiness factor" of the learner to remember the item for repetition R. To determine the Optimal Factors (OFs), an optimisation algorithm was used with the goal to achieve a knowledge retention rate of 95%. Wozniak and Gorzelanczyk defined two different classes of OF values. In the first one, the OFs were initialised with $OF(EF, R) = 1.5$, for all EF and R values. It was intended to verify the validity of the optimisation algorithm. The second one was determined in function of EF and R, where EF had a value between 1.3 and 3.2. It was used to ensure a faster convergence of the initial OFs to their optimal values. Some of the predetermined OF values established in the second version can be seen in figure 2.4. For the first repetition, they are always equal to 5.

Predetermined matrix of optimal factors (EF - easiness factor, R - repetition number)													
Optimal factors (OFs) before the experiment													
R\EF	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0	2.1	2.2	2.3	2.4	2.5
1	5.00	5.00	5.00	5.00	5.00	5.00	5.00	5.00	5.00	5.00	5.00	5.00	5.00
2	1.30	1.40	1.50	1.60	1.70	1.80	1.90	2.00	2.10	2.20	2.30	2.40	2.50
3	1.30	1.40	1.50	1.60	1.70	1.80	1.90	2.00	2.10	2.20	2.30	2.40	2.50
4	1.30	1.40	1.50	1.60	1.70	1.80	1.90	2.00	2.10	2.20	2.30	2.40	2.50

Figure 2.4: Optimal factors in function of R and EF (Source: (Wozniak and Gorzelanczyk, 1994))

Based on their discovered values for their optimal factors, Wozniak and Gorzelanczyk defined a formula to determine the optimum repetition intervals that lead to a knowledge retention rate of 95% as:

$$I(EF, 1) = 5 \quad \text{(for R = 1)}$$

$$I(EF, R) = I(EF, R-1) \times (EF - 0.1 + e^{-2.3 \times R + 5}) \quad \text{(for R} \neq 1)$$

The value of the easiness factor usually lies between 1.3 (the most difficult items) and 2.8 (the easiest items) in this case. To determine the exact value, the students needed to evaluate their responses by themselves, based on a 0-5 grade scale, where 0 expressed very bad and 5 excellent.
Several algorithms of the SuperMemo method exist, including the popular SM-2 algorithm (Wozniak, 1998). It is one of the earliest implementations of the SuperMemo method and has inspired many other algorithms in different applications due to its efficiency and simplicity. It is used amongst others in Anki (Elmes, 2017) and Mnemosyne (Mnemosyne, 2017). In the SM-2 algorithm, the value of the easiness factor is calculated as:

$$EF' = EF + (0.1 - (5 - q) \times (0.08 + (5 - q) \times 0.02)) \quad \text{(for q} \geq 3)$$

The variable q expresses the quality of the response and is measured on the 0-5 grade scale. Its value is usually based on self-assessment. EF' is the new value of the easiness factor and EF the old one. EF has a lower bound of 1.3 and is initialised with 2.5. Moreover, its value will only be updated if the quality q is greater or equal to 3 (correct reply).
The optimum repetition interval *I* after the *n*-th repetition is determined as:

$$I(n) = I(n-1) \times EF \quad \text{(for n} > 2)$$

It defines the subsequent repetition interval and is expressed in days. The only exceptions are that if n = 1, then $I(1) = 1$ and if n = 2, then $I(2) = 6$.

Similar to the Leitner system, if the learner does not know an answer ($q \leq 2$), then repetitions restart from the beginning for that particular item ($n = 1$), without changing the EF value.
Both, the optimal factors in (Wozniak and Gorzelanczyk, 1994), as well as the SM-2 algorithm in (Wozniak, 1998) were determined heuristically. They are therefore not guaranteed to be optimal, but have proved their usefulness in practice, especially the SM-2 algorithm, which is widely used.

2.2 Classroom Response Systems

A classroom response system, also commonly referred to as a student response system (SRS), is a system in which students usually have to answer multiple-choice questions by means of an electronic device. Questions are often grouped together to form a quiz. In early CRSs, students used small and simple devices, which are commonly referred to as "clickers", to answer questions asked by the teacher (Nielsen et al., 2013). As the name suggests, CRSs are used to collect, aggregate and display the learners' results in the classroom (Siau et al., 2006). More precisely, the instructor shows the questions in front of the class, usually on a projector screen and students provide their answers on their device.
With the bring your own device (BYOD) wave (Wang, 2015) and the advances made in technology, learners can bring their own devices, such as smartphones and laptops to the classroom. They do not need to rely on the simple clicker tools anymore. The advantage is that modern devices have more computational power than the clickers and teachers can ask students to perform a diversity of tasks. Therefore, question types are no longer limited to simple multiple-choice questions.
Moreover, the BYOD wave makes it easier to apply gamification to SRSs. Gamification is the process of using video game elements in non-game applications with the goal to improve user experience and to create engaging workplaces. Typical game elements are for example points, badges and leader boards that are used to create reward and reputation systems, as well as to facilitate large collaborations between students (Deterding et al., 2011). However, introducing such elements does only increase the likelihood for a gameful experience, its occurrence is not guaranteed by any means (Huotari and Hamari, 2012).
In the following, two popular modern Game-based Student Response Systems (GSRSs) will exemplarily be presented. There are a number of other comparable systems such as Poll Everywhere (Poll Everywhere, 2017) and Verso (Verso Learning, 2017), which will not be the subject of this work.

2.2.1 Kahoot!

Kahoot (Kahoot!, 2017) is a popular GSRS that can be used in the classroom. Teachers can create a "Kahoot", in form of a *quiz*, *survey*, *discussion* or *jumble* to motivate students to test and improve their knowledge by means of a competitive game. Only instructors need to create an account. Students can join a Kahoot with any device that allows for web browsing, such as a tablet, laptop or smartphone (Dellos, 2015). All they need to do is to provide a pin to join the quiz, as well as to decide on a nickname. This guarantees anonymity. Instructors can also use a Kahoot that was created by another person and was made publicly available. Learners can either compete against each other in the "Classic" mode, or team up and use a single device to give one answer per team in the "Team mode". Once a quiz has been started, the teacher's screen, which is intended to be shown via a projector to the students, shows the questions with the remaining time, as well as the answers. Students can see the proposed answers on their device, but neither the questions, nor the remaining time, nor the correct answers. This information is displayed on the teacher's screen only.
If the instructor creates a classical *quiz*, he can provide up to 4 answer possibilities, within which he can select multiple ones as being correct. However, the student can only choose one of them. The *discussion* mode is similar, but there is no correct nor wrong answer. Moreover, it includes a single question only. The *survey* mode can include multiple questions, but there are no correct or wrong answers. Finally the last mode is *jumble*, where the student needs to put 4 provided answers in the right order. The *team* mode is not yet available for the latter one. The two modes *quiz* and *jumble* use a score and leader board to motivate students to compete against each other, as well as to foster gamification. The calculated scores are based on whether the provided answer is correct or not, as well as on the speed of the reply.
One of the main advantages of Kahoot! is the flexibility it provides by using a web browser and thus the wide range of devices it supports. Moreover, it is easy and fast to deploy in the classroom, since teachers can create quizzes in advance and students do not need to create an account. Finally, instructors can also save the results of a quiz, which may help them to prepare course material for upcoming lectures.

2.2.2 Socrative

Similar to Kahoot!, in Socrative (Socrative.com, 2017), teachers need to create an account, students do not. They can simply join a room that was created by the instructor with any personal device that uses a web browser. Socrative also includes multiple modes, namely *quiz*, *space race* and *exit ticket*.

In the first two, the teacher can decide between several sub-modes, namely *instant feedback*, *open navigation* and *teacher paced*. In the *instant feedback* sub-mode, students answer questions and get instant feedback. They cannot change their provided answers once confirmed. Note that these are the default settings, the instructor can for example also decide to not provide feedback to the learners. In the *open navigation* sub-mode, students can decide in which order they want to reply to the questions. They might even come back to a question later on, before they submit all their answers at once. In the *teacher paced* sub-mode, the instructor controls the speed and flow of the questions. This can be useful if he wants to explain the correct answer before switching to the next task. Note that none of the sub-modes includes a time limit, but the instructor can finish a quiz at any time. A created quiz can be used in both the *quiz* and the *space race* modes and can include several types of questions amongst *multiple choice*, *true/false* and *short answer*. The difference between the two modes is that in the *quiz* mode, a student provides answers for himself, whereas in the other mode, teams of learners can compete against each other and can see their progress, for example in form of a rocket, progressing along the teacher's screen. Each student needs to reply on his own and gets, depending on the setting, either randomly assigned to a team or may decide for himself which one he wants to join. As soon as one member provides a correct answer to a question that has not been correctly answered before, the associated team can observe its progress on the instructor's screen. If one team gets all the answers correct, their rocket will arrive at the finish line. This mode supports the gamification aspect of Socrative very well.
In the *exit ticket* mode, the learners need to provide a quick feedback on how well they understood the presented material. Finally, teachers can view and save a report of the results.

2.3 Personalised Response Systems

In CRSs, feedback is usually directly provided by the teacher during lectures. They are designed to be easy and fast to use in the classroom environment. Moreover, they target groups of students and aim at providing the best possible feedback directly after a quiz was played. They do not intend to tell the individual learner when he needs to repeat which questions of the quiz again. This is supposed to be decided by the teacher.
In contrast, personalised response systems focus on individual learners and their personal knowledge growth. Their purpose is to help students to improve themselves over a larger period of time and to inform them when they need to repeat a question again. Moreover, by means of spaced repetition, they often try to minimise the time spent learning, while maximising the amount of information

that is retained by the individual learner. In order to do so, personalised response systems apply algorithms to optimise the intervals between repetitions. In this section, an overview of some of the most popular personalised response systems will be provided.

2.3.1 SuperMemo World

SuperMemo World (SuperMemo World sp. z o.o., 2017) promotes the SuperMemo method. In the late 1980s, it was the first company which applied computers to optimise learning intervals between repetitions by means of spaced repetition. Currently, it is the world leader in research on human long-term memory. SuperMemo World offers a collection of over 100 courses, but most of them require premium access and are not free of charge. Courses are made up of flashcards that may contain images. Moreover, the voice synthesizer of a device can be used to speak the questions and answers on cards out loud. Members can also create and share their own flashcards for free with other users and take benefit of flashcards created by them.
When a card gets presented to a learner, he needs to figure out the correct answer that is written on its back. Once he thinks he knows it, he may flip the card over to look at the solution. He needs to indicate if he either knew, almost knew or did not know it at all. The next time to repeat the card will then be scheduled according to this self-evaluation. The self-assessment grade scale is similar to the quality of the response of the SM-2 algorithm. The difference is that the latter one is based on a grade scale from 0-5, instead of a scale that is made up of 3 values only. It is worth mentioning that there is no time limit until when the learner has to flip over a card, which results in a less stressful learning experience. Furthermore, SuperMemo World offers two different options to consider course material. The first one is to use a device that has a web browser and an Internet connection. The second one is to download the application that SuperMemo World offers either for Microsoft Windows, Android or iOS. The advantage of using the application over a web browser is that it enables users to study both online and offline. This is especially beneficial when no Internet connection is available, or when travelling abroad. Sharing learning content between multiple devices is also supported.

2.3.2 Anki

Anki (Elmes, 2017) is another personalised response system that is based on flashcards. It can be used by anybody who needs to remember a lot of content. Its spaced repetition algorithm relies on the SuperMemo SM-2 algorithm. It is

available for Windows, MacOS, Linux, iOS and Android. Only the iOS application is not free of charge.
In Anki, after the learner thinks he knows the solution of a card, he needs to tell the application how well he remembered it and the system schedules the next repetition based on this information. The intervals for the different possible schedule choices are customisable. The learner can for example define the repetition interval for a card that he easily remembered. However, the application dictates when it is time to repeat an item. In the standard mode, if no cards for a deck are due yet, the learner cannot decide that he wants to repeat the content anyway. If he wishes to do so, he needs to create his own, custom study programme, where he can for example increase the amount of new cards for a given day, or review cards ahead. Overall, this approach does not seem to be very flexible. Moreover, if the learner did not study with Anki for several days or even weeks, he risks to get overwhelmed by a large amount of cards that are overdue, which can lead to frustration. On the other hand, there also is a risk that the learner may run out of cards. Therefore, Anki is well suited for large decks with a lot of content, where the learner either always has some cards to review, or new cards to study. A field where the application can be used with great success is for studying new languages, as this task involves recalling a huge amount of vocabulary. As Godwin-Jones (Godwin-Jones, 2010) pointed out, Anki uses the notion of "fact", which is a vocabulary item with its definition. It is useful for studying such facts in general, but not for learning complex, interconnected topics.
Nonetheless, advantages of Anki (Elmes, 2017) are that the cards support multiple data sources, including images, audio sources and videos. To include an audio voice can be especially useful when studying the pronunciation of vocabulary of a foreign language. Another advantage is that flashcards can be synchronized across multiple devices, which enables to study anywhere at any time. Moreover, to get started easily, it is possible to download decks that have been shard by other customers. Furthermore, there is a large amount of additional add-ons available that were created by other users, which enable a customised Anki experience. Finally, it also provides individual statistics.

2.3.3 Mnemosyne

Another popular free of charge flashcard system is Mnemosyne (Mnemosyne, 2017), which is also based on the SuperMemo SM-2 spaced repetition algorithm. However, the authors made modifications to the original algorithm to better deal with early and late repetitions. Moreover, Mnemosyne adds some randomness to the repetition intervals. The developers use the SM-2 algorithm over newer versions, because they are sceptical about whether or not the new SuperMemo algorithms provide relevant practical benefits overs the SM-2 algorithm. Fur-

thermore, the complexity of the newer versions is much higher. Mnemosyne is collecting anonymous data from their learners in order to improve the efficiency of their algorithm. Their card grade scale, which is based on self-assessment, ranges from 0 to 5. This is exactly as in the SM-2 algorithm. In Mnemosyne, the values 0 and 1 express that the card is new or that it has been completely forgotten. Grade 1 cards will be repeated less often, however cards of both grades will reappear repeatedly in the same study session until being rated with a value of at least 2. Latter cards will be presented within one or more days, depending on their grade. The higher the provided rating, the less frequently they will be shown.
Members can also share cards with the Mnemosyne community. Moreover, adding images, sounds and videos on cards is also supported, as well as synchronisation of learning material between different devices. Furthermore, learners can assign multiple tags to cards, to be able to sort them by categories. This is especially useful when studying a subset of cards from a very large set.
3 standard card types are available: *front-to-back only*, *front-to-back and back-to-front* and *vocabulary*. The *front-to-back only* type consists of a simple question-answer pair that goes only in one direction. *Front-to-back and back-to-front* is a simple question-answer pair that goes in both directions. The *vocabulary* type is a more complex variant of the *front-to-back and back-to-front* card type, which includes pronunciation and notes as additional optional fields. Furthermore, there are a few plug-ins to provide more card types, such as *map* and *sentence*.
Overall, Mnemosyne and Anki rely on the same principles and appear to be very similar.

2.3.4 Flashcards Deluxe

Flashcards Deluxe (Thomason, 2017) offers two spaced repetition study modes. The first one is based on the Leitner system and the second one is inspired on the algorithms used in SuperMemo and Anki.
The Leitner based mode should be used by the learner if he needs to remember learning content over a short period of time, within a couple of weeks. It is based on the number of repetitions per card. Each card includes a winning streak value, which will be reset to 0 as soon as the learner does not know the answer. If he gets it correctly, the value will be augmented by 2 for a "strongly correct" reply and by 1 otherwise.
On the other hand, the second mode is based on time intervals, instead of number of repetitions. This mode is generally more efficient for long-term retention, as well as if the user wants to learn a large amount of information. In this mode, a card reply is evaluated on being either wrong, correct with some hesitation, or correct without hesitation. The next interval will be scheduled according to this evaluation. In both modes, it is the application that decides when the learner

should revise a card.
Flashcards Deluxe is available for iOS and Android but only a light version is free of charge. It includes a wider range of question types than the previous applications. In addition to traditional flashcards, it supports, but is not limited to multiple choice questions, spelling and drawing tasks. Moreover, it supports flashcards that are made up of 3 sides instead of the classical two-sided cards.
The student can divide cards into categories in order to form sets. This has the advantage that, if there is a large amount of cards available, then the learner can rely on these smaller sets for his study sessions, which is more manageable. However, each card is considered as an independent task. Assigned categories to it are not taken into account when scheduling its next repetition.

2.3.5 Gradint

The Gradint programme (Brown, 2017) uses a variant of the Pimsleur spaced repetition method. It is available amongst others for Microsoft Windows, Mac OS X, Linux and Android. It is solely based on audio recordings, which fits the underlying spaced repetition method perfectly.
As in the original Pimsleur method, Gradint does not make a time distinction between repetitions whether the learner correctly recalls a term or not. Its main use case is learning the pronunciation of vocabulary and phrases of a foreign language. Learning material is taught by means of audio flashcards. Users need to feed the programme with words and phrases of both languages. They can either rely on the voice synthesizer of their device or record the voice of a person, which can be the learner himself, or somebody that speaks the language in question very well, or even the chance given, a native speaker. An advantage of audio flashcards over traditional cards is that they can be listened to while performing other daily tasks. However, an obstacle of using Gradint is that the recording, as well as the creation of diverse audio flashcards requires a large amount of initial time investment. Relying solely on the voice synthesizer is not a good idea, since it does not always articulate the words very well. Moreover, the ideal solution, to record a native speaker, is not always feasible, or the time period, within which such a person is available, is limited. This can constitute a limiting factor in the learning progress, as described by Brown and Robinson in (Brown and Robinson, 2003). Nonetheless, they mention in their work that it is hard for a person with print disabilities, such as dyslexia, to learn a new language, since most courses rely mainly on written documents and not on audio sources. Therefore, programmes like Gradint, which are solely based on audio recordings, can be helpful to these users. Moreover, as the programme targets learners with low vision, it includes a separate mode that makes all the fonts of the user interface bigger.

2.4 Chapter Conclusion

This chapter explained that spreading learning content over multiple learning sessions is, in terms of time investment and learning results, more efficient than studying the whole material in one single session. Spaced repetition is an effective medium to find the optimum repetition intervals and to strengthen the human memory. Moreover, overlearning can be of great benefit when the learning material needs to be retained for several days only. The chapter introduced several popular approaches and methods of spaced repetition.

Furthermore, it presented some CRSs, as well as personalised response systems. CRSs do not urge to integrate spaced repetition, as they are usually used in the classroom at one specific moment in time and immediate feedback is granted by the teacher. In the early CRSs, where students used clickers, it was hardly possible to integrate personalised feedback, due to the simplicity of the devices. Only with the arrival of the BYOD wave, more sophisticated devices could be used to answer questions. But this did not change the fact that CRSs were still mainly used in the class and thus did not need to integrate spaced repetition.

On the other hand, the personalised response systems target individual learners rather than groups of students. They are meant to be used on a regular basis and take advantage of spaced repetition to schedule the best time for users to revise learning material. However, all of the presented personalised response systems focus on teaching vocabulary or simple atomic, independent tasks in general. Most of their algorithms are based on the SuperMemo SM-2 algorithm and none of them promote studying of more complex subjects that can be classified into categories. The important difference is that for learning atomic topics like vocabulary, every word must be remembered for itself, in contrast to learning subjects, grouped into categories. The latter requires understanding of more complex, possibly interconnected study fields. This has already been pointed out by Schimanke et al. in (Schimanke et al., 2013). Moreover, they explain that to apply the existing spaced repetition algorithms on such a categorisation needs a certain degree of adjustment to determine the best possible learning intervals. Furthermore, as they mention that when using flashcards with vocabulary, content creation does not take a lot of time, which is not the case for more complex subjects, where content production usually takes more effort. Less content implies more repetitions of the same learning material.

The personalised response systems that were presented in this chapter do all mainly focus on the use of flashcards. There definitely is a lack of integration of spaced repetition into more complex platforms. To be more precise, there are no platforms which support the understanding of categories, where multiple questions can be related to each other, rather than simple, atomic tasks. Furthermore, there are no systems which promote a wider range of tasks that can be performed by the student in order to keep his interest. Learner motivation can be improved amongst other things by including more game-based elements. Finally, some applications, such as Anki and Flashcards Deluxe use very strict time intervals, which do not leave a lot of freedom to the learner to decide when he wants to revise learning material. This can either result in too much learning content that needs to be repeated or nearly nothing at all.

Furthermore, neither the CRSs, nor the personalised response systems include learning material in form of document support to propose to the individual learner when trying to understand his difficulties. This could potentially help him to improve.

The next chapter proposes a personalised mobile response system, that fosters understanding of categories, includes document support and integrates spaced repetition. Moreover, it allows playing a multitude of different question types. It tries to reuse the positive aspects and ideas of this chapter, as well as to find a solution for the mentioned shortcomings.

3 | Personalised Mobile Learning

Contents

Mobile learning (m-learning) offers new possibilities to extend the abilities of electronic learning (e-learning). It does not have to deal with restrictions related to a specific time or location. Therefore m-learning offers new dimensions for education (Bartel and Hagel, 2014).

This chapter presents the main concepts of a mobile game-based learning application that takes benefit of the opportunities offered by m-learning in order to provide a personalised learning experience anywhere at anytime.

The first part defines the requirements that take into account the findings, as well as shortcomings from the previous chapter. Then Yactul, an existing system that is composed of a classroom tool and two mobile learning applications, is presented. Afterwards, a lightweight modular ontology is introduced. It is used to include document support in the mobile application. Section 3.4 explains how feedback is provided to an individual student in form of document support, as well as of a personalised performance measure. In section 3.5, the two different game modes that allow a personalised mobile learning experience are explained. The last section introduces common challenges and limitations that are encountered in the context of a personalised mobile learning application.

3.1 Requirements

This work presents a personalised mobile response system, targeted primarily to university courses. It incorporates learning material, as well as personalised feedback.

More precisely, its requirements are to extend an already existing Android prototypical mobile application for individual offline learning inside a game-based student response framework to provide both documents support, as well as performance feedback to an individual user. It shall include spaced repetition in the Coaching mode to help the student improve himself and suggest questions to him that he needs to revise. The solution should try to reuse the best ideas from existing personalised response systems, while improving on their shortcomings. Being targeted to university courses, a special focus shall be given on promoting the understanding and classification of complex topics and categories rather than simple, atomic topics, like for example vocabulary (Schimanke et al., 2013), as courses may include complex, interrelated topics. This is in contrast to previous state of the art personalised response systems. Moreover, the proposed solution shall allow the learner to quickly and easily see his progress. To help a student to better understand his problems and to improve himself when he does not know the answer to a question, the solution shall integrate and suggest related learning material to him in form of document support, once his underlying difficulties are understood. Therefore, it shall provide a mapping between learning aspects of the proposed questions and the corresponding learning material. It shall focus on the individual knowledge growth, while giving the learner a certain degree of freedom to decide which questions and learning material he wishes to repeat. The feeling of an individual learning experience shall not be neglected. Moreover, in order to motivate the learner to use the application, a special focus shall be given on applying gamification techniques to enhance the learning experience.

The main use case that shall be used for the courses, quizzes, questions and documents inside the prototypical application shall be the domain of computer programming. However, the application should be extensible to other domains of university courses and even later on to other learning institutions.

3.2 Yactul

This section presents Yactul (Grévisse et al., 2017), an existing and extensible game-based student response framework, used to create and play quizzes in a university context. This work enhances the mobile application, which forms a part of the Yactul system, to integrate document support and suggestions. Moreover, it designs a new, enhanced version of the already included simple spaced repetition algorithm from the iOS mobile application and includes it in the Android application. However, this section focuses on the state of Yactul before these modifications are made. Subsequent sections explain how the document support, as well as the new spaced repetition algorithm are created and integrated.

3.2.1 Overview

One of the most important goals of the extensible GSRS platform Yactul is to combine the best ideas from existing popular GSRSs such as Kahoot! and Socrative, while overcoming their shortcomings. Yactul seeks to create a game-based student response framework that fosters an active and continuous learning experience inside, as well as outside of the classroom. Measurement of the individual knowledge growth, as well as of the progress of the whole class is important in order to support a continuous learning style. It is composed of a classroom tool, which runs on a web server and two mobile learning applications, one for Android and one for iOS. Game-based elements that are used in the classroom tool include time pressure and a leaderboard. While the classroom tool is used in the classroom to measure the performance of the overall class, the mobile applications focus on the knowledge growth of the individual learner.

In order to take advantage of the advances made in technology and the BYOD wave, Yactul lays a special focus on including a wide and extensible range of different question types, called activity types. This diversity provides the advantage that students will not get bored quickly.

The currently supported activity types include, but are not limited to a *Simple Question type*, which provides multiple answer alternatives, where only one is correct. A *Multiple-Choice Question type*, which is similar to the Simple Question type, but multiple responses may be correct. A *Simple Focus Question type*, where only one answer is correct. The difference to the Simple Question is that only one alternative is shown at a time and the student needs to hit a buzzer when the correct answer appears. A *Multiple-Choice Focus Question*, which is similar to the Simple Focus Question, but multiple answers can be correct. A *Build Pairs Question type*, where the learner needs to build pairs between terms of a left and a right column.

Furthermore, as Grévisse et al. continued to explain, having the possibility to reuse previously created questions, called activities in Yactul, in multiple quizzes, is often a shortcoming in existing GSRSs. In Yactul, teachers can create activities and include them in multiple quizzes. This is especially useful for the classroom tool, if the instructor wants to reuse an activity for example in multiple lectures.

The next subsection gives an overview of the system architecture of Yactul, which should highlight the fact that due to its extensible nature, it is relatively easy to include new, custom activity types.

3.2.2 System Architecture

The extensible architecture of Yactul, presented by Grévisse et al. in (Grévisse et al., 2017) can be seen in figure 3.1. It follows an explanation of its different components.

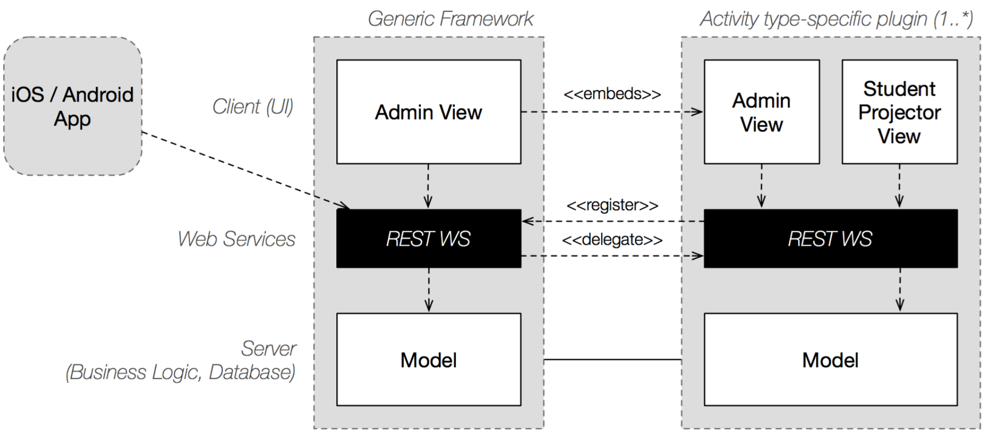


Figure 3.1: Extensible Yactul architecture (Source: (Grévisse et al., 2017))

The generic framework module is used to administer and manage a Yactul installation and constitutes the core module. It includes the management of the different activity types. Each such type is represented by another, separate module. As long as an activity module respects a certain architecture and communication interface, it can be registered at the core module. This demonstrates the extensibility of Yactul with regards to new activity types.

Each time the core module receives an activity-specific request, it delegates it to the concerned activity module. Each such module contains its own specific student, projector and administrator views. The student can see the student view on his own device, including his achieved score and the question and answer possibilities. Moreover, he needs to submit his answers to the questions on his device. The projector view displays the activities and overall results of the class. The administrator view is normally used by the teacher to create the activities and quizzes. The views of Yactul, being based on HTML5, can be used on any type of device that disposes of a modern web browser. The employed communication protocol between the different modules is based on REpresentational State Transfer (REST) web services.

The core module is also responsible for communicating the activities and quizzes data to both Android and iOS applications. In order to do so, the mobile applications use the REST architectural style to send HyperText Transfer Protocol (HTTP) requests to the server. They receive a response in a JavaScript Object Notation (JSON) format, which is a lightweight data interchange format. A small example of a snippet of a received JSON document can be seen in listing 3.1. It shows the property names and values of an activity inside a quiz. After the mobile applications received their response, they can work independently from the Yactul server.

```
{
  "id":23,
  "activityOrder":6,
  "type":"SimpleQuestion",
  "dtype":"SimpleQuestion",
  "difficulty":1,
  "time":40,
  "name":"SimpleQuestion : Programming 2",
  "description":"SimpleQuestion about
                Programming 2",
  "text":"When passing arguments to functions,
         C uses call-by-reference",
  "category":["alma:Pointer",
             "alma:CallByReference",
             "alma:CallByValue"],
  "answers":["true", "false"],
  "solution": "false"
}
```

Listing 3.1: Document snippet of a Yactul activity

3.2.3 Mobile Applications

To promote a continuous learning experience, the mobile applications can be used after the class to play the quizzes from the lecture again anywhere at anytime. In the versions, as described in (Grévisse et al., 2017), there are two modes. The first one is the Sequential mode and is organised in quiz groups and quizzes. It is called sequential because the order of the quizzes is exactly the same as presented in the classroom during lectures. The quiz groups are helpful to arrange related quizzes from a course together. Their current usage is to organise quizzes by their respective courses. In this mode, the student first selects one or multiple courses from the groups and then he chooses the quizzes he wants to play, which

are commonly arranged by the weeks of the semester.
The second one is the Coaching mode. It allows a student to select a list of categories from the available quiz groups. One topic may appear in several groups. Therefore, if the learner decides that he wants to study the content of several groups simultaneously, the resulting categories may combine activities from multiple groups. As a consequence, the Coaching mode concentrates on teaching categories and topics rather than content related to a single course only. It therefore supports a global learning style. An abstract representation of the two modes is shown in figure 3.2

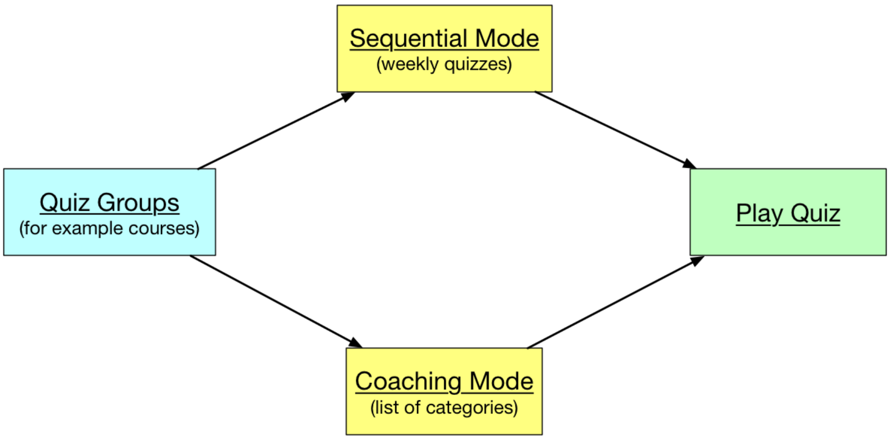


Figure 3.2: Abstract representation of the two modes of the mobile applications

The difference between a sequential and a global learning style that Grévisse et al. (Grévisse et al., 2017) made, corresponds to the differentiation between sequential and global learners as explained in (Felder and Silverman, 1988). Felder and Silverman state that some students learn the course material in the exact same order as presented during lectures. They define them as sequential learners. However, other students are not comfortable with this way of studying, as they need to figure out their own methods for solving problems. They are referred to as global learners. It is therefore important to distinguish between the two modes to fit the needs of every person.
Moreover, the Coaching mode inside the iOS application follows the Leitner spaced repetition system. It is responsible for choosing the activities that are presented to the learner. In this mode, colours are used, as depicted in figure 3.3, to show the student which categories he should revise. A green colour tells him that he answered the activities of the concerned topic overall correctly during the past interrogations. Therefore, he does not necessarily need to repeat it. A red colour on the other hand recommends repeating the concerned category, since its content has been answered poorly overall.
Both, the Sequential and the Coaching mode keep track of the student's progress in the played activities. This allows to present the learner with the colour scheme

for the different categories, as well as to decide which activities to show him in the Coaching mode.

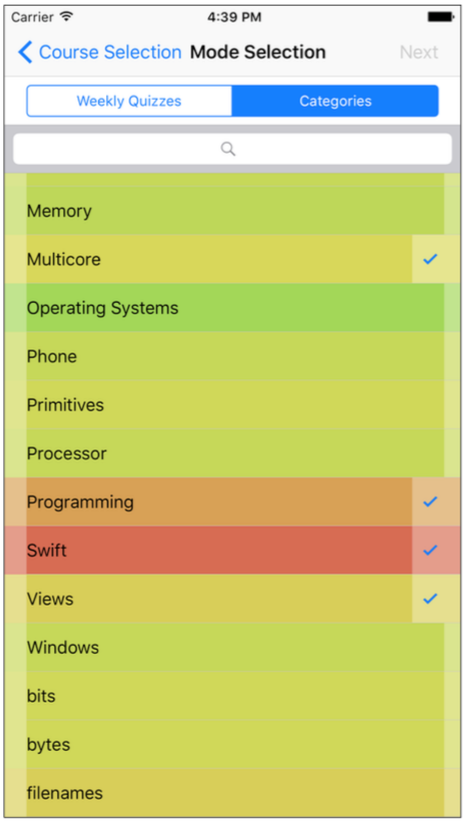


Figure 3.3: Coaching colour feedback in the iOS mobile application (Source: (Grévisse et al., 2017))

3.3 ALMA

Adaptive Literacy-aware learning Material IntegrAtion (ALMA) is an ecosystem which comprises an extensible and lightweight modular ontology, targeted to the domain of computer programming (Grévisse et al., in press). This ontology includes 4 modules. The first two are specific to a programming language, as well as to an Integrated Development Environment (IDE). The first of them is the Java module, which contains concepts that are based on the Eclipse Java Development Tools (JDT) model. The second one is the C module, which contains concepts from the Eclipse C/C++ Development Tooling (CDT) project. The Stack Overflow (Stack Exchange Inc., 2017) module constitutes the third one and includes a set of selected tags from the Stack Overflow website. It is useful for retrieving tag-related articles.
These 3 modules are mapped to the fourth module, namely the ALMA Core module, which contains abstract programming concepts. Moreover, ALMA Core is

language-independent and its concepts include human-readable labels, which are useful for annotating learning material. They are helpful for tagging and adding new custom documents. Furthermore, the abstract ALMA Core concepts are not necessarily bound to a single programming language, since constructs like loops appear in several languages.
An outline of the 4 modules, together with the abstract programming labels, as well as the learning material from the Stack Overflow website is provided in figure 3.4. Finally, the ALMA ontology is extensible to include new modules with respect to new programming languages, IDEs and platforms.

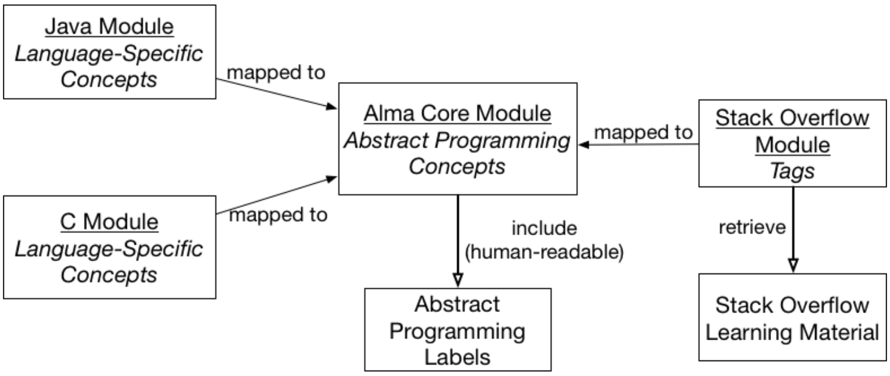


Figure 3.4: Mapping between the 4 modules of the lightweight modular ALMA ontology

Its main purpose is to integrate annotated learning material, which is related to programming concepts, into an IDE. Concepts in a piece of code can be extracted by means of the underlying Abstract Syntax Tree (AST).
Moreover, including the annotated learning material in other systems can also be useful. The next section explains amongst other things how such an integration is realised in the Yactul system.

3.4 Feedback Support

3.4.1 Integrating ALMA into Yactul

A student can use one of the two Android or iOS mobile applications to play a quiz for himself after the lecture. The advantage is that he is not limited to the lecture time slots and that he can therefore take as much time as he needs to understand the quiz activities. As a consequence, the learner might feel the urge to consider related learning material when he does not understand the answer to a question.
Finding the corresponding passage providing the required explanations within the course material, can be very time consuming. Moreover, locating useful resources

of high quality online often is not a trivial task. Therefore, to help the student to find appropriate learning material for an activity in a faster and more convenient way, integrating relevant resources directly in the mobile application may be helpful.

The ALMA ontology provides learning material support for learning aspects related to programming education. These learning resources can be reused and integrated into the Yactul system. In a first step, the abstract programming concepts from the ALMA Core module need to be retrieved. Then in a second step, for each such concept, the related documents are requested. In order to do so, the REST architectural style is used to send a HTTP GET request to the ALMA server. Note that the learning material must therefore contain annotations of the concepts that are defined in the ALMA ontology in order to be retrievable. Annotated learning resources could for example include lecture slides and notes (Grévisse et al., in press).

To give a concrete example, a JSON document snippet of an abstract programming concept from the ALMA Core module is depicted in listing 3.2.

```
{
  "label": "Pointer", "uri": "alma:Pointer"
}
```

Listing 3.2: Document Snippet of an ALMA Core concept

Each concept of such a JSON resource consists of a pair. The Uniform Resource Identifier (URI) uniquely identifies the ALMA Core concept and is provided together with its human-readable label. An excerpt of the learning material that is related to the Alma Core concept *alma:Pointer* is shown in listing 3.3. These learning resources can be reused inside the activities of a quiz in the Yactul mobile application. Each such resource is given as a pair that is composed of a title and a Uniform Resource Locator (URL).

If the teacher wants to create for example a Simple Question activity in Yactul, then he can provide several categories that describe its content. These categories need to be provided as human-readable labels from the ALMA Core module as depicted in figure 3.5. The related ALMA Core concepts of the provided labels can then be retrieved and the documents that are annotated with them can be shown to the learner with their respective title. This is convenient, because the tagging mechanism of the activities already existed in Yactul and was used, for example, by teachers to filter activities per categories, or by students in the Coaching mode of the mobile application to select the list of categories that they wished to be questioned about. However, no relation to the ALMA ontology existed. As a consequence, the suggestion of the documents, as well as the categories, by which

```
[
  {
    "title": "The C Programming Language
              (Second Edition)",
    "url": "https://ds-pub.uni.lu/teaching/
            programming2/Bible.pdf#page=107"
  },
  {
    "title": "Memory Addresses",
    "url": "https://ds-pub.uni.lu/teaching/
            programming2/scrapbook/#17"
  },
  {
    "title": "First Pointer Example",
    "url": "https://ds-pub.uni.lu/teaching/
            programming2/scrapbook/#18"
  }
]
```

Listing 3.3: Excerpt of the documents related to the ALMA Core concept *alma:Pointer*

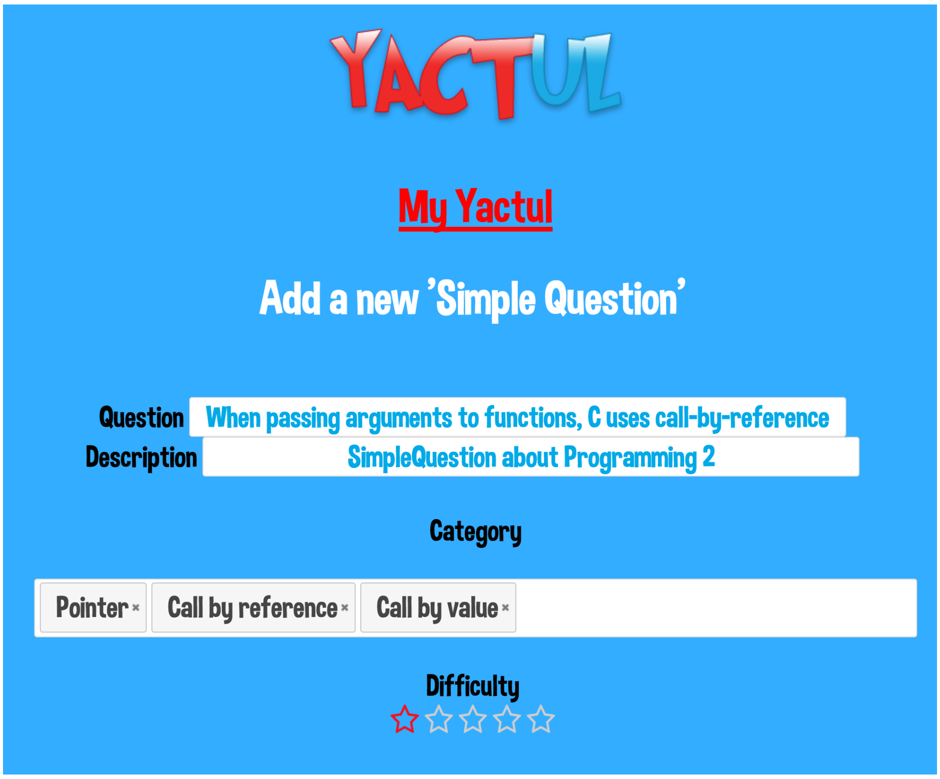


Figure 3.5: Administrator view - creation of a new Simply Question activity

the student can filter the activities inside the mobile application, are all based on the ALMA Core concepts.

In figure 3.5, which is related to the listing 3.1, the teacher indicated the following human-readable labels for his activity *Pointer*, *Call by reference* and *Call by value*. The related ALMA Core concepts are *alma:Pointer*, *alma:CallByReference* and *alma:CallByValue*. They are provided as property values of the property name *category* inside the listing 3.1. These ALMA Core concepts can be retrieved by the mobile applications from the core module of the Yactul server. As the human-readable labels of the ALMA Core concepts are provided by the teacher in the category field on a per activity basis, learning material that is closely related to each question can be provided.

Documents from the Stack Overflow website, the scrapbook web resources of the Programming 1 and Programming 2 lectures from Rothkugel of the Professional Bachelor Course in Computer Science at the University of Luxembourg, as well as some annotated Portable Document Format (PDF) documents from the handouts from Zampunieris for the same Programming 1 lecture, are currently annotated and can thus be retrieved within the mobile application (Université du Luxembourg, 2017c,a,b; Rothkugel, 2017b,a).

To summarise the document retrieval and integration procedure, figure 3.6 provides an overview of the whole process. Note that both, the ALMA ontology, as well as the mobile applications target the domain of programming education.

Finally, the fact that the teacher can provide several categories that describe an activity, is in line with the objective to foster understanding of complex, interrelated learning topics. One activity may require knowledge across different concepts that have been introduced during the lecture, which is different from a task that does not require a lot of reflection and only requires learning by heart. For example, the provided set of categories for the activity depicted in figure 3.5 shows that several concepts need to be understood to master its content. In this case, the student needs to have comprehended the notions of pointer, call by reference and call by value. This is very different from a simple task where only learning by heart is required. Annotated learning material for all these categories can be revised.

3.4.2 Documents and Solutions

A survey performed by Nielsen et al. (Nielsen et al., 2013) shows that the correct usage of a CRS in a classroom is important to help students to improve themselves. They asked students what they considered as the most important aspect of CRSs with regards to their own learning. Most of them replied that the teacher's explanation is most valuable for them. He should therefore explain all the correct and wrong alternatives, even if most of the learners answered the concerned question correctly.

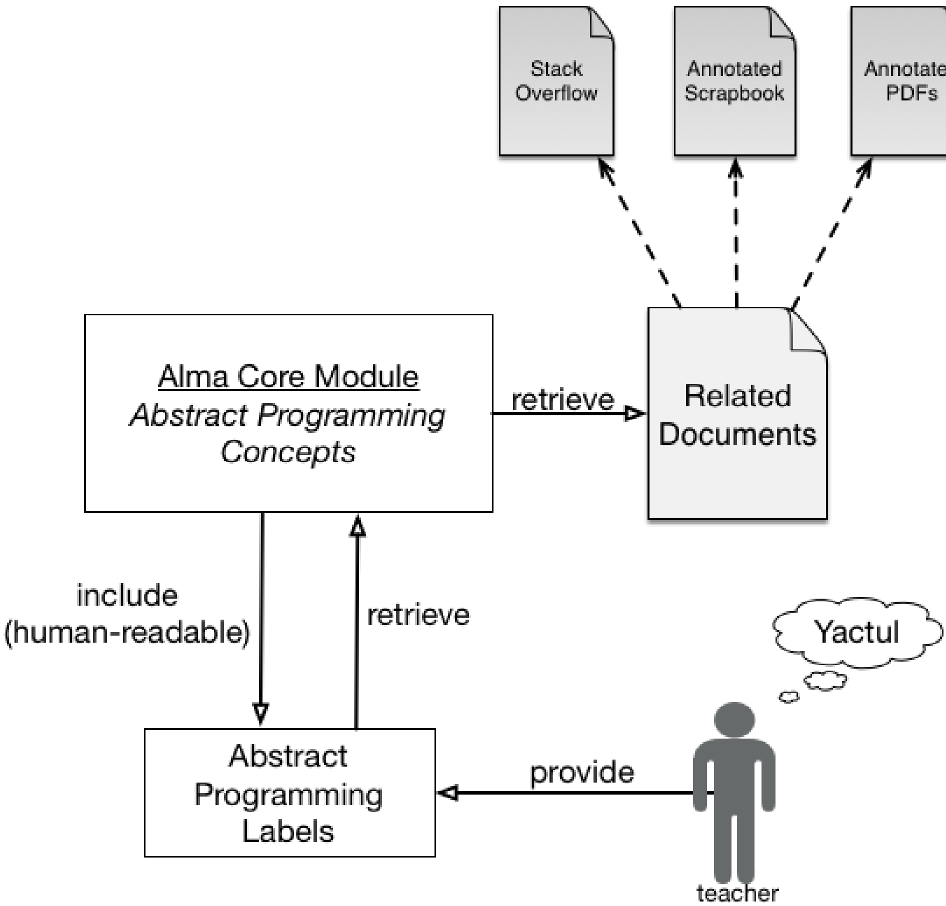


Figure 3.6: Document Retrieval - Integrating ALMA into Yactul

This shows how important it is for students to receive an explanation for the different activities and why the learning material support inside the mobile application is so beneficial for them. To reflect this result indirectly in the personalised response system, it is therefore important to provide the possibility for the student to always consider related learning material of an activity, even if his provided answer was correct. Moreover, to imitate the teacher's feedback that he provides in the class, in most cases either directly after a question was played, or more rarely after the entire quiz, it is therefore important to provide the possibility to the student to revise related learning material directly after he played an activity. In the proposed solution, the learning material that is related to the concepts of an activity can be directly considered after the learner has provided his answer.

Moreover, Nielsen et al. also found out in their survey that students want to have the alternative to say that they do not know the answer to a question. They do not want to blindly guess if they have no idea. The reason is that they fear that if a lot of them guessed the answer correctly, then the teacher would not explain the solution well enough. It is therefore important to provide this alternative to students in the personalised mobile response system as well. Even if there is no risk that the teacher will not explain the solution sufficiently well afterwards, students may fear to guess an answer, because they might think that this could mislead the spaced repetition algorithm of the mobile application that calculates when each activity should be repeated. In the long run, this will not make a big difference for the underlying algorithm, since the learner needs to guess the correct answer several times in a row in order to have a significant impact. However, psychologically, not being able to tell that the student does not know the answer, can make a big difference to him. He might feel more comfortable to be honest with himself than guessing. Therefore, for each activity type, the student has an alternative, which is to say that he does not know the answer.

Furthermore, Felder and Silverman (Felder and Silverman, 1988) made distinctions between different kinds of learners and explained that students learn in different ways. In the mobile application, as some learners may prefer to consider the documents first before they want to see the correct solution of the activity or vice versa, providing them with the freedom to decide in which order they want to consider both, as well as if they even want to consider the related learning resources or the correct solution, is their own choice. This decision may not even solely depend on the student himself, but also on the activity that he is unfamiliar with. He could for example think that he possibly knows the correct answer to a question after considering the related learning material and might want to figure out on his own if he is correct or not the next time when the activity is presented to him. On the other hand, if he does not know what the answer to the activity could be, he might decide that he wants to see the correct solution. Providing the learner with the flexibility to figure out what is best suited for him in each individual case,

fosters the individual learning experience of the mobile application and is therefore supported for each activity type.
Calling up the documents related to an activity is quick and easy, which has to be the case in order to provide the learner with a sufficiently large advantage compared to the process of looking up related learning material on his own. Therefore, a list of the relevant and related learning resources is proposed to him. Moreover, being able to quickly switch back and forth between the learning material and the corresponding activity is possible as well. In order to do so, the documents that are retrieved from the ALMA server, which are all accessible over a URL, as can be seen in listing 3.3, are displayed separately in a web browser. This way, the student can quickly switch back and forth between the considered learning material and the quiz. An advantage offered by current web browsers, is that the learner can usually create several tabs and can therefore quickly switch between different learning resources as well.

3.4.3 User Freedom

User freedom is important in different aspects for enabling a personal learning experience. As explained in the previous section, providing the student with the possibility to decide if and how he wants to consider the correct answers of an activity, as well as the related learning material, is one aspect.
From chapter 2 is known that personalised response systems often use strict time intervals in their spaced repetition algorithm in order to find the optimum intervals to schedule repetitions. In other words, the applications decide for the student when it is time for him to repeat which learning item. The purpose of this scheduling is to minimise the number of repetitions, such that the learner is still very likely to remember the answer. However, in the context of university courses with assignments, projects, exams and deadlines, this is not very realistic. For example if a student has an exam the following day in a course and he wishes to repeat the related learning material by using the personalised response system, he does probably not want to repeat the activities from all his courses that he is most likely to forget soon. The retrieved questions could in this case very easily not be related to the course that he wants to revise for at all. Therefore, to find the optimum repetition intervals can be useful when there is no time pressure, but this is usually not the case for a lot of university students.
The mobile game-based learning application that is presented in this work therefore leaves the student the freedom to revise for the courses he wants to. More precisely, the underlying spaced repetition method provides the learner with the choice to decide, for which courses and categories he wishes to revise and shows him the activities that are the most valuable for him in this learning context. Therefore, the solution is a trade-off between optimum repetition intervals and

the freedom to let the student decide what he wants to learn. Note that the spaced repetition algorithm only decides in one part of the application which activities to present to the learner. This is the Coaching mode and will be presented in a later section of this chapter.

Another aspect of user freedom that the proposed solution takes into account, is that students do not always have the time to repeat activities when it is best for them to do so. In applications like Anki, if the learner does not repeat flashcards for several days or even weeks, he risks to get overwhelmed by the large amount of content that he needs to revise, once he decides to use the application again. This can lead to user frustration very fast, because the learner does potentially not know where he should start with his repetitions. Moreover, some students might not want the application to tell them when they should revise, as they can get annoyed by such an authoritative behavior rather quickly. This is avoided in the proposed solution. The underlying spaced repetition algorithm works with probabilities, which express how likely an activity is to be shown in an upcoming quiz. These probability values do not change over time, which tackles the problem to not overwhelm the learner with activities that he needs to repeat once he did not use the application for several weeks. The subsection 3.4.4 presents the spaced repetition algorithm and shows more concretely how these probability values are calculated.

Finally, the last aspect of user freedom that is taken into account, is that if the student uses the Coaching mode, which is based on the spaced repetition learning algorithm, he has the possibility to play activities as long as he wants. This is possible because of the probability values that are used to decide which activity to present to the learner, instead of the strict repetition intervals from other applications, which schedule the revisions to a fixed date.

3.4.4 Spaced Repetition Algorithm

Different Alternatives

The spaced repetition algorithm that is used inside the Coaching mode of the application is based on the SM-2 algorithm (Wozniak, 1998) from the SuperMemo method. There are several reasons for this choice. First, the other alternatives that could have been chosen instead are the Pimsleur method, the Leitner system or a neural network approach.

Starting with the first one, the Pimsleur method does not make a distinction in repetition intervals if a learning item is correctly remembered or not. Moreover, it is solely based on oral recalls. Being able to present a student with the activities that he overall answered the worst, is one of the most important aspects of the individual learning experience of the spaced repetition algorithm. Therefore, being able to distinguish between correct and wrong replies must be taken into account in any case. While relying on oral recalls could be beneficial for some of the activity types and some of the learning tasks, such as for example studying vocabulary, for other tasks, it is not a viable alternative. Taking the example of the Build Pairs Question type, it is hardly possible to construct pairs from a left and a right column orally. Additionally, in order to not bore a student too quickly, the diversity of the different activity types constitutes an important aspect of the Yactul system. Relying purely on audio recalls would make the majority of these types, like the Build Pairs one, nearly impossible.

Secondly, the Leitner system, being based on boxes of flashcards, provides only the possibility to distinguish between correct and wrong answers. A wrong reply results in the activity being moved backwards to a box, where it is repeated more frequently. In case of a correct answer, it is moved forward to the next box, where it is repeated less often. However, other factors that are important in the mobile application, such as the time that it took to reply to the activity, or how the student answered to other, similar activities that include at least partially the same categories, are not taken into account. This was not the purpose of the Leitner system, since the first version was originally designed for paper flashcards. If the system needs to take such factors into account, then this would make the model more complex. This is not recommended for paper flashcards, since the learner would need to do all the resulting calculations by hand and he would probably spend more time doing so than revising his learning material. However, in the context of the personalised mobile response system, taking additional parameters into consideration is possible, because the calculations do not need to be performed by hand anymore. Therefore, the Leitner system is not well suited for the application neither.

Finally, a system that is based on neural networks generally requires a very large amount of training data to perform well. Hence, collecting enough data in a university context is impossible, since it will require many participants in order to receive enough data and to train the network sufficiently well.

Therefore, the only possible choice that is still left for a spaced repetition algorithm is based on the SuperMemo method. The biggest problem with its underlying spaced repetition method however is that it is mainly targeted to scheduling repetitions for digital flashcards with simple tasks. Like pointed out by Schimanke et al. in (Schimanke et al., 2013), applying such an algorithm to a learning environment that focuses on the understanding of categories, rather than simple atomic tasks, needs a certain degree of adjustment. This is exactly the case for the proposed personalised mobile application. It focuses on the teaching of activities, where several ones may treat the same or similar categories. Moreover, activities from different courses can also overlap in their content. Therefore, the proposed solution must take this categorisation into account in its spaced repetition algorithm. Overall, the SuperMemo method appeared to be the most flexible to integrate such a categorisation and to take additional factors into account as

well, such as the time needed to reply to an activity. It seems therefore to be the best alternative. Moreover, there are several formulations and enhancements of the spaced repetition algorithm that is used in the SuperMemo method, which are designed for computer implementations.
The decision to base the custom algorithm on the SM-2 formulation of the SuperMemo method has several reasons. Firstly, it is widely used in existing popular personalised response systems that are based on flashcards, such as Anki and Mnemosyne. Secondly, newer formulations of the SuperMemo method involve a lot of additional complexity and the SM-2 algorithm proved its usefulness and efficiency in existing applications. Moreover, the personalised mobile response system, as mentioned in the previous section, uses probabilities that determine how likely it is for an activity to be presented to the learner, in contrast to optimum repetition intervals. It constitutes thus a compromise between optimum intervals and user freedom anyway. A newer and much more complex formulation of the SuperMemo method, which performs slightly better than the SM-2 algorithm, will therefore not have a significantly large enough impact on the underlying application. Moreover, as explained in subsection 2.1.3, the SuperMemo method is determined heuristically and is therefore not guaranteed to be optimal in any case.

Custom Spaced Repetition Algorithm

The algorithm inside the Coaching mode uses spaced repetitions for the selection of the activities to present to the student. As the natural logarithm function $ln(x)$ is the inverse function of the exponential function e^x and the forgetting curve decays exponentially over time, $ln(x)$ describes the "importance" of the student to repeat a learning item. For example, immediately after the learner revised an activity, he is most likely to still remember it. Therefore his importance to repeat this item should be very small. Figure 3.7 shows in green: $f(x) = ln(x)$ and in orange $g(x) = e^{-x}$ and illustrates this reasoning. Note that the importance of repeating an activity augments very fast at the beginning and stabilises over time, which is in line with the nature of the forgetting effect.

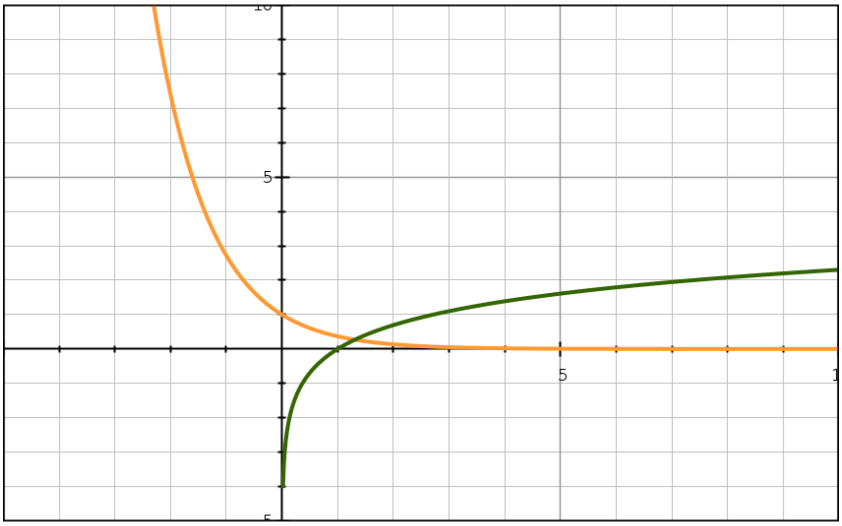


Figure 3.7: Two functions, in green: f(x) = ln(x), in orange g(x) = e^{-x}

The following formula is used for the calculation of which activity to repeat next in the Coaching mode:

$$f(x) = -ln\left(\frac{x}{101}\right) \quad \left(\text{for priority x } (1 \leq x \leq 100)\right)$$

X describes the priority of the repetition of an activity, where 1 is the highest and 100 the lowest priority. Note that $-ln\left(\frac{x}{101}\right)$ is chosen instead of $ln\left(\frac{x}{101}\right)$, because $0 < \frac{x}{101} < 1$ for $1 < x < 100$. Therefore $ln\left(\frac{x}{101}\right)$ would be negative for these values. Moreover, as 1 is the highest priority, it should have the largest value, since the results of the formula will be used later on to define the probabilities that the activities will be repeated. Priority 1 expresses that an activity should be repeated very soon, which is the case for $-ln\left(\frac{x}{101}\right)$. But before presenting how the probabilities are calculated, the priorities themselves are defined as follows:

$$P(0) = 1$$

$$P(1) = 1$$

$$P(2) = 6$$

$$\text{for n} > 2\text{: } P(n) = P(n-1) \times EF$$

P(n) is the priority of the activity after the n-th correct repetition in a row and *n* is initialised with 0. For each correct reply of a student to an activity, n is increased by 1. If he gives a wrong answer, then n is set to 0. Moreover, if P(n) > 100, then P(n) = 100. If it is a fraction, then it is rounded up to the nearest integer value. A small example of 8 activities is given in table 3.1 to illustrate how the related probabilities are calculated based on their priority values and the defined formula $f(x) = -ln\left(\frac{x}{101}\right)$.

activity	x	f(x)	probability (in %)
1	1	4.615	29.8
2	6	2.823	18.2
3	6	2.823	18.2
4	15	1.907	12.3
5	19	1.671	10.8
6	30	1.214	7.8
7	65	0.441	2.8
8	100	0.001	0.065

Table 3.1: Mapping of priorities to probabilities, where x = priority, f(x) = $-ln\left(\frac{x}{101}\right)$, $\sum f(x) = 15.495$

Note that *P(0)* and *P(1)* both have the highest priority value of 1. This is due to the mobile application integrating overlearning. From the observation that overlearning can greatly benefit only those learners that need to retain the information for several days in memory, as discussed in subsection 2.1.1, the assumption is made that if the student repeats an activity several times in a row, then it is more likely that he needs to remember it for a longer time period. As a consequence, he will thus not greatly benefit from overlearning, which is therefore not included if the user repeats an item more than twice correctly in a row. Furthermore, as students benefit a lot from overlearning for short retention intervals, in the order of a couple of days, in the beginning, it can help them to correctly recall the information if they use the application again in several days. Note however that because of the freedom of the learner to decide in the Coaching mode how long he wishes to repeat, as well as because of the fact that the algorithm works with probabilities, it is not guaranteed that overlearning will always take place.
Finally, the second variable from the priority calculations is *EF*, which is the easiness factor of the student to retain the corresponding activity in memory. The EF value is the place where the categories are taken into account. It is calculated as:

$$EF = \frac{2}{3} \times EFActivity + \frac{1}{3} \times \frac{\sum EFCategories}{|EFCategories|}$$

Therefore, the easiness factor of the activity itself counts for $\frac{2}{3}$ and the one of the sum of all the categories for $\frac{1}{3}$. Each category that the teacher indicated for the corresponding activity is taken into consideration in $\sum EFCategories$ and is defined as:

$$EFCategory = \frac{\sum EFActivities}{|EFActivities|}$$

The activities that are included in $\sum EFActivities$ are all those, where the teacher provided the concerned category when he created them. For each such activity, the value of *EFActivity* is initialised with 2.5 and must always be greater or equal to 1.3 and smaller or equal to 3.0. As a consequence, the value of *EFCategory* must always be greater or equal to 1.3 and smaller or equal to 3.0 as well, since it is calculated as the average of its related *EFActivity* values. Finally, the overall calculated *EF* sum needs to be in the same domain as well.
In case of a correct reply, the value of *EFActivity* is updated with the following formula, where *EFActivity'* is the new value of the easiness factor and *EFActivity* the old one:

$$EFActivity' = EFActivity + (0.1 - (5-q) \times (0.08 + (5-q) \times 0.02)) \quad \text{(correct answer)}$$

Note that this is the exact same formula as seen in the SM-2 algorithm (Wozniak, 1998), where it is used to calculate the value of the entire easiness factor. However, in the proposed customised SM-2 algorithm, this formula is only used to calculate the *EFActivity* component of the corresponding activity. The value of the quality *q* depends on the time that the student needed to reply to the activity. It is measured in a grade scale from 3 to 5, where 5 expresses that the response was perfect and 3 that the learner had serious difficulty recalling it. More precisely, it is defined as depicted in table 3.2. The *total time* is the overall time available for the learner to answer the activity. *Time left* is the time that is left after he provided his choice.

q	**quality**	**time range**
5	perfect response	time left $\geq \frac{2}{3}$ total time
4	good response	$\frac{2}{3}$ total time $>$ time left $\geq \frac{1}{3}$ total time
3	serious difficulty	$\frac{1}{3}$ total time $>$ time left

Table 3.2: Correct reply - quality measure scale from 3 to 5

In case of a wrong reply, *EFActivity'* is calculated as:

$$EFActivity' = EFActivity - 0.20 \quad \text{(wrong answer)}$$

This reduction punishes the *EFActivity* value of wrongly replied activities and as a consequence also the *EFCategory* value of its related categories. This is important to determine in which categories the learner struggles the most and to foster the

understanding of complex, interrelated learning topics.
The *EFActivity* value is decreased by 0.20 in case of a wrong reply, because a perfect response augments it by 0.10.

$$(0.1 - (5 - 5) \times (0.08 + (5 - 5) \times 0.02)) = 0.1 \quad \text{(perfect response, q = 5)}$$

Therefore, one wrong reply to an activity is equivalent to two subsequent perfect answers. As a wrong reply also counts if the learner does not provide an answer at all in the given time limit, or if he admits that he does not know the solution. A small example of how the priority of an activity is calculated, is given below.

activity	number repetitions	priority	EFActivity	categories
1	3	15	2.4	alma:CallByValue alma:Pointer
2	3	16	2.6	alma:CallByValue

Table 3.3: Algorithm example - priority calculation

Assume that in table 3.3, the next activity that will be scheduled is activity 1 and that the provided answer of the learner will be correct with a quality *q* of 5. Then for the calculation of the next priority for the activity with IDentifier (ID) 1:

$$EFActivity' = 2.4 + (0.1 - (5 - 5) \times (0.08 + (5 - 5) \times 0.02)) = 2.5$$

$$\sum EFCategories = \frac{2.5}{1} + \frac{2.5 + 2.6}{2} = 5.05$$

$$EF = \frac{2}{3} \times EFActivity + \frac{1}{3} \times \frac{\sum EFCategories}{|EFCategories|} = \frac{2}{3} \times 2.5 + \frac{1}{3} \times \frac{5.05}{2} \approx 2.508$$

$$P(4) = P(3) \times EF = 15 \times 2.508 = 37.62$$

As we round the priority up to the nearest integer value if it is a fraction, finally *P(4) = 38*.

As a conclusion it can be said that the algorithm integrates time intervals, as well as an understanding of categories. Moreover, the risk of the student to get overwhelmed with content that he needs to revise is eliminated, since the underlying probability values do not change if he does not repeat at all for several days. However, the algorithm still provides the best results if the student repeats regularly, since it is based on the spaced repetition learning technique. Furthermore, in the Coaching mode, the learner has the possibility to repeat activities as long as he wants. This is due to the used probability values, which do not perform a strict scheduling of repetitions. The presented algorithm does not dictate the student

that there are no learning items that are due yet. A factor that has not been taken into account, is the difficulty degree, provided by the teacher who created the activity. This is done on purpose, because the algorithm adapts automatically to how well the student performs. Moreover, the provided difficulty degree is subjective and is probably not accurate for every learner.
Nonetheless, as the SM-2 algorithm itself was determined heuristically, the introduced custom algorithm that is based on it still needs to be tested on students to see if it really helps them to improve their individual learning experience. In general, there is close to no formal understanding on the design of systems that are based on spaced repetition (Reddy et al., 2016). Thus testing the custom spaced repetition algorithm in practice can probably help to improve it.

3.5 Game Modes

This section presents the two game modes that are available in the mobile application, namely the *Sequential mode* and the *Coaching mode*. A special focus is given on which parts have been changed and what has been added in comparison to the mobile applications as they were introduced in (Grévisse et al., 2017). Moreover, as Grévisse et al. explained, in contrast to the classroom tool of Yactul, which uses competition to foster gamification, the mobile applications need other instruments to support it, such as the applied time pressure and the variety of activity types. This section also focuses on new game elements that have been introduced in the application to improve the individual user experience and to create a more engaging environment.

3.5.1 Sequential Mode

In the Sequential mode, the learner needs to first choose one or several courses. Then, based on his selection, he has the possibility to choose between two different sub-modes. In the first one, he can play quizzes on a weekly basis, which are arranged in the same order as they were presented during the lectures. This mode already existed in the mobile application before and it was defined as the complete Sequential mode. However, in the new version, a second sub-mode was added, in which the learner can choose a number of categories which he wants to focus on from the different courses. This mode was defined before as being the Coaching mode in (Grévisse et al., 2017). It is now integrated in the Sequential mode, because it allows the learner to play every single activity that appears in the chosen categories exactly once. The spaced repetition algorithm does not decide in this mode which activities are presented to the student. However, it remembers his performance for the Coaching mode, as well as for the colour feedback. The

Coaching mode is presented hereafter.
Another additional change that is made to promote the game-based learning experience, is that the colour scheme is integrated in the courses, weekly quizzes and categories selections to provide feedback at different levels, as opposed to solely being integrated in the category selection. This has the advantage that the learner can already see at the scope of the courses if he understood the content more or less well or rather poorly. Compared to figure 3.3, the Android application integrates the colour scheme in a less striking way, namely it shows the colours in rectangles on the right-hand side of the screen. An example is provided in chapter 4, which highlights important aspects of the implementation of the application.
The colour mapping itself is based on the priorities from the algorithm in subsection 3.4.4. As the priorities of the activities have values ranging from 1 to 100, where 1 is the highest and 100 the lowest priority, 1 is mapped to the reddest and 100 to the greenest colour. Priorities in-between are mapped to their respective colour between the highest and lowest priority values. Therefore, there are 100 different colour values for the activities, ranging from red to green. The colour of a category is based on the average of the priority values of its included activities. The same goes for the courses. However, because of the formula for calculating the priorities $P(n) = P(n-1) \times EF$, the mapping of the different colours cannot be linear, since the priority values augment slowly at the beginning and very fast at the end. This is due to the multiplication of the old priority with the EF value, where the easiness factor is always greater or equal to 1.3. In a linear colour mapping, the colours would stay too long in the red-orange zone and would then almost immediately switch to green. The main goal of the colour scheme is to provide the learner with an overview of his progress and to strengthen his confidence. Therefore, not staying too long in the red zone is important, because this could demotivate him. However, at the same time, it is important to not move too quickly towards a green colour, but to provide him with a steady progressing learning experience.
As categories can be included in multiple courses, the new Sequential mode supports both sequential, as well as global learners (Felder and Silverman, 1988). The advantage of this approach is that it always provides the student with an opportunity to repeat every activity for one or several categories exactly once, or to repeat the quizzes as seen during the lecture. This would not be possible in the Coaching mode, since there, the algorithm from section 3.4.4 schedules the repetitions.

3.5.2 Coaching Mode

To promote the gamification aspect of the mobile application, in the Coaching mode, the courses and their related categories are represented in form of a graph.

An abstract representation of it can be seen in figure 3.8. It shows how categories are related to courses. The used colour scheme is the same as in the Sequential mode. Categories can be included in several courses. Therefore, as can be seen in "category 4" of the graph, it is linked to "course 1" and "course 2". The links that are defined between a category and a course reflect how well the student understood the category inside the course. For "category 4", the student understood the content related to "course 2" very well and the one related to "course 1" more poorly. The colours of the category nodes themselves represent the overall performance of the learner in that category. The graph promotes the fantasy aspect of the mobile application, as well as the challenge of the student to improve himself with the goal to have a graph, in which most of the nodes are coloured in green (Malone, 1980). Introducing a new challenge for an individual learner is important, since he cannot compete against his colleagues anymore, as it is the case in the classroom tool of Yactul.

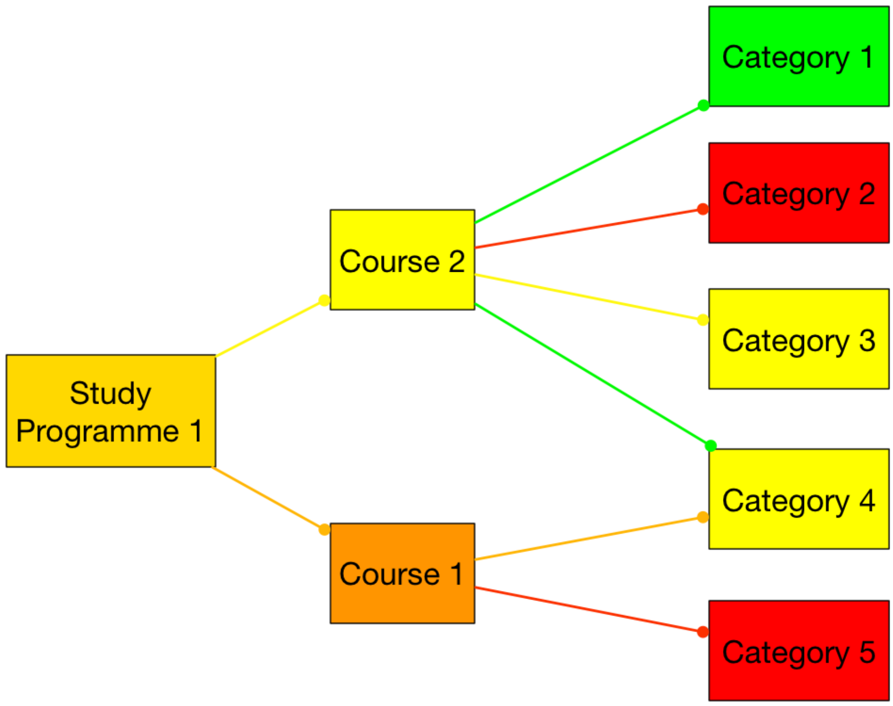


Figure 3.8: Tree Structure - abstract Coaching graph representation

The usage of the graph has several advantages. First, it allows the learner to see at a glance which categories have been studied in which courses over the whole semester. It therefore provides him with a good overview. Second, the colour usage shows him directly which parts he still needs to revise. In figure 3.8 for example, it is strongly recommended that the student repeats the activities that are related to "category 2" and "category 5". Additionally, if he does not want to revise the activities related to a course or category and instead wants to study the corresponding learning material, he can also touch long on a course or

category and he will be presented with the related documents. This can be helpful if he for example wants to revise the learning material before he plays a quiz.
The student can select several categories and courses in the graph that he wants to include in his upcoming quiz. The related activities that will be selected are the following ones.

- If the student selects a course and no category that is related to it, all the activities from the entire course will be included in the activity pool of the quiz.
- If he only selects a category and no course related to it, all activities that are tagged with the respective category will be included in the activity pool.
- If he selects a course and at least one category that is related to it, then an "and" operation will be performed between the course and the category, meaning that only the activities from the selected course tagged with the chosen category will be included in the pool.

Furthermore, as it is shown in figure 3.8, the logic behind the tree structure of the courses together with their related categories can be extended further to include another abstraction layer. Study programmes can for example be added, because there may be students that are enrolled in two programmes at the same time at a university. This is also reflected in the graph and is the reason for the choice of a tree structure over for example star topologies, where a course is in the center and all its related categories are connected to this central point. In a tree structure, it is easier to add another abstraction layer, whereas with star topologies it is hard to add for example the study programme as another abstraction layer, since it is only composed of a central object and additional items that are connected to it.
It is also possible to argue that the activities that are inside a category could be added to the graph and linked to all of their respective categories. This is avoided on purpose, since the mobile application teaches the notions of concepts and more complex topics, rather than single activities.
As the graph fosters the gamification aspect, it should motivate students to use the application, because it is graphically more appealing than for example a traditional list of courses and categories. It is possible to argue that the Sequential mode could also be represented in form of a graph. However, as this mode offers the additional possibility to the student to repeat quizzes on a weekly basis, it would make the graph overcomplicated and no longer readable anymore. A possible alternative could be to offer two separate graphs for the Sequential mode, one for the weekly quizzes and one for the categories. This is not currently realised, but could be done in the future.
In contrast to the Sequential mode, in the Coaching mode, the spaced repetition

algorithm from subsection 3.4.4 takes over the task to decide which activities to present to the student. Therefore, the questions that the learner answered overall poorly are most likely to appear more frequently in a single study session, whereas the ones that he answered well have a high chance to not appear at all. Moreover, the Coaching mode concentrates on teaching concepts and categories that the student did not understand well, in contrast to weekly quizzes, because firstly, the spaced repetition algorithm wants to help the learner to improve himself in the study areas that he did not know well. Secondly, if the student wishes to study for each week, it is most likely that he wants to repeat all the activities that were done during that time period. Letting the spaced repetition algorithm decide which activities to present to him, would potentially not show him all these questions. Thirdly, the Coaching mode targets global learners (Felder and Silverman, 1988). Finally, as already pointed out earlier, in the Coaching mode, the student has the opportunity to repeat activities for as long as he wants to.

3.6 Challenges And Limitations

This section presents general challenges and limitations that are encountered in the context of a personalised mobile learning application.

First, as there are a variety of mobile devices with different screen sizes, the application needs to be more or less scalable to every possible size. This is especially a problem for the graph, because there is a risk that it could very quickly gain in size, mainly in its height, due to the potentially large number of categories. Therefore, the portrait orientation seems to be the better choice over the landscape one. However, for small devices this still constitutes a barrier and it is hard to find a good solution. Zooming in and out of the graph is one possibility to provide the student with a rough overview. The presented personalised mobile application allows this, as well as moving the graph around.

Moreover, the edges that link courses to categories can quickly become very numerous. As a consequence, the graph can become overfilled. To cope with this problem, an edge is only shown if the learner selects its related course or category. A concrete example of this scenario is provided in chapter 4. However, the graph is still at risk of becoming unmanageable very quickly. A possible solution for this is to only show one part of it. For example to depict the courses only if the student zooms out. The categories can then be shown only when he zooms in. This is not currently supported by the mobile application but it could be an idea for an improvement. Another improvement could be to expand the graph on the fly. For example at the beginning, the student would only be presented with the courses. As soon as he selects one, he would see its related categories. The advantage of this approach could be that the graph would be more clearly arranged. However,

the disadvantage would be that if the learner wanted to see a global overview of his overall performance in all the courses and categories, he would first need to expand the entire graph. Therefore, a possible solution could be to give the learner the choice to decide between two alternatives. One where the nodes would be expanded on the fly, and one where the entire graph would be directly visible.
Older devices often have a low resolution and therefore problems to show the colours of the graph properly. Furthermore, they often only dispose of a very limited amount of computing capacity. This can result in performance issues, especially for larger graphs.
Another problem is that it is difficult to find a framework for 2D game development that is not too complicated for a simple graph inside a prototypical Android implementation. Apple offers for iOS the SpriteKit framework (Apple Inc., 2017), which can be used to create simple 2D games. However, Android only offers more complex alternatives, such as the Open Graphics Library (OpenGL) ES Application Programing Interface (API) (Google Inc., 2017a), libGDX (Mario Zechner, 2013) and AppGameKit (The Game Creators Ltd., 2017).
Finally, the mobile application needs to store a lot of user data to manage all the quizzes, activities and information needed for the spaced repetition algorithm. Using a SQLite database is a popular approach in Android. In order to do so, it is either possible to write Structured Query Language (SQL) queries or to use an Object-Relational Mapping (ORM) technique. However, popular ORM solutions, such as OrmLite (Gray Watson, 2017) and GreenDao (greenrobot, 2016) provide only limited and lightweight functionality to persist Java objects.

4 | Implementation

Contents

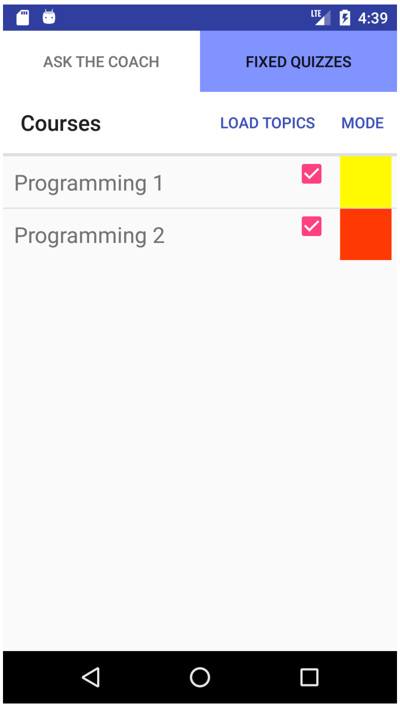
This chapter provides in its first section an overview of the most important aspects of the Android implementation with a special focus on the realisation of the personalised feedback. The second part presents a discussion on some potential improvements and trade-offs of the prototypical Android implementation.

4.1 Android Development

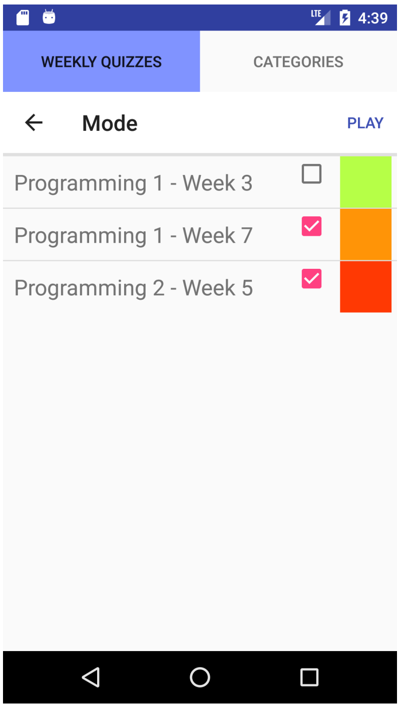
The application takes advantage of the fact that a lot of students possess an Android device. This section provides an overview of the most important aspects of the developed personalised game-based learning application.

4.1.1 Personalised Feedback

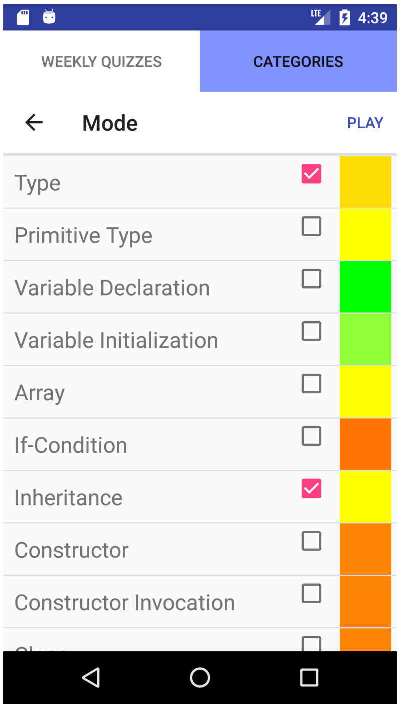
The colour scheme used in the Sequential mode can be seen in figure 4.1. Figure 4.1a presents the colour feedback that is provided for each course. Figure 4.1b shows the one for the weekly quizzes and 4.1c the one for the categories of the two selected courses. The reasons to show the colour feedback only in rectangles on the right-hand side of the screen are firstly to not discourage the student if he answered nearly all the activities poorly. If the colours stretch over the whole screen, then this could result in student frustration rather quickly, because nearly the entire display would be shown in red. Secondly, the categories, weekly quizzes and courses stay more readable with this unobtrusive approach.
The graph of the Coaching mode uses the same colours as the Sequential mode. It can be seen in figure 4.2. The categories and courses that are framed by a blue rectangle are the ones that the learner selected to revise. In this case, he wants to repeat the activities that are related to the “array” or “type” concepts and occur in the “programming 1” course. It is noticeable that only the edges that are related to the selected categories and courses are shown. However, the screen size of the



(a) Courses



(b) Weekly quizzes



(c) Categories

Figure 4.1: Sequential mode

relatively small smartphone constitutes a limitation, since not every course and concept is visible at once. Furthermore, the graph could become even much larger for more courses and categories. Section 3.6 provides possible solutions to this problem.

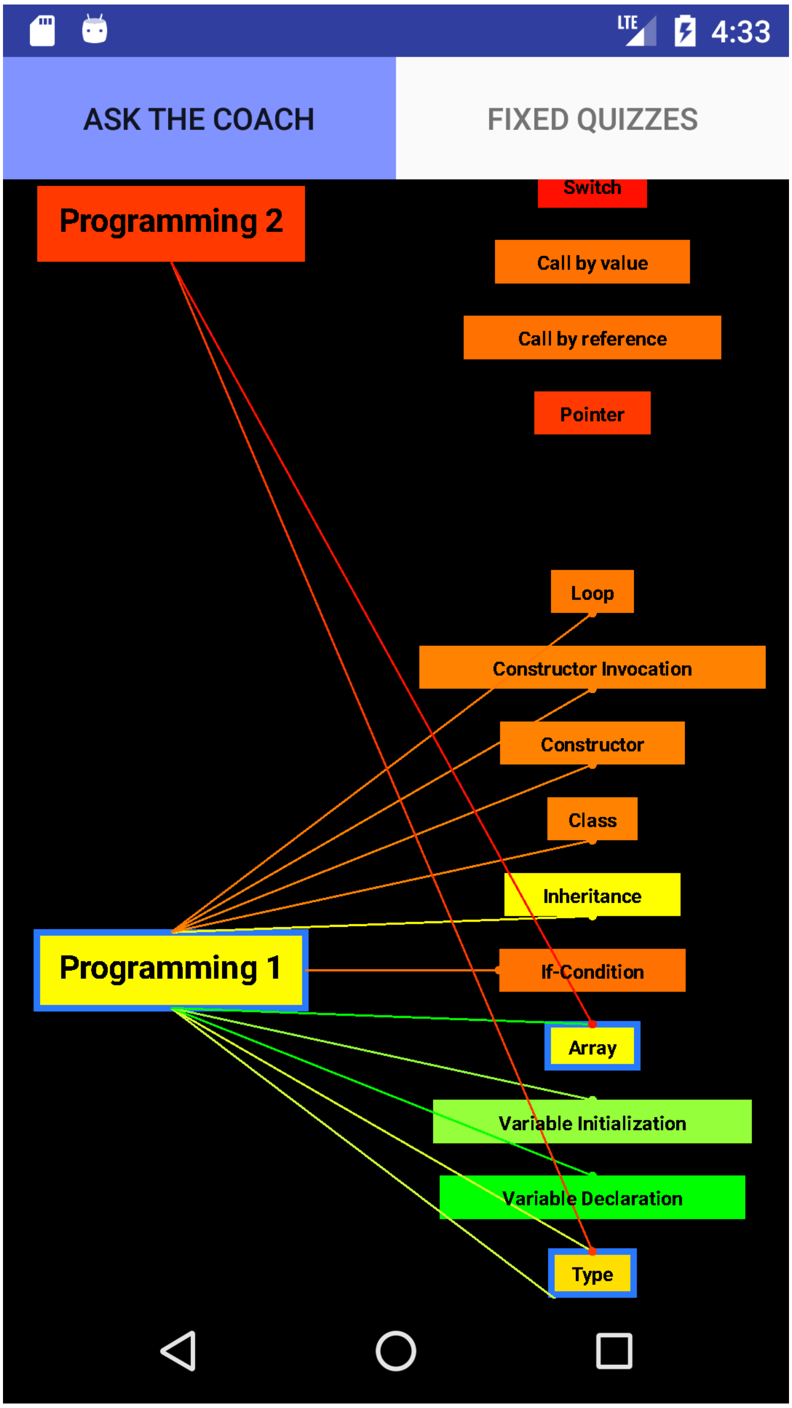
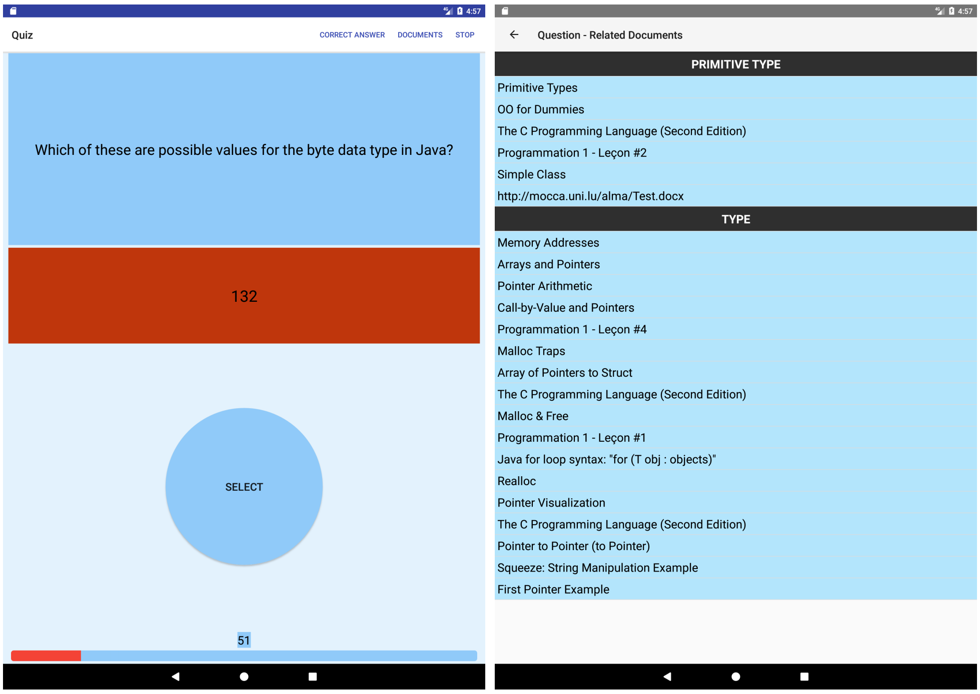


Figure 4.2: Coaching mode - graph

Figure 4.3 shows in 4.3a an example of a Multiple-Choice Focus activity. The learner provided the wrong answer and he can now decide if he wants to see the correct one or to consult the related learning material. 4.3b displays the list of documents that are related to the corresponding activity. "Primitive type" and "type" are the two categories that the teacher provided for the question. Its suggested documents are retrieved from the ALMA server. Furthermore, after each played activity, the student can decide if he wants to stop the quiz or not. This counts for both, the Sequential, as well as the Coaching mode and can be seen in the top right corner of figure 4.3a.



(a) Multiple-Choice Focus activity (b) Document suggestion list

Figure 4.3: Activity feedback

If the learner touches "load topics" on the courses screen of the Sequential mode as depicted in figure 4.1a, then in a first step, the REST architectural style is used to send a HTTP GET request to the Yactul server. The application receives a JSON document as response, which includes the quiz groups, quizzes and activities. The obtained data is then saved locally in a SQLite database. However, the prototypical application currently only works on a static JSON document, because the usage of the abstract programming concepts from the ALMA Core module is not yet implemented on the Yactul server. In a second step, all the ALMA Core concepts are retrieved from the ALMA server. Then for each such concept, the application sends a HTTP GET request to receive the related learning material. The resulting documents for the "primitive type" and "type" categories are depicted in figure 4.3b. The communication between the mobile application and the ALMA server is based on REST web services. Both, the concepts and the learning resources are retrieved in form of a JSON document. Subsequently,

the received data is saved in a SQLite database. Once the application obtains all the necessary information, it does not require a connection to the ALMA, nor to the Yactul server anymore. The only time when a student needs an Internet connection afterwards, is if he wishes to consider the related learning material of an activity. This supports the idea of being able to use the application to play quizzes anywhere at anytime.
A small demonstration of the application is presented in video 4.4 (duration 2:15 minutes). However, it requires a PDF viewer that enables embedded videos.

4.1.2 Major Frameworks

This part presents the two most important frameworks that are used in the Android application.

- Jackson (FasterXML, LLC., 2017) is used for parsing and traversing the JSON document files.
- OrmLite (Gray Watson, 2017) is the lightweight ORM technique that is employed for persisting annotated Java classes to the SQLite database.

4.1.3 Database Model

Figure 4.5 shows the SQLite database of the Coaching data. The ALMA categories (table: *AlmaCategory*) are saved together with their related documents (table: *AlmaDocument*). They are related to their respective activities (table: *CoachingActivity*). Each activity contains its current *priority* and *EFActivity* values, which are needed in the custom spaced repetition algorithm as explained in section 3.4.4. However, the *EFCategory* values do not need to be saved, since they can be calculated based on the EFActivity values. Each activity is contained in one *CoachingGroup*, which is used to represent the different courses.
Currently, the application cannot show the learner his performance of his past quizzes. However, the necessary support is already provided in the database and is enabled by the two tables *CoachingPastQuiz* and *CoachingPastQuiz_-CoachingActivity*.
Being able to consider the past performances of the quizzes can be especially useful if the student wants to repeat the related learning content that he did not understand well at a given point in time. Therefore, providing him with an overview, consisting of the number of activities that he answered wrongly, correctly, or not at all, as well as the average priority of the played activities at that moment in time, can give him a good hint which learning material he should revise. A more detailed feedback of how he performed in each activity is also possible to display

Figure 4.4: Small video (2:15 minutes) - prototypical mobile application (requires a PDF viewer that enables embedded videos)

with the provided data in the database, as it stores the priority in *priorityPastActivity* of each activity for a quiz that was played in the past. Moreover, this allows the learner to see his progress, compared to how he performed at a given point in time in the past and enhances the individual learning experience.

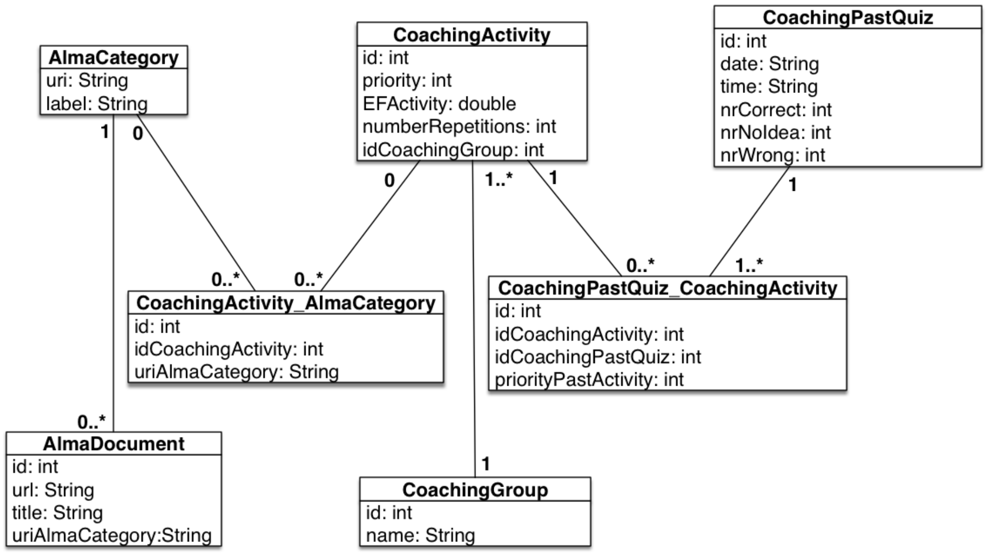


Figure 4.5: UML class diagram - Coaching database

4.2 Discussion and Improvements

In the current implementation, the only time the application retrieves the data from the Yactul and ALMA servers is when the student touches "load topics" in the courses screen of the Sequential mode, as can be seen in figure 4.1a. The advantage of this approach is that the learner can play quizzes without requiring an Internet connection, as long as he does not wish to consider the related learning material from the web resources. This approach benefits from m-learning, since it allows the student to learn anywhere at anytime. This is especially useful when no Internet connection is available, or when travelling abroad. Moreover, the Android and iOS applications were originally designed to be used as offline learning applications, as described in (Grévisse et al., 2017). However, as the learning material that is related to the ALMA Core concepts from the ALMA server is accessed over a URL, the mobile applications cannot completely work offline anymore in any case. Moreover, additional documents that the learner will not be able to retrieve, if he decides to load the topics only once during the first usage of the application, may be added to the ALMA server. Therefore, he might not be able to see the documents that are most valuable to him. To get the related learning material dynamically from the ALMA server when the student wants to consider either the learning resources of a course, concept or activity, will on the other hand always include the most recent documents. Furthermore, the time delays to receive a reply from the ALMA server with the related material are not significantly high. Therefore, a dynamic approach would have a lot of advantages. However, each time the learner wants to consult the overview of the documents only, either by performing a long touch on a category or a course in the Coaching graph, or by retrieving the documents of an activity, a request would already be needed to be sent to the ALMA server. This could potentially result in many requests, which is not advisable, especially when travelling abroad, or when no Internet connection is available.

In the current approach, the student does only need an Internet connection if he wants to consult a specific learning resource, but not as soon as he wants to see the overview of the available documents. This method results in less requests that need to be sent to the ALMA server, with the trade-off that the learner does potentially not receive the most up-to-date documents.

As both approaches have their advantages and disadvantages, a solution that would fit the needs of every possible situation would be to retrieve the learning material dynamically from the ALMA server if the learner is connected to a Wireless Local Area Network (WLAN). If the device is either connected to the mobile network or does not have an Internet connection at all, the data would be retrieved from the persistent database from figure 4.5.

The OrmLite framework, which is the used ORM technique for mapping Java classes to the SQLite database, provides only limited functionality due to its lightweight nature. This results in several nested loops when querying the database in the implementation. As a consequence, for a a potentially large number of courses and categories, this can lead to performance issues, especially for mobile devices with a limited amount of computing capacity and battery life.

Finally, as it is difficult to find a framework for 2D game development for Android that is not too complex for the simple prototypical graph in the Coaching mode, a custom extension of the *SurfaceView* class (Google Inc., 2017b) is instead used to create it. The advantage of this approach is that it is easier to use than a complex 2D game engine. However, the disadvantage is that a custom approach is more difficult to maintain in the long run and offers only limited functionality compared to advanced game engines.

5 | Conclusion

5.1 Summary

This thesis presented an approach for enhancing the individual learning experience of the prototypical Android game-based mobile application that is included in the Yactul system and used for playing quizzes. A method was introduced for integrating learning material support and enhancing the provided personalised feedback to help a student to improve himself. The proposed solution is inspired by existing personalised response systems. However, these approaches focus on teaching vocabulary or simple atomic tasks in general and most of them are solely based on flashcards. The fact that they treat each learning item as a separate task is also reflected in their underlying spaced repetition algorithms.

On the other hand, the proposed solution of this thesis does not deal with flashcards, but with a variety of question types, which are included in quizzes and can potentially treat the same or similar topics. Moreover, courses which group quizzes together, may overlap in their learning content and categories. As a consequence, to promote an understanding of complex, potentially interrelated learning areas, the underlying spaced repetition algorithm provides a categorisation of the activities. This is in contrast with traditional approaches. Therefore, an evaluation of existing spaced repetition algorithms was performed to evaluate whether an existing algorithm is suited for integrating a categorisation of learning topics, as well as other parameters, such as the time that the learner needed to answer an activity. The developed algorithm was finally based on the SuperMemo SM-2 spaced repetition algorithm, as it appeared to be the most flexible and best suited for these requirements.

While the sequential mode is used to play every activity of the chosen categories or weekly quizzes exactly once, the graph in the coaching mode uses the developed custom spaced repetition algorithm to decide which activities to present to the learner. This algorithm constitutes a trade-off between optimum repetition intervals and the freedom to let the learner decide which topics he wants to revise. To keep the motivation of a student, a special focus was given on using game-based elements. In both modes, colours were used to inform the student about his current knowledge state. Moreover, to help the learner to understand the correct answers of the activities, related learning material is provided to him, either directly after he played an activity, or if he touches long on a course or category in the coaching graph. The document support, the graph, the colour feedback and the spaced repetition algorithm result in the individual learning experience. However, there is room for improvement to provide a better and more individual experience, which will be discussed amongst other things in the next section.

5.2 Future Work

The presented personalised mobile response system provided an approach for integrating personalised feedback and document support to help the individual student to improve himself. However, there are different enhancements and extensions to improve the personalised learning experience. Some of them were already explained throughout this work.

The developed spaced repetition algorithm needs to be tested in practice to see if it really facilitates the learning tasks for different types of students, in terms of engagement, motivation, time investment and achieved learning outcomes. The obtained practical results could be used to fine-tune the different parameters of the spaced repetition algorithm, such as the maximum and minimum priority values, the calculations for the easiness factors of the categories and the activities, as well as the colour mapping of the priorities. Moreover, additional factors could be taken into account as well to make the algorithm more accurate and useful for university students, such as the number of rewarded European Credit Transfer System (ECTS) credits per course, assignment deadlines and exam dates.

For the prototypical coaching graph, there are several improvements and extensions. First, other abstraction layers can be added to include for example study programmes to support students that are enrolled in multiple programmes at the same time. Second, the visibility of the graph can be improved by showing only one part of it if the learner zooms out. Additionally providing two different graph view alternatives can further help to clearly arrange the graph. In the first one, the learner needs to explicitly fold and unfold the nodes of the graph by selecting the vertices that he wishes to examine in greater detail. This way, the graph does not get overloaded too quickly. In the second alternative, the entire graph will directly be visible, which allows the learner to get the global picture of his overall performance in all the courses and categories over the whole semester. Moreover, extending the idea of the graph to the sequential mode is also possible and will further enhance the game-based learning experience of the individual student. However, as this mode includes two sub-modes, namely the weekly quizzes and the categories, a graph view needs to be provided for each one, to keep a clearly arranged representation and to not overwhelm the learner with too many concepts at once.

Furthermore, in order to provide the student with the most recent learning material from the ALMA server, the documents could be received dynamically from

the server if the learner is connected to a WLAN. Otherwise, the learning sources could be retrieved from the coaching database.
The usage of the OrmLite framework only provides limited functionality due to its lightweight nature. As a result, in the implementation, there could be performance issues for large graphs, due to the nested loops, especially on older Android devices. Until now, there does not exist an optimal ORM technique for Android that provides all the necessary functionality. Therefore, using a server to store the database instead would be a viable option. However, the application would need to communicate with the back-end server over the HTTP protocol, which would require a constant Internet connection in order to be able to play quizzes. This would completely destroy the possibility to use the application offline and thus restrict the potential to be able to play quizzes anywhere at anytime. The best solution would therefore be to enhance the performance of the local SQLite database. However with current existing OrmLite techniques, this is hard to do for Android, but who knows which possible alternatives the future might bring.
Furthermore, the abstract programming concepts from the ALMA Core module need to be integrated in the Yactul server, because until now, the Android application works on a static JSON document, as there currently are no restrictions on the Yactul server, which presume the categories the teacher can provide when he creates or edits an activity.
The suggestion and presentation of the learning material can be further improved, in order to enhance the personalised learning experience. First, as the ALMA ontology and the Yactul server are developed at the same institution as the Android mobile application, namely the University of Luxembourg (Université du Luxembourg, 2017c), advantage can be taken from the fact that the whole system is available at the same facility. Thus extending the document support is relatively easily possible. For instance, students could receive the possibility to add their own, custom learning material, which could be shared among all the learners. Moreover, documents could be added for other programming languages, because currently, there is only support for the Java and C programming languages. In the mobile application itself, the learner could receive the possibility to rate the documents with up to 5 stars. This would inform him which learning material was the most valuable for him personally for a specific category. This would thus further enhance the individual learning experience.
Better personal feedback of the learner's progress, including the possibility to revise the results of the quizzes that he played in the past, could also be provided. The coaching database already includes tables to support such a feedback. However, the past quizzes are not yet saved in the database.
After each quiz, the student could receive the opportunity to consult a local overview of his performance, which is separated from his global achievement.
Finally, an additional extension could be to provide a discussion board to offer

students the possibility to talk about the solutions of the activities with their colleagues. The sharing of thoughts and reflections should imitate the traditional discussions that are often held in CRSs in the classroom between small groups of students or among the entire classroom, to talk about the solutions of a quiz (Roschelle, 2003; Scornavacca et al., 2009; Kaleta and Joosten, 2007; Penuel et al., 2007). Moreover, as students do sometimes understand the explanations of their colleagues better than the documents and answers offered by the teacher, providing the facility to discuss the solutions on a per activity basis among students can further help the individual learner to improve himself.

List of Abbreviations

ALMA Adaptive Literacy-aware learning Material IntegrAtion

API Application Programming Interface

AST Abstract Syntax Tree

BYOD Bring Your Own Device

CDT C/C++ Development Tooling

CRS Classroom Response System

DGBL Digital Game-Based Learning

ECTS European Credit Transfer System

EF Easiness Factor

E-Learning Electronic Learning

GSRS Game-based Student Response System

HTTP HyperText Transfer Protocol

ID IDentifier

IDE Integrated Development Environment

JDT Java Development Tools

JSON JavaScript Object Notation

M-Learning Mobile Learning

OF Optimal Factor

OpenGL Open Graphics Library

ORM Object-Relational Mapping

PDF Portable Document Format

REST REpresentational State Transfer

RNN Recurrent Neural Network

SRS Student Response System

SQL Structured Query Language

UML Unified Modeling Language

URI Uniform Resource Identifier

URL Uniform Resource Locator

WLAN Wireless Local Area Network

Bibliography

Apple Inc. (2017). SpriteKit. https://developer.apple.com/spritekit/. [last accessed: 29-July-2017].

Bahrick, H. and Phelps, E. (1987). Retention of Spanish Vocabulary Over 8 Years. *Journal of Experimental Psychology: Learning Memory and Cognition*, 13(2):344–349.

Bartel, A. and Hagel, G. (2014). Engaging students with a mobile game-based learning system in university education. In *2014 IEEE Global Engineering Education Conference (EDUCON)*, pages 957–960.

Brown, S. S. (2002–2017). Gradint, a program for self-study of foreign languages. http://people.ds.cam.ac.uk/ssb22/gradint/. [last accessed: 10-July-2017].

Brown, S. S. and Robinson, P. (2003). Addressing Print Disabilities in Adult Foreign-language Acquisition. In *Proceedings of the 10th international conference on human–computer interaction (HCII 2003)*, volume 4, pages 38–42.

Burguillo, J. C. (2010). Using game theory and Competition-based Learning to stimulate student motivation and performance. *Computers & Education*, 55(2):566 – 575.

Custers, E. J. (2010). Long-term retention of basic science knowledge: a review study. *Advances in Health Sciences Education*, 15(1):109–128.

Dellos, R. (2015). Kahoot! A digital game resource for learning. *International Journal of Instructional Technology and Distance Learning*, 12(4):49–52.

Deterding, S., Sicart, M., Nacke, L., O'Hara, K., and Dixon, D. (2011). Gamification. Using Game-design Elements in Non-gaming Contexts. In *CHI '11 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '11, pages 2425–2428, New York, NY, USA. ACM.

Ebbinghaus, H. (1913). *Memory: A Contribution to Experimental Psychology*. Number 3. Teachers College, Columbia University. (original 1885).

Ebner, M. and Holzinger, A. (2007). Successful implementation of user-centered game based learning in higher education: An example from civil engineering. *Computers & Education*, 49(3):873 – 890.

Edge, D., Fitchett, S., Whitney, M., and Landay, J. (2012). MemReflex: Adaptive Flashcards for Mobile Microlearning. In *MobileHCI 2012 Conference on Human Computer Interaction with Mobile Devices and Services*. ACM.

Elmes, D. (2017). Anki. https://apps.ankiweb.net/. [last accessed: 9-July-2017].

Erhel, S. and Jamet, E. (2013). Digital game-based learning: Impact of instructions and feedback on motivation and learning effectiveness. *Computers & Education*, 67:156 – 167.

FasterXML, LLC. (2017). Main Portal page for the Jackson project. https://github.com/FasterXML/jackson. [last accessed: 3-August-2017].

Felder, R. M. and Silverman, L. K. (1988). Learning and teaching styles in engineering education. *Engineering education*, 78(7):674–681.

Fies, C. and Marshall, J. (2006). Classroom response systems: A review of the literature. *Journal of Science Education and Technology*, 15(1):101–109.

Garris, R., Ahlers, R., and Driskell, J. E. (2002). Games, Motivation, and Learning: A Research and Practice Model. *Simulation & Gaming*, 33(4):441–467.

Godwin-Jones, R. (2010). Emerging technologies: from memory palaces to spacing algorithms: approaches to second-language vocabulary learning. In *Language, Learning & Technology*, volume 14, pages 4–11.

Google Inc. (2017a). OpenGL ES. https://developer.android.com/guide/topics/graphics/opengl.html. [last accessed: 29-July-2017].

Google Inc. (2017b). SurfaceView. https://developer.android.com/reference/android/view/SurfaceView.html. [last accessed: 5-August-2017].

Gray Watson (2017). OrmLite - Lightweight Object Relational Mapping (ORM) Java Package. http://ormlite.com/. [last accessed: 29-July-2017].

greenrobot (2016). greenDAO the best way to access SQLite. http://greenrobot.org/greendao/. [last accessed: 29-July-2017].

Grévisse, C., Botev, J., and Rothkugel, S. (2017). Yactul: An Extensible Game-Based Student Response Framework for Active Learning. In *Ponencias del XVIII Encuentro Internacional Virtual Educa*, Bogotá, Colombia.

Grévisse, C., Botev, J., and Rothkugel, S. (in press). An Extensible and Lightweight Modular Ontology for Programming Education. In *Proceedings of the 12th Colombian Conference on Computing*, Cali, Colombia. Springer CCIS.

Huotari, K. and Hamari, J. (2012). Defining Gamification: A Service Marketing Perspective. In *Proceeding of the 16th International Academic MindTrek Conference*, MindTrek '12, pages 17–22, New York, NY, USA. ACM.

Hwang, G.-J. and Wu, P.-H. (2012). Advancements and trends in digital game-based learning research: a review of publications in selected journals from 2001 to 2010. *British Journal of Educational Technology*, 43(1):E6–E10.

Kahoot! (2017). Kahoot! `https://kahoot.com/`. [last accessed: 7-July-2017].

Kaleta, R. and Joosten, T. (2007). Student Response Systems: A University of Wisconsin System Study of Clickers. *Research Bulletin*, 2007(10).

Khan Academy (2017). Khan Academy. `https://www.khanacademy.org/`. [last accessed: 8-July-2017].

Leitner, S. (1995). *So lernt man lernen: der Weg zum Erfolg*. Herder, 16 edition.

Loftus, G. R. (1985). Evaluating Forgetting Curves. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 11(2):397–406.

Malone, T. W. (1980). What makes things fun to learn? Heuristics for designing instructional computer games. In *Proceedings of the 3rd ACM SIGSMALL symposium and the first SIGPC symposium on Small systems*, pages 162–169. ACM.

Mario Zechner (2013). libGDX. `https://libgdx.badlogicgames.com/index.html`. [last accessed: 29-July-2017].

Mnemosyne (2017). The Mnemosyne Project. `http://mnemosyne-proj.org/`. [last accessed: 9-July-2017].

Mubarak, R. and Smith, D. C. (2008). Spacing Effect And Mnemonic Strategies: A Theory-Based Approach To E-Learning. In Nunes, M. B. and McPherson, M., editors, *e-Learning*, pages 269–272. IADIS.

Nielsen, K. L., Hansen, G., and Stav, J. B. (2013). Teaching with student response systems (SRS): teacher-centric aspects that can negatively affect students' experience of using SRS. *Research in Learning Technology*, 21(1):18989.

Papastergiou, M. (2009). Digital Game-Based Learning in high school Computer Science education: Impact on educational effectiveness and student motivation. *Computers & Education*, 52(1):1–12.

Penuel, W. R., Boscardin, C. K., Masyn, K., and Crawford, V. M. (2007). Teaching with student response systems in elementary and secondary education settings: A survey study. *Educational Technology Research and Development*, 55(4):315–346.

Piech, C., Bassen, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L. J., and Sohl-Dickstein, J. (2015). Deep Knowledge Tracing. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems 28*, pages 505–513. Curran Associates, Inc.

Pimsleur, P. (1967). A Memory Schedule. *The Modern Language Journal*, 51(2):73–75.

Poll Everywhere (2017). Poll Everywhere. `https://www.polleverywhere.com/`. [last accessed: 20-July-2017].

Reddy, S., Labutov, I., Banerjee, S., and Joachims, T. (2016). Unbounded Human Learning: Optimal Scheduling for Spaced Repetition. *arXiv preprint arXiv:1602.07032*.

Rohrer, D., Taylor, K., Pashler, H., Wixted, J. T., and Cepeda, N. J. (2005). The Effect of Overlearning on Long-Term Retention. *Applied Cognitive Psychology*, 19(3):361–374.

Roschelle, J. (2003). Keynote paper: Unlocking the learning value of wireless mobile devices. *Journal of Computer Assisted Learning*, 19(3):260–272.

Rothkugel, S. (2017a). Welcome to the C Programming Scrapbook. `https://mocca.uni.lu/teaching/programming2/scrapbook/`. [last accessed: 25-July-2017].

Rothkugel, S. (2017b). Welcome to the Programming 1 Scrapbook. `https://mocca.uni.lu/teaching/programming1/scrapbook`. [last accessed: 25-July-2017].

Schimanke, F., Mertens, R., Vornberger, O., and Vollmer, S. (2013). Multi Category Content Selection in Spaced Repetition Based Mobile Learning Games. In *2013 IEEE International Symposium on Multimedia*, pages 468–473.

Schmitz, B., Czauderna, A., Klemke, R., and Specht, M. (2011). Game Based Learning for Computer Science Education. In *Computer Science Education Research Conference*, CSERC '11, pages 81–86, Open Univ., Heerlen, The Netherlands, The Netherlands. Open Universiteit, Heerlen.

Scornavacca, E., Huff, S., and Marshall, S. (2009). Mobile phones in the Classroom: If You Can't Beat Them, Join Them. *Communications of the ACM*, 52(4):142–146.

Sharples, M. (2000). The design of personal mobile technologies for lifelong learning. *Computers & Education*, 34(3):177–193.

Siau, K., Sheng, H., and Nah, F. F. H. (2006). Use of a Classroom Response System to Enhance Classroom Interactivity. *IEEE Transactions on Education*, 49(3):398–403.

Socrative.com (2017). Socrative. https://www.socrative.com/. [last accessed: 7-July-2017].

Stack Exchange Inc. (2017). Top Questions. https://stackoverflow.com/. [last accessed: 26-July-2017].

Sung, H.-Y. and Hwang, G.-J. (2013). A collaborative game-based learning approach to improving students' learning performance in science courses. *Computers & Education*, 63:43 – 51.

SuperMemo World sp. z o.o. (1991–2017). SuperMemo. https://www.supermemo.com/en/frontpage. [last accessed: 8-July-2017].

The Game Creators Ltd. (1999–2017). AppGameKit. https://www.appgamekit.com/. [last accessed: 29-July-2017].

Thomason, E. (2017). Flashcards Deluxe. http://flashcardsdeluxe.com/flashcards/. [last accessed: 10-July-2017].

Trees, A. R. and Jackson, M. H. (2007). The learning environment in clicker classrooms: student processes of learning and involvement in large university-level courses using student response systems. *Learning, Media and Technology*, 32(1):21–40.

Université du Luxembourg (2017a). Bachelor Professionnel en Informatique. https://wwwen.uni.lu/formations/fstc/bachelor_professionnel_en_informatique/programme2. [last accessed: 25-July-2017].

Université du Luxembourg (2017b). Bachelor Professionnel en Informatique. https://wwwen.uni.lu/layout/set/print/formations/fstc/bachelor_professionnel_en_informatique/programme2?f=BAPROINFOR&c=0005C8655. [last accessed: 25-July-2017].

Université du Luxembourg (2017c). University of Luxembourg. https://wwwen.uni.lu/. [last accessed: 6-August-2017].

Verso Learning (2017). Verso. http://versoapp.com/. [last accessed: 20-July-2017].

Wang, A. I. (2015). The wear out effect of a game-based student response system. *Computers & Education*, 82:217–227.

Wozniak, P. A. (1998). SuperMemo 2: Algorithm. https://www.supermemo.com/english/ol/sm2.htm. [last accessed: 20-July-2017].

Wozniak, P. A. and Gorzelanczyk, E. J. (1994). Optimization of repetition spacing in the practice of learning. *Acta Neurobiologiae Experimentalis*, 54:59–62.