

LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN  
Department “Institut für Informatik”  
Lehr- und Forschungseinheit für Programmier- und Modellierungssprachen  
Prof. Dr. François Bry

**Master thesis**

## Gamification and Reification for Backstage 2

Manuel Hartman  
manuel.hartmann@stud.ifi.lmu.de

Bearbeitungszeitraum: 10. 11. 2017 bis 11. 06. 2018  
Betreuer: Sebastian Mader  
Verantw. Hochschullehrer: Prof. Bry



## Abstract

The motivation and competence of students to learn are important factors to complete their degree successfully; both factors can be self-regulated or supported by others. The aim of this work was to develop a framework (Progress Evaluation Framework) that helps both students and teaching staff to evaluate the students' learning progress during the term and thereby enable students themselves and teaching staff to support this learning progress and the motivation to learn. A special aspect of the Progress Evaluation Framework is that it allows the students to receive tangible and continuous feedback for their learning progress by continuously visualising the learning progress.

This framework is implemented as an extension of Backstage 2, which visualizes the learning progress of the students in a novel way. The students are encouraged to shape a visualisation, that is a landscape, to their likings, thereby creating a personalised visualisation of their learning progress. This personalised visualisation can help students and teaching staff to evaluate the learning progress by reifying the abstract task of learning in a visualisation and can also motivate students to stay committed because they want to improve and develop their visualisation further. The implementation of the Progress Evaluation Framework was tested for its usability and further improved following the feedback of this study. In the next step, its capacity to help students to stay motivated and focused on their learning goals should be evaluated in a long term study.

Die Motivation und Kompetenz der Studierenden zu lernen sind wichtige Faktoren für den erfolgreichen Abschluss des Studiums, beide Faktoren können selbst reguliert oder von Anderen unterstützt werden. Ziel dieser Arbeit war es, ein Framework (Progress Evaluation Framework) zu entwickeln, der sowohl den Studierenden als auch den Lehrenden hilft, den Lernfortschritt während des Semesters zu bewerten und damit die Studierenden selbst und die Lehrenden in die Lage versetzt, diesen Lernfortschritt und die Lernmotivation zu unterstützen. Eine Besonderheit des Progress Evaluation Framework ist, dass die Studierenden durch die kontinuierliche Visualisierung ihres Lernfortschritts ein konkretes und kontinuierliches Feedback zu ihrem Lernfortschritt erhalten.

Dieses Framework ist als Erweiterung von Backstage 2 (eine webbasierte Lernplattform) implementiert, das den Lernfortschritt der Studierenden auf eine neuartige Weise visualisiert. Den Studierenden wird ermöglicht, eine Visualisierung, in Form einer Landschaft, nach ihren Vorstellungen zu gestalten und so eine personalisierte Visualisierung ihres Lernfortschritts zu schaffen. Diese personalisierte Visualisierung kann Studierenden und Lehrkräften helfen, den Lernfortschritt zu bewerten, indem sie die abstrakte Tätigkeit des Lernens in einer Visualisierung vergegenständlichen und die Studierenden motiviert, engagiert zu bleiben, weil sie ihre Visualisierung verbessern und weiterentwickeln wollen. Die Implementierung des Progress Evaluation Framework wurde auf ihre Usability getestet und mit dem Feedback dieser Studie weiter verbessert. Im nächsten Schritt sollte die Eigenschaft, den Studierenden dabei zu helfen, motiviert zu bleiben und sich auf ihre Lernziele zu konzentrieren, in einer Langzeitstudie evaluiert werden.



## **Scope of work**

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbstständig angefertigt, alle Zitate als solche kenntlich gemacht sowie alle benutzten Quellen und Hilfsmittel angegeben habe.

München, June 3, 2018

.....



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Related work</b>	<b>3</b>
2.1	Gamification . . . . .	3
2.2	Intrinsic and Extrinsic Motivation . . . . .	5
2.3	Loss Aversion . . . . .	7
2.4	Feedback . . . . .	8
2.5	Self-Regulated Learning . . . . .	10
<b>3</b>	<b>Progress Evaluation Framework</b>	<b>13</b>
3.1	Progress Review Framework (PRF) . . . . .	17
3.1.1	Representation of courses – The course model . . . . .	18
3.1.2	Evaluation of accomplishments – The point system . . . . .	19
3.1.3	Interface to Backstage . . . . .	20
3.1.4	Interface to Progress Visualisation Framework . . . . .	20
3.2	Progress Visualisation Framework (PVF) . . . . .	20
3.2.1	Cultivating a garden . . . . .	21
3.2.2	Robot construction kit . . . . .	22
3.2.3	Creating a landscape . . . . .	22
<b>4</b>	<b>Development</b>	<b>25</b>
4.1	Software Stack . . . . .	25
4.2	Form Concept to Implementation . . . . .	25
4.2.1	User Interface . . . . .	27
4.2.2	Business Logic . . . . .	32
4.2.3	Input and Output . . . . .	32
4.2.4	Data Model . . . . .	33
4.3	Implementation . . . . .	34
4.3.1	Client side . . . . .	34
4.3.2	Server side . . . . .	39
<b>5</b>	<b>Evaluation</b>	<b>41</b>
5.1	Introduction . . . . .	41
5.2	Executive Summary . . . . .	41
5.3	Methodology . . . . .	41
5.3.1	Sessions . . . . .	42
5.3.2	Participants . . . . .	42
5.3.3	Evaluation Tasks/Scenarios . . . . .	42
5.4	Results . . . . .	42
5.4.1	Task Completion Success Rate . . . . .	43
5.4.2	Time on Task . . . . .	43
5.4.3	Errors . . . . .	44
5.4.4	Summary of Data . . . . .	44
5.4.5	Overall Metrics . . . . .	45
5.5	Recommendations . . . . .	45
5.5.1	Free exploration (Task 1) . . . . .	45
5.5.2	Edit the landscape (Task 2) . . . . .	46
5.5.3	Edit the background landscape (Task 5) . . . . .	46
5.5.4	Save the changes (Task 6) . . . . .	46
5.6	Usability Test Summary . . . . .	46

<b>6 Conclusion</b>	<b>47</b>
<b>A Data Model</b>	<b>55</b>
A.1 Progress Evaluation Framework . . . . .	55
A.1.1 Reification Object . . . . .	55
A.1.2 Segment Object . . . . .	56
A.1.3 Mission Object . . . . .	57
A.1.4 Visualisation Object . . . . .	58
A.2 Third Party Tools . . . . .	61
A.2.1 ShoeBox sprite sheet JSON format . . . . .	61
A.2.2 Tiled Map Editor JSON map format . . . . .	62
<b>B File Listing</b>	<b>64</b>
B.1 Client Side . . . . .	64
B.1.1 Assets . . . . .	67
B.2 Server side . . . . .	68
B.3 Share . . . . .	69
<b>C Usability Test</b>	<b>70</b>
C.1 Background Questionnaire . . . . .	70
C.2 Post-Task Questionnaire . . . . .	70
C.3 Post-session Overall Subjective Questionnaire . . . . .	71
C.4 Task Scenarios . . . . .	71



### 1 Introduction

To understand and improve the way people learn and how to motivate them to learn has received much attention by research in recent years and multiple scientific fields have produced important findings. With the introduction of the bachelor / master system to the German academic teaching as part of the Bologna Process, studying has changed and learning has to meet the needs of students of personal support even in large courses. Gamification uses an interesting set of techniques from game design to create more engagement and can be used for providing feedback to users [110]. The focus of gamification is often rather on the economic interests of the operating companies [111] and not on a positive impact towards the ability of self-determination and self-regulation of users. However, this work wants to help students to develop the ability of self-determination and self-regulation as part of their learning process. To achieve this it develops a tangible and continuous feedback system, that allows users to visualize their learning progress and thereby they are enabled to evaluate and reify it. This visualisation gives feedback about the learning progress, can improve self-estimation, and simplifies comparability.

Therefore, this work does not only focus on the findings around gamification, but instead integrates findings of scientific research in multiple fields. Out of practical observation and experience the concept of gamification was established and then proven by research, where game concepts and mechanics are used in non-game contexts [110, 17]. With the incentive theories a model for intrinsic and extrinsic motivation exists which helps to understand the motivation of people to do things [15, 77, 13]. There are well-established theories for feedback based on the research of behavioural psychology, which describe when and how to give feedback to achieve requested results [38, 37]. Educational psychology provides the concept of self-regulated learning, where learners create and set learning goals for themselves and overhaul and adapt these goals during and after the learning process [65, 68]. Even the field of behavioural economics has interesting findings (loss aversion) that can help to improve the way people learn [47, 46].

However, although individual aspects like engagement, motivation, feedback, and self-regulated learning are explored, there is a lack of work that combines all these findings in one learning system to study its impact on the learning behaviour of students. The present thesis presents a framework to allow students to evaluate and visualize their learning progress and teaching staff to assess and support the students' learning progress. Starting from the idea of a tangible and continuous feedback system with game elements this work began with an extensive literature research, to have a solid foundation to build on. The second chapter discusses the results of this literature research and then in the third chapter originating from the concepts of gamification the Progress Evaluation Framework is developed. To fulfil its goals the Progress Evaluation Framework combines gamification with concepts of the incentive theories, and educational psychology and introduces the concept of "reification" for visualizing, and reifying abstract work (like learning). The framework has two components one that evaluates students' learning progress (Progress Review Framework) and a second that allows them to reify their progress with an individually customisable visualisation (Progress Visualisation Framework). Different possible visualisations are explained and compared in this section. In the fourth chapter the implementation of both components for the learning management system Backstage 2 [69] are characterized. For the Progress Visualisation Framework the visualisation "creating a landscape" is implemented, where the students can shape a landscape, to reify and visualize their learning progress. The next chapter examines a usability study of the implemented visualisation and its editor. This study showed that the user interface concept is working and it also showed promising reactions of the participants towards the usefulness of the framework. The last chapter summarises the results and findings of this thesis and discusses possible future work.



## 2 Related work

This work is based on the scientific results of prior research in several scientific fields. The most relevant of which are gamification, intrinsic and extrinsic motivation as part of incentive theories, loss aversion as part of behavioral economics, feedback as part of behavioral psychology, and self-regulated learning as part of social psychology. The next sections introduce these scientific fields, their ideas and assumptions which are relevant for this thesis. Gamification is the the base for the development of this work but because it has different goals then “common gamification” it is necessary to consider other research fields as well to modify and extend gamification.

### 2.1 Gamification

The term “gamification” emerged in the digital media industry and is used there by business and marketing departments [82]. Towards the end of the Year 2010 the term gets more popular [17], but it existed since 2002 when it was first used by Nick Pelling [86]. Other terms for gamification include “productivity games” [60], “surveillance entertainment” [31], “funware” [87], “playful design” [25], “behavioral games” [19], “game layer” [70] or “gameful design” [17]. “Gamification” is the most widely used of all these terms and has been adopted in scientific work.

The basic idea behind gamification is not quite new and has its roots in loyalty and bonus programs. For example, airlines have had frequent-flyer programs since the 1980s and coffee shops or sandwich bars often use a loyalty program based on the rule “buy 10 get 1 free” [110]. Foursquare was one of the first well-known services that used gamification and helped it to gain traction in interaction design and digital marketing [111]. It is seen as a future method for marketing and customer engagement [110].

Not only industry is interested in gamification, also academia [17, 61, 40] has begun to investigate the potential of gamification. Deterding sees a combination of several factors as the reason for great increase of gamification: 1) the possibility to track everyday activity with cheap sensor, 2) web analytics that allow individual tracking, and 3) the broad acceptance of video games [16].

Gamification is the idea to use game elements, mechanics, and dynamics outside of games to increase user engagement with a system. Research in different academic areas showed that game elements can successfully be used for increasing user motivation and engagement. Seaborn and Fels found research in the domain of: education, online communities and social networks, health and wellness, crowdsourcing, sustainability, orientation, computer science and engineering, research, marketing and computer-supported cooperative work [82]. The reason to gamify a system is to encourage behaviour change in users of a system. The desired behaviour change can be, e.g., increased participation, improved performance, or greater compliance [82].

Human-computer interaction (HCI) has been trying to use game design and game elements in other contexts for quite a long time. Games and the fact that they are fun to play was subject of research as early as in the 1980, especially how to use game elements for enjoyable interfaces [59, 58]. Wright et al. searched under the name “funology” for design patterns for joy of use with direct reference to game design [108].

Games with a purpose (GWAP) are another approach, which uses human participation and knowledge to solve problems a computer system cannot solve on its own with games [93]. Here the game is used to make a rather simple or boring task fun for the users. McGonigal establishes alternate reality games (ARGs), as a concept of “antiescapist” games that try to change how people act and think in their everyday lives [61]. Seaborn and Fels carefully expresses that ARGs can be seen as “the fully-fledged counterpart to gamification” [82].

Deterding et al. defines gamification as “the use of game design elements in non-game contexts” [17]. They see in the term gamification a new umbrella term related to the idea of gamefulness, gameful interaction, and gameful design [18, 17]. The distinction between gamefulness, gameful interaction, and gameful design in contrast to the established concepts of playfulness,

playful interaction, or design for playfulness is fundamental for Deterding et al. [18, 17]. Gamification does not lead to full-fledged games which is an important distinction to serious games and GWAPs [17]. All three use elements of games in a new way out of the normal usage in an entertainment game, but gamified systems only incorporate elements of games and do not want to create the full game experience [17]. Deterding et al. see improvement of the user experience, which can also be described as the joy of use or amount of engagement, as the most likely design goals for a gameful experience [17].

Huotari and Hamari take the different view of service marketing to define gamification as: “a process of enhancing a service with affordances for gameful experiences in order to support user’s overall value creation” [40]. This definition says that if a service is able to support the user with a gameful experience, then it has gamification. More over the experience of gamification is subjective and defined by the user [40]. They stress that gamification is a process where the goal is to create a gameful experience for the user by complying the service with affordances for that purpose and cannot be based on a set of methods or mechanics [40].

Gamification normally is implemented by adapting the scoring elements of video games, like levels, points, and achievements for the context of the application [64].

We also have to understand what a game is. The research field of game studies stresses the idea that games are built from multiple necessary conditions and properties. There is not that single condition that creates a game by its own in addition, the properties go beyond game artefacts and include the situated, socially constructed meaning [10, 88, 44, 17]. A quite simple definition is: “A system in which players engage in an artificial conflict, defined by rules, that results in a quantifiable outcome” [80, p. 80].

Juul defined a game as: “a rule-based formal system with a variable and quantifiable outcome, where different outcomes are assigned different values, the player exerts effort in order to influence the outcome, the player feels attached to the outcome, and the consequences of the activity are optional and negotiable.” [43]. He further proposes six main features that every game has: rules, variable, quantifiable outcomes, value-laden outcomes, player effort, player investment, and negotiable consequences, with respect to real life effects [44].

For this thesis the distinction between game and play is important. Caillois introduced his concept of *paidia* and *ludus* to describe two different aspects of play [8]. On the one hand *paidia* (playing) characterizes free-form, expressive, improvisational behaviors and meanings and on the other hand *ludus* (gaming) describes rule-based playing with determined goals. Gamification mostly focuses on design elements of gaming (*ludus*) [17, 32]. Therefore, Deterding et al. suggest to adapt McGonigal’s [61] term “gamefulness” [17]. “Gamefulness” is the counterpart for “playfulness”.

Gamification and its concepts are criticized from different scientific areas. Game designers criticise the existing stock implementations of gamification (that add points, badges, and leaderboards to motivate) as too simple [6, 49, 55] and not implementing the essential parts of games [74]. Robertson goes so far that gamification only implements “the thing that is least essential to games and representing it as the core of the experience” [74, para. 4]. Not feedback systems create the pleasure of games, but the possibility of meaningful choices in the pursuance of interestingly hard goals [74]. Bogost uses the degrading phrase of “exploitationware” for gamification [5], as software that extracts real work and value from users in return for virtual points or tokens [6]. Gamification often has a focus on extrinsic motivation instead of intrinsic motivation [64, 82], which leads to two problems: first, extrinsic rewards induce the “reward loop”, where the user expects a reward for every task they do in the system and the system has to keep them in this loop forever [110]. Second, extrinsic motivation is known to reduce intrinsic motivation.

Deterding et al. see gamification as a tool to create engaging workplaces or facilitate mass-collaboration [18]. Kapp predicts a “dramatic increase in gamification of learning and instruction” [48, para. 8]. Muntean, for example, uses gamification to cause certain behaviours or rectify others and sees in that possibility a benefit for e-learning [63].

Hamari et al. as well as Seaborn and Fels showed in their surveys that most of the reviewed

studies attest gamification positive effects and benefits [36, 82]. Positive effects on engagement and enjoyment by gamification was found by [21, 56, 62]. Even if most of the studies “reported positive results for some of the motivational affordances of the gamification implementations” [36, p. 4], a detailed consideration of the effects and when and how they are working is necessary. Gamification does not work in all scenarios; for example in utilitarian service settings it might not be effective [35, 42]. The effects gamification has can be out of curiosity of the user for this new idea and may not last in the long term [35, 23, 24], but removing gamification while user are still engaged can have negative implication [89]: here loss aversion can be a factor [34]. Gamified learning tasks showed increased motivation, engagement and even enjoyment in all the studies [36], but also negative effects were found, like increased competition [33], task evaluation difficulties [20] and design features [21, 20]. Users with high motivation is one of the goals of gamification and it also is important for the topic of this thesis. To get a deeper understanding of motivation the next section establishes the concept of intrinsic and extrinsic motivation.

## 2.2 Intrinsic and Extrinsic Motivation

Motivation is the urge to do something. Ryan and Deci describe a motivated person as “someone who is energized or activated toward an end” [77, p. 54] and an unmotivated person as someone “who feels no impetus or inspiration to act” [77, p. 54]. In another paper they write that “the fullest representations of humanity show people to be curious, vital, and self-motivated” [78, p. 68], arguably this is the state of mind that students should have. To get there Ryan and Deci postulate that the psychological needs of competence, autonomy, and relatedness need to be satisfied [77].

People can have highly different reasons for their motivation. Ryan and Deci distinguish between different pairs of oppositional motivation: “value an activity” vs. “external coercion”, “abiding interest” vs. “a bribe”, and “a sense of personal commitment to excel” vs. “from fear of being surveilled” [78, p. 69]. In all these examples the first reason for motivation is internally motivated and the second one on the other hand externally pressured, all these causes of motivation are used by people as a fundamental dimension to add up their own and others’ behaviour [11, 39, 76]. When compared, people with a self-endorsed motivation show more interest, excitement, and confidence than people with only external control for an activity and as a result the first group has enhanced performance, persistence, and creativity [75, 83], self-esteem [14], and general well-being [79].

Ryan and Deci define intrinsic motivation as “the doing of an activity for its inherent satisfactions rather than for some separable consequence” [77, p. 56]. An intrinsically motivated person does not need any external incentives, pressures, or rewards, he or she is moved to act because of curiosity, fun, or challenge of an activity. White found the phenomenon of intrinsic motivation first, when studying animal behaviour [96]. Many animals engaged in exploratory, playful, and curiosity-driven behaviours without reinforcement or reward [96]; the same is true for humans from birth onwards [77]. Intrinsic motivation is in the human nature, a well being person is going “to seek out novelty and challenges, to extend and exercise one’s capacities, to explore, and to learn” [78, p. 70]. Skinner interpreted intrinsic motivation from the view of operant theory, in which all behaviour is motivated by rewards, and therefore the activity itself had to be the reward for intrinsically motivated activities [85]. Ryan et al. rather emphasises on the innate psychological needs for competence, autonomy, and relatedness [77].

Beside intrinsic motivation as an important type of motivation, there is an other type of motivation which can also be self-determined [15]. Extrinsic motivation occurs whenever “an activity is done in order to attain some separable outcome” [77, p. 60]. After early childhood, many activities are technically not intrinsically motivated. Other factors like social pressure or instrumental value replace intrinsic motivation [78]. This is especially true for educational activities, which are often not designed to be intrinsically interesting, but rather to engage students in such activities, so they value and self-regulate the activity and carry them out on their own without external pressure

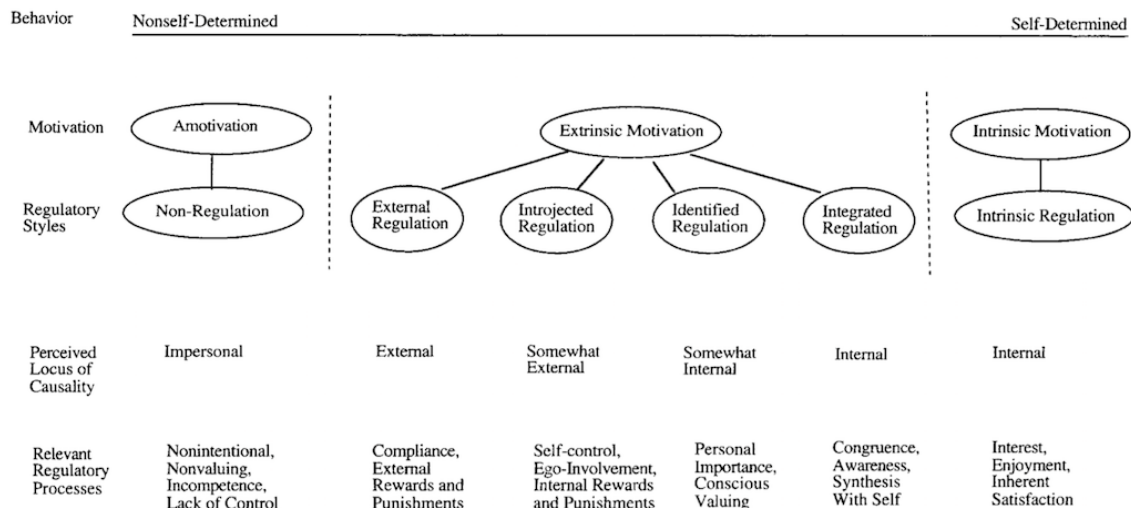


Figure 2.1: From Ryan et al. illustrates the dimensions of motivation in the self-determination theory with regulatory styles, perceived locus of causality, and relevant regulatory processes [78, p. 72].

[77].

The effects of extrinsically motivated activities on ongoing persistence, behavioural quality, and well-being are topic of several studies e.g.[78, 12, 13]. Deci found that extrinsic rewards can undermine intrinsic motivation [12] and Deci et al. further affirmed this finding with a meta analysis on motivation in education [13]. They reviewed 128 studies and nearly all showed a negative effect from external reward on internal motivation [13]. To counter this negative effect a shift of focus away from rewards to motivate students is necessary, to an activity design that is capable to offer more interesting learning activities, to provide more choice, and to ensure that activities are optimally challenging [13].

With the Self-Determination Theory (SDT) Deci & Ryan introduced a concept to categorise motivation based on diverse reasons and goals, for which someone wants to do something [15]. The basic distinction between intrinsic and extrinsic motivation is kept, but both intrinsic and extrinsic motivation have been further differentiated by detailed theories.

The Cognitive Evaluation Theory (CET) is part of the Self-Determination Theory and proposed “that underlying intrinsic motivation are the innate psychological needs for competence and self-determination” [13, p. 3]. It was introduced by Deci & Ryan to show in detail factors in social contexts which generate difference in intrinsic motivation [15]. The Cognitive Evaluation Theory reasons “that interpersonal events and structures (e.g., rewards, communications, feedback) that conduce toward feelings of competence during action can enhance intrinsic motivation for that action because they allow satisfaction of the basic psychological need for competence” [77, p. 58].

The Organismic Integration Theory (OIT) is part of the Self-Determination Theory and was established “to detail the different forms of extrinsic motivation and the contextual factors that either promote or hinder internalization and integration of the regulation for these behaviors” [77, p. 61] [78, p. 72]. Figure 2.1 illustrates the organismic integration theory and shows the different motivational types, arranged from left to right in terms of the degree to which the motivations go out from the self.

Intrinsically motivated students are what education desires and with the Self-Determination Theory there is a framework which can help to orchestrate students motivation but beside motivation there are other psychological effects which can be used to manipulate the activity of students.



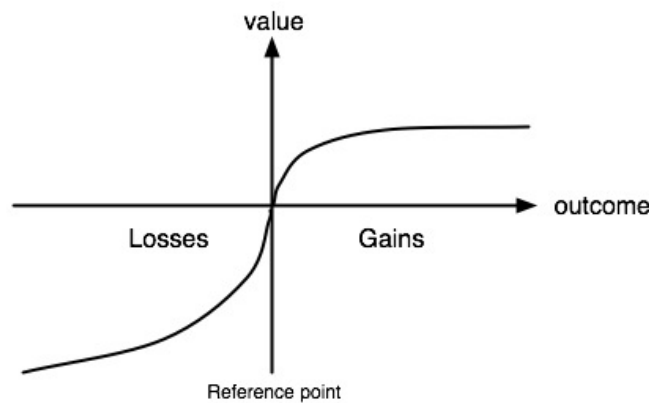


Figure 2.2: Visualisation from [103], that illustrates the value function for loss aversion, where every person rates losses and gains from an individual reference point [45]. The x-axis shows the the expected outcome and the y-axis shows the value with this outcome is assessed.

The scientific research in the field of decision theory shows promising approaches, with loss aversion to be introduced in the next chapter.

### 2.3 Loss Aversion

In economics and decision theory, loss aversion describes the phenomenon of losing an object is attached more importance by people than gaining the same object. For example, when someone loses 10€ they will lose more satisfaction, compared to the satisfaction the same person gains when they find 10€ [99].

Loss aversion was identified by Kahneman and Tversky [46] and is part of the prospect theory [47]. The prospect theory concludes that individuals in decision-making situations behave irrationally when uncertainties play a role [47]. This violates the assumption of the neoclassical economic decision theory that homo economicus is used to maximising and acts rationally [34]. People do not rate the utility of an outcome by its absolute value in wealth or welfare, but rather from a changing neutral reference point [45]. They further describe that negative changes (losses)



Figure 2.3: Dead crops on a field in FarmVille after the player missed the time slot for harvesting [9].

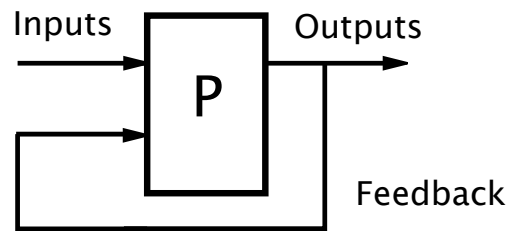


Figure 2.4: A feedback loop where the output is routed back as a input of the process [97].

are experienced more strongly than positive changes (gains) [45]. One study suggested that the ratio of the slopes of the value function in two domains is about 2:1 [90]. Figure 2.2 shows a schematic value function to illustrate the different ratio. In the context of behavioural economics, loss aversion is related to several other biases.

Hamari had a look on how social games mechanics and design patterns can be explained with loss aversion and the prospect theory [34]. One popular mechanic in social games he described is the decay of crops in farming games, where players must login and harvest these crops regularly [34]. For example in FarmVille the crops wither, if they are not harvested in time (Figure 2.3). This mechanic of decay triggers loss averse tendencies, to provoke players to return to the game and to increase customer retention and it can be transferred to other in game possessions [34].

The last two sections discussed psychological phenomena which can be used to manipulate or orchestrate students indirectly but in education a very obvious form of inducement is common namely feedback.

## 2.4 Feedback

A technical view on feedback would describe it as the back routing of outputs of a system to its inputs, which leads to a manipulation of the next outputs and thereby forms a loop (Figure 2.4) [97]. For education Butler and Winne defined feedback as “information with which a learner can confirm, add to, overwrite, tune, or restructure information in memory, whether that information is domain knowledge, meta-cognitive knowledge, beliefs about self and tasks, or cognitive tactics and strategies” [7, p. 275]. In the first definition feedback is a simple signal but in the second feedback is an information “about the content and/or understanding of the constructions that students have made from the learning experience” [38, p. 82], that can be accepted, modified, or rejected by the student [54]. The main target of feedback in the context of education is “to reduce discrepancies between current understandings and performance and a goal” [38, p. 86]. To fulfil this target it is important that the feedback is aimed at students at the proper level, because only then feedback is able to deploy its full potential and lead to the requested effect of narrowing the gap of desired and current knowledge [38].

Hattie and Timperley developed a framework to assist the task of giving feedback (Figure 2.5), that can help to lead to a more productive outcome [38]. From the perspective of the students three important questions need to be answered by the feedback they receive: “Where am I going?”, “How am I going?”, and “Where to next?”. Alternatively these dimensions can be named feed up, feed back, and feed forward [38].

**Where Am I Going?** The answer to this question, should give a student a clue where they stand in the moment in regard to the attainment of the learning goals [38]. This feedback creates a relation between overall learning goals and the concrete task it is given for. Hattie and Timperley give some examples how different such judgements can look like [38]:



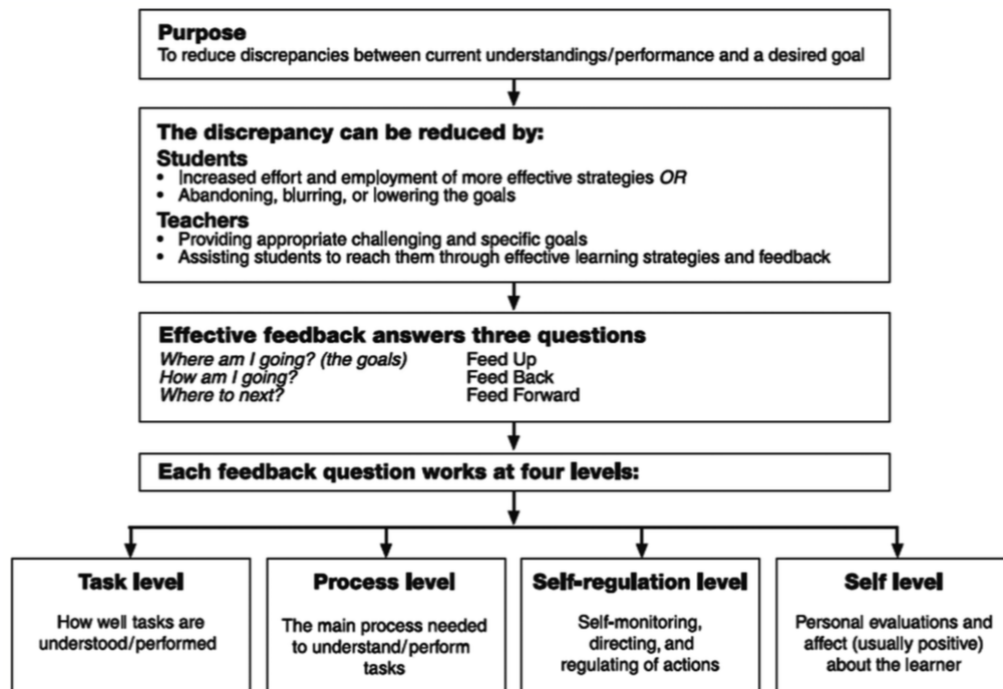


Figure 2.5: Hattie and Timperley’s model of feedback, shows the three questions: “Where am I going?”, “How am I going?”, and “Where to next?” and the four levels: task, process, self-regulation, and self, that feedback should answer to be effective [38, p. 87].

- direct about the task, like “passing a test” or “completing an assignment”;
- comparative to other performances, like “doing better than Mary” or “doing better than last time”;
- social in context to the environment, like “not getting a detention” or “seeking teacher approval”;
- engagement related, like “singing a song” or “running a race”; or
- automatic and triggered outside of specific awareness, like “doing well on a task” or “seeking more challenging tasks”.

**How Am I Going?** The answer to this question compares the performance of a task with some anticipated standard, to previous accomplishments, and/or to success or failure on a certain part of the task; this feedback can be given by a teacher, a peer, or a themselves [38]. Feedback is especially effective, if it consists of information about progress, and/or about how to proceed [38].

**Where to Next?** The answer to this question should give the student information that induces to greater potential for learning. Hattie and Timperley describe that the common reaction to the question “Where to Next?” is the simple answer “more”, more information, more tasks, and more expectations [38]. Instead, feedback should lead to “enhanced challenges, more self-regulation over the learning process, greater fluency and automaticity, more strategies and processes to work on the tasks, deeper understanding, and more information about what is and what is not understood” [38, p. 90]; only then this feedforward question can bring great improvements on learning.

In Hattie and Timperley's framework the three questions can be answered at four major levels (Figure 2.5) and the effectiveness of feedback depends on the level it is directed [38]. The four levels feedback can be directed to are feedback about the task (FT), feedback about the processing of the task (FP), feedback about self-regulation (FR), and feedback about the self as a person (FS) [38].

**Feedback about the task** Feedback is directed at the task and is a statement if the presented solution is valid or invalid. Feedback about the task is also known as corrective feedback and is related to task accomplishment and reviews correctness, neatness, behaviour, or some other criteria [38]. It is the most common type of feedback, but has the problem that it is hard for students to be generalised to other tasks and can lead to strategies where students focus on an immediate goal and not the big picture [38].

**Feedback about the processing of the task** Feedback is aimed at the process which was used to solve a task. This level of feedback has shown to be more effective in enhancing deeper learning than feedback on the task level [2]. Several studies stressed on the item of a deep understanding of learning and concluded that it involves the construction of meaning (understanding) and thereby helps to solve more difficult or untried tasks [71, 94, 81]. Earley et al. found an interactive effect between feedback about the task and feedback about the processing of the task, where feedback about the task can enhance task confidence and self-efficacy, which leads to more effective and creative information and strategy searching [22].

**Feedback about self-regulation** Feedback is directed at the students' self-regulation level and can stimulate the skill in self-evaluation or the confidence to engage further on a task. Commitment, control, and confidence and how they interact is part of self-regulation, which "addresses the way students monitor, direct, and regulate actions toward the learning goal" [38, p. 93]. Feedback at students self-regulation level has effects on "the capability to create internal feedback and to self-assess, the willingness to invest effort into seeking and dealing with feedback information, the degree of confidence or certainty in the correctness of the response, the attributions about success or failure, and the level of proficiency at seeking help" [38, p. 93].

**Feedback about the self as a person** Feedback is personal and it is directed at the "self". Personal feedback, such as "Good girl" or "Great effort," is often no real feedback to the task, it has little or no information about performing the task, and the student's self-concept influences the reception [38]. If the praise does not contain any reference to the task it cannot answer any of the three questions and sidetracks attention from the task, but it is also possible to recognise the work, self-regulation, commitment, or processes relating to the task and its performance [38].

With the right approach in giving feedback in combination with the other concepts discussed in this chapter this work wants to support student's learning and foster their Self-Regulated Learning.

## 2.5 Self-Regulated Learning

Self-Regulated Learning (SRL) describes the ability to create and set learning goals for yourself and to overhaul and adapt these goals during and after the learning process [65]. Pintrich and Zusho defined self-regulated learning as "an active constructive process whereby learners set goals for their learning and monitor, regulate, and control their cognition, motivation, and behaviour, guided and constrained by their goals and the contextual features of the environment" [68, p. 64]. Nicol and Macfarlane-Dick emphasised the fact that this definition widens to the aspects of motivation, behaviour, and external requirements [65].

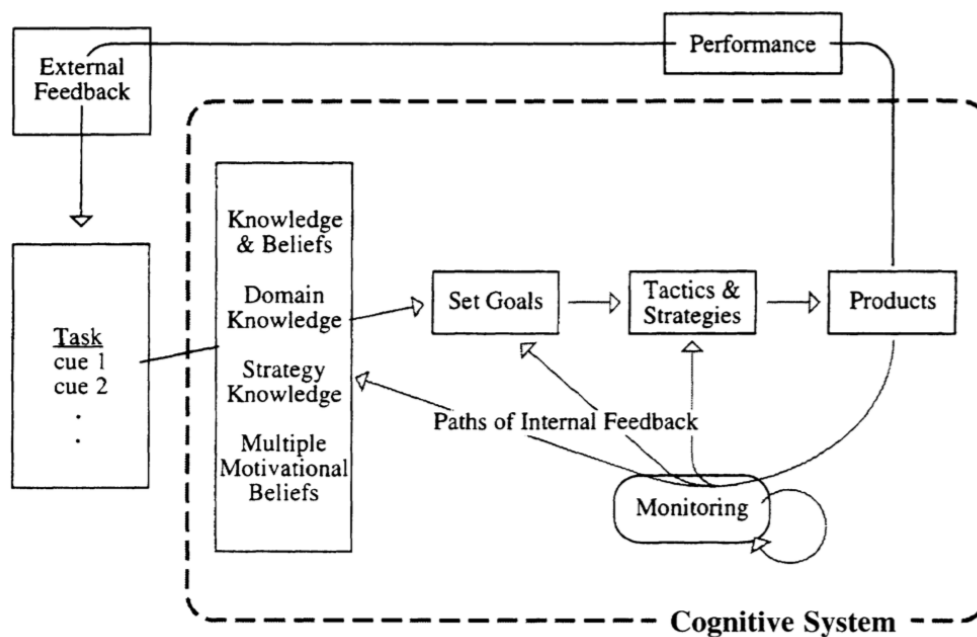


Figure 2.6: Butler and Winne's model of self-regulated learning, from [7, p. 248].

Based on the ideas of Kuhl and Goschke [53], Butler and Winne developed a model of self-regulated learning (Figure 2.6). For students self-regulated learning normally begins with an external academic task. Based on their knowledge and beliefs they analyse the task's properties and requirements and build an interpretation of the task. The next step for students is to set goals adapted from their interpretation. Then they apply their tactics and strategies on the goals to create products (e.g. the solved task or newly acquired knowledge). [7]

By monitoring these processes on the mental and behavioural sides, students generate internal feedback about themselves for their engagement and the progressively arising products. The internal feedback is then the basis to adjust the interpretation of elements of the task and adapt the engagement. When students adapt their engagement, it can lead to creating new goals and resetting or extending existing ones, reevaluating tactics and strategies to adapt available skills or even adding new ones. [7]

External feedback on their performance gives students additional information, which may confirm, add to or conflict with their interpretations of the task and the learning progress. This feedback can be used by the students to alter knowledge and beliefs, which in the next step can be used to reinterpret subsequent self-regulation. [7]

If somebody masters self-regulated learning, they unlock the most effective way of learning [7]. To help and enable students to improve their abilities in self-regulated learning is the overall goal of this thesis and the next chapter specifies in detail what it does.



### 3 Progress Evaluation Framework

The Progress Evaluation Framework renders interactions of students with Backstage 2 as personalised and customisable visualisations. This course based visualisations illustrate the learning progress to motivate students and give them feedback about their learning progress. Figure 3.1 shows the implemented visualisation of “creating a landscape”. The hill with the tree in the middle represents the lecturer “Prädikatenlogik erste Stufe” (First-Order Predicate Logic), which is part of the course “Logik und diskrete Strukturen” (Logic and discrete Structures), here a student has shaped the landscape and planted a tree to visualise their learning progress. The left part of the visualisation is unchanged, the flat grass land is the default state. The desert in the right thirds represents another lecture where this student did not achieve their learning goals. The following sections clarify the considerations, which lead to this visualisation.

The goal of this work is to build a reification system, which is explained below, for Backstage 2 [69], consisting of two parts, the Progress Review Framework for evaluating the learning progress of students and Progress Visualisation Framework allowing students to visualise their progress. Such a reification system can help students and teaching staff. Students can use it to evaluate their learning progress and compare themselves to each other if they want. The teaching staff can use this system to observe the learning progress of students. The primary user group are bachelor students at the beginning of their studies but students in every stage of their studies

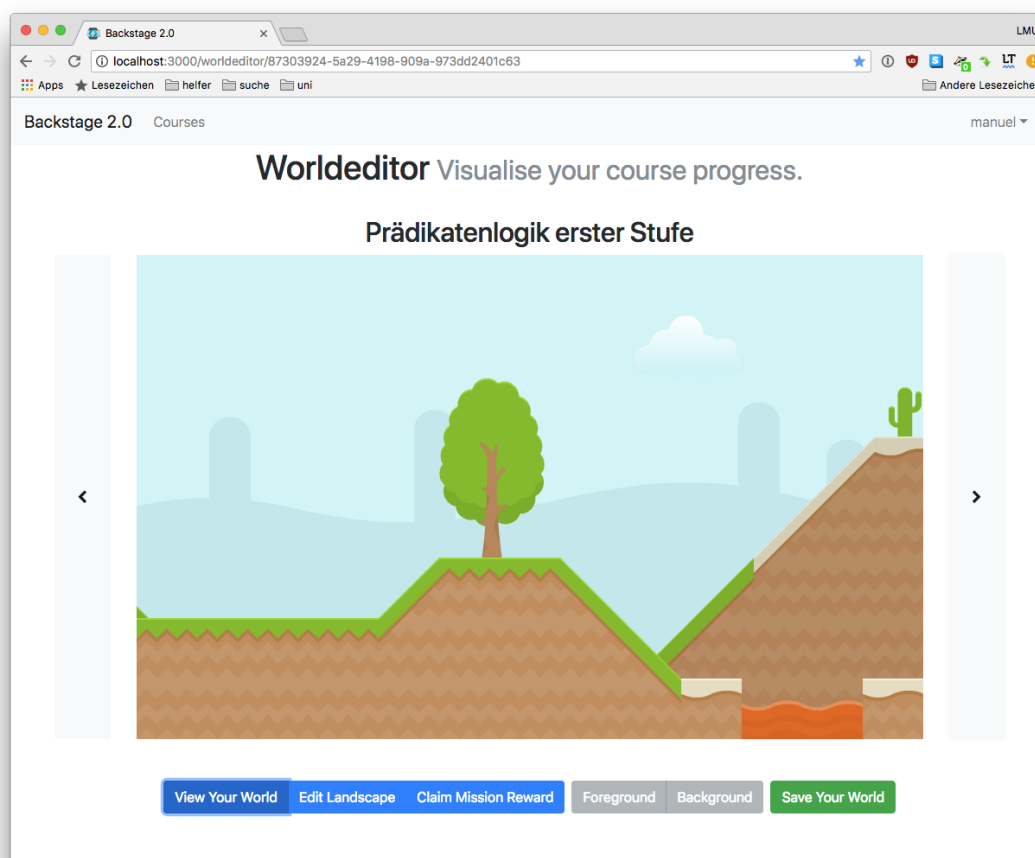


Figure 3.1: Screen shot of the visualisation “creating a landscape”, that shows the visualisation of the lecture “Prädikatenlogik erste Stufe” in the middle and also the visualisations of the preceding and following lecture.

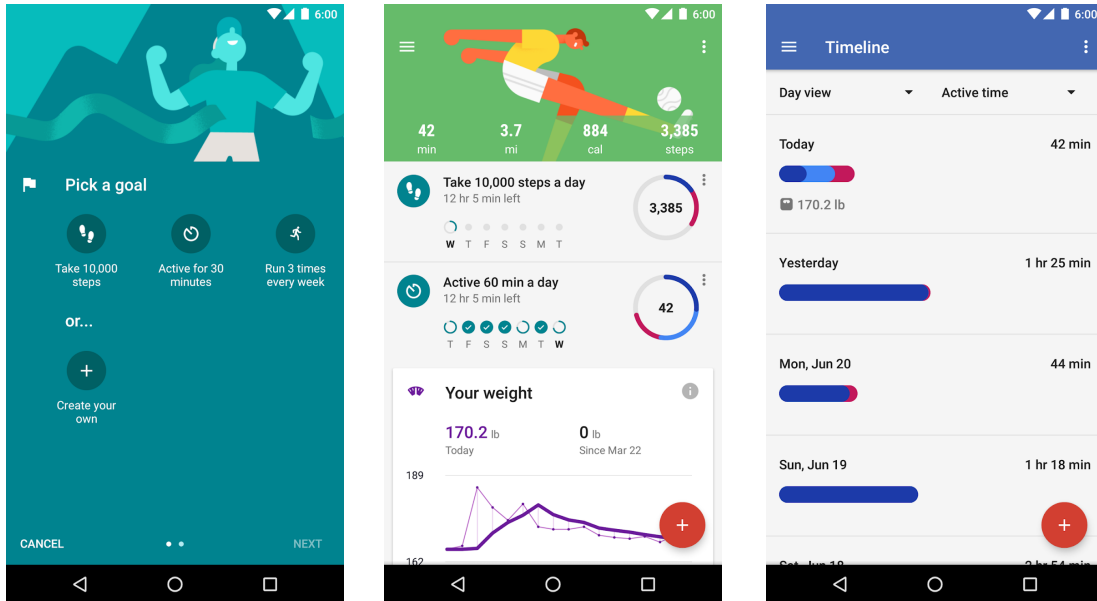


Figure 3.2: Screenshots of Google Fit (by [30]) form left to right: setting goals for the training, overview of the activities of this day, and a timeline of the activities of the last days.

can benefit and the reification system could be adapted for any task based work. Freshmen often have to attend mass lectures and do not have built up a learning routine yet. In mass lectures with several hundred participants teaching staff cannot take care of every student individually. In this situation a system that reviews the learning progress of students can help both the teaching staff to support struggling students when necessary and the students to built up a sustainable learning behaviour.

The reification system builds on the assignment system of Backstage 2, which allows teaching staff to give students tasks and allows students to maintain their own to-do list, which is filled with user created assignments. All types of assignments and other information from Backstage 2 are interesting for evaluation by the reification system e.g.:

- Attending to lecture
- Doing Exercises
- Using the annotation system of Backstage 2
- Performing individual tasks

The Progress Review Framework (PRF) reviews completion of assignment and other participation to the course and rates it to build a score. The Progress Visualisation Framework (PVF) allows students to build a visualisation in a playful way to reify their learning and participation progress. The possibilities to shape the visualisation in part and also a decay of the visualisation depend on the score from the Progress Review Framework. Decay in this context means that parts of the visualisation can age, decline or withered, e.g. a flourishing landscape could become a desert, all based on the participation of the students.

The term “reification” is used in different contexts (Gestalt psychology, Marxism, computer science, fallacy, knowledge representation, linguistics, and statistics) and has different meanings in these contexts but for this work the simple definition is to the point. “Reification is making something real, bringing something into being, or making something concrete.” [104] The idea to make something concrete is the essence of this work. In the case of Progress Review Framework and Progress Visualisation Framework it is the abstract process of learning that should become tangible for the students and the progress of students’ learning should become monitorable for teaching



Figure 3.3: Screenshots of Epic Win form left to right: to-do list, character profile, and progress as path.

staff. Depending on the subject, lectures can be highly abstract, which makes it hard to keep the intrinsic motivation high, if the student cannot see or has no direct benefit from this lecture. A visualisation can help here to make the learning progress concrete and keep the motivation up. With loss aversion, the phenomenon that losing an object is attached more importance by people than gaining the same object (see Section 2.3), a bias exists that can be used to motivate students who have already invested time in their visualisation to further do their assignments, because they do not want their visualisation to decay. More over, a visualisation makes the own performance for the students easier to judge and can work against common biases like the overconfidence effect [100] and illusory superiority [98].

Many of the goals this work aims to achieve with its reification system are similar to goals of gamification. Gamification emphasises on gaming, with a focus on competition and control, where this work focuses on reifying (visualising), playing and self-determination. Therefore, the research on gamification is a solid base to develop a reification system. Many academic and commercial implementations of gamification exist, Google Fit, Epic Win, and Forest App get showcased here because they are particularly interesting to compare the Progress Evaluation Framework against them, or they have unique ideas from whom the Progress Evaluation Framework can adapt.

Google Fit [29] is a health-tracking platform developed by Google which uses gamification to motivate the users. It lets users create activity goals (Figure 3.2, left) and track their training's progress (Figure 3.2, middle and Figure 3.2, right). Its focus is on allowing users to evaluate their training. Google Fit's gamification uses visualisations like pie and line charts to visualise the users' progress, marks days with a tick where the user reached his goals (Figure 3.2, middle) and shows messages like "Nice work! That was your longest run yet." The charts and ticks are similar to a point systems and the messages function as achievements.

Epic Win [72] is a to-do list app (Figure 3.3, right) with gamification and some borrowings from role-playing games. Figure 3.3 (middle) shows the character profile which is similar to role-playing games where the user sees their level and abilities. To visualise the overall progress the character of the user wanders along a path where they can find loot (Figure 3.3, right). With the visualisation of the users' progress in form of a character that improves their abilities, the path they wander and the loot they collect Epic Win reifies the usage of a to-do list. The narration as an adventurer on an epic quest, lent from role-playing games, is what separates Epic Win from other





Figure 3.4: Screenshots of Forest App [26] from left to right: creating a timer to stay focused and not use the smart phone, the forest the user has planted on one particular day, and a selection of different trees the user can plant.

gamification. Giving the gamification a narrative could help the user to understand and relate to the app and its purpose and makes it fun to use.

Forest App [26] is a smart phone app which helps users to stay focused and not get distracted by their smart phones. User plants a virtual tree whenever they want to stay focused, the planting is combined with a timer and when the user does not leave the app until the timer is over the tree will grow; otherwise the tree dies. The progress of the user gets reified in a visualisation of the trees, they start with a plain field in the morning and every completed timer places a tree on it and the failed ones lead to dead trees. Besides a tree, a completed timer gives the user some virtual coins; these coins can be used to buy different looking trees for the virtual forest, but can also be used to plant a real tree (Forest App donates to a NGO). The planting of real trees is a very interesting idea for reification, because it transforms virtual reward, in form of coins, in a real object. This is physical manifestation of the efforts of the user, which they theoretically could visit, see and touch.

This work is going further on the aspect of the visualisation by making it a playful experience in comparison to Google Fit with its common implementation of gamification and Epic Win with its narrative based implementation. The process of manipulating and shaping the visualisation should be fun, so students relate to their visualisations and want to improve and preserve them, which potentially leads to greater involvement in doing assignments and to an enhanced effort in learning. The Progress Review Framework uses basic gamification with points and trophies. The points are only used for internal calculations and are not visible for students. Based on fulfilment of assignments students get points and the ability to place extra graphics on their visualisation. The decay of the visualisation is also based on the points but this mechanic is lent from social games where it is used to sway users into frequent interaction. This work tries to use the mechanic of decay to help unmotivated students to steadily learn for lectures. The Progress Evaluation Framework uses gamification mechanics purposefully when these mechanics help to achieve its goals. Overall the Progress Evaluation Framework is more on the side of playing than gaming and allows the students who use it a fun way to visualise their learning progress.

On the one hand the visualisation of the Progress Evaluation Framework gives students feedback to their learning progress and participation for a lecture and can help them to have a proper



Logic and discrete structures			
Logic			
Propositional logic	First-order predicate logic	Resolution	Natural Deduction
Lecture	Lecture	Lecture	Lecture
Exercise	Exercise	Exercise	Exercise
Discrete structures			
Natural numbers	Whole numbers	Modular arithmetic	Combinatorics
Lecture	Lecture	Lecture	Lecture
Exercise	Exercise	Exercise	Exercise

Figure 3.5: Structure of the the course “logic and discrete structures”.

judgement of their own performance and capability. On the other hand, with the reward system for completed assignments and the decay system that impacts the visualisation negatively the Progress Evaluation Framework can motivate students to invest some extra effort to complete assignments. A third aspect is the option for individual tasks (assignments) that students can create for themselves or can be created by teaching staff for them. This allows to adapt the learning to the individual speed and style and is of equal rank for Progress Evaluation Framework and its review. One aspect that cannot be ignored when giving a system like the Progress Evaluation Framework to students is the possibility that they try to cheat and game it. The Progress Evaluation Framework cannot prevent fraud by the students but it is built so that student who cheat only create self-deception. There is no real profit in cheating because the system is not built for competition between the students, it is built to help the individual student. The next sections describe in detail how the Progress Review Framework and Progress Visualisation Framework work.

### 3.1 Progress Review Framework (PRF)

The Progress Review Framework has two main functions and two main interfaces. The main functions are implementing the data model for the Progress Evaluation Framework and a point system to evaluate the data. The interfaces connect the Progress Review Framework on the one hand to Backstage 2 and on the other hand to the Progress Evaluation Framework. University courses have quite different structures: there are lectures, exercises, practical courses and many more forms of organisation. For example, the course “logic and discrete structures” (Figure 3.5) has two main topics, “logic” and “discrete structures” and each of the main topics has four subtopics. The subtopics are composed of a lecture and an exercise.

This is a common structure for courses in which the lecture conveys the knowledge and the exercise is for practising and deepening. Figure 3.5 shows the hierarchical order of the topics of the course, but it does not necessarily show the chronological order of the topics. Depending on the organisational structure the order of the lectures can be fixed or self-determined by the students. Topics are bigger chunks of the subject of the course to structure subtopics. A subtopic is the content of one lecture. These factors are relevant for the course model of the Progress Review Framework and the next section discusses how to map all this structure to one abstract data model.

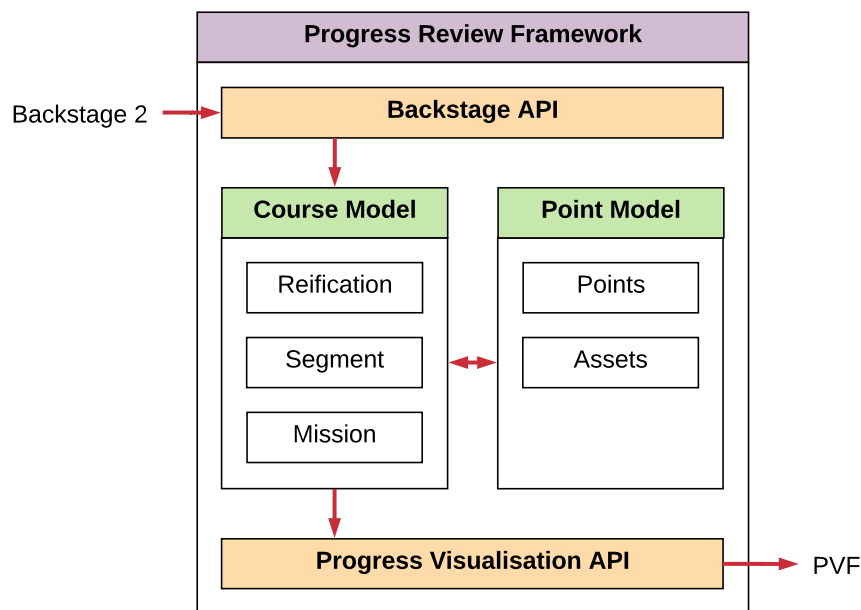


Figure 3.6: Structure of the Progress Review Framework, the arrows show the flow of data.

### 3.1.1 Representation of courses – The course model

The big challenge for the data model of the Progress Review Framework is to map all the different courses and their different organisational structures to one abstract data model. The main goals are to have a simple but also flexible data model that deals with the existing Backstage 2 course data model but can also be extended for other learning platforms. The course model consists of Reification, Segment, and Mission (Figure 3.6). These three build a hierarchy, Reification which comprises Segments and Segments, which comprises Missions. Compared to the example course “logic and discrete structures”, the Reification represents the whole course and the Segment represents a subtopic like “propositional logic” (Figure 3.7). The Mission is not derived directly from the course or its sub elements, but it can be appointed to them.

**Reification** wraps general information on a course, like its title, start and end date, meta information, and knows the Segments of this Reification. For every course that should be reified with the Progress Evaluation Framework one Reification exists.

**Segment** breaks a course down into short and manageable pieces. It holds information like a title, start and end date and meta information. A Segment normally represents a subtopic of a course and also represents the work of about one week. Depending on the length of the term 12 to 18 Segments are combined in a Reification.

**Mission** represents a task for a student and pertains to a Segment. It has a title, a description to clarify the task and notes to allow documentation of the progress of the Mission. The Mission is the central data structure; it can be created by the system, by a student or copied from an existing Mission. Teaching staff can define predetermined Missions, like “attend the lecture” or “do the exercise”, but the students are encouraged to create their own missions to adapt the process of learning to their needs. The Missions meta information link it to its Segment and its owner (the student who has to solve it), indicate if the Mission is a copy of another Mission, and provide the data for the review of the progress like start date, end date, type, and points. The type of Mission can be point, single level, or multi level and describes how many steps are necessary to solve it and

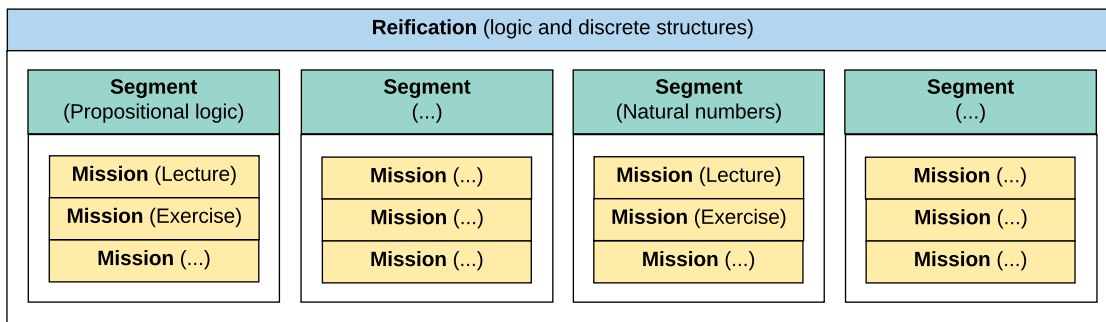


Figure 3.7: Example of the hierarchy of Reification, Segment and Mission for the course “logic and discrete structures”.

influences the visualisation of it. For point and single level Missions the students only have to do one step, for example read a text. Multi level Missions include a number of steps. A good example is the exercise where the student has to download the exercise sheets, do the tasks of the exercise and upload their solution which is then corrected by the teaching staff. When the Progress Review Framework evaluates the accomplishments of the students the mission type is also necessary.

### 3.1.2 Evaluation of accomplishments – The point system

Based on the completion of their Missions students’ accomplishments get evaluated. Missions can be system monitored and self monitored. For a system monitored Mission the Progress Review Framework automatically determines if the Mission was solved successfully or not. For self monitored Missions students by themselves determine if they have solved the Mission successfully. The Progress Review Framework in combination with Backstage 2 can monitor:

- Presence in the lecture (student is logged in to Backstage 2)
- Participation in a lecture quiz
- Usage of the annotation feature for questions or answers
- Reading the lecture slides (as long as students view the slides with Backstage 2)
- Submitting their solution to an exercise

Missions with other subjects need to be monitored by students themselves. On the one hand, self monitored Missions make the Progress Review Framework very flexible, students can monitor every task they want with it; on the other hand, they open the system for fraud, as the Progress Review Framework has no idea if a self monitored Mission was really solved by a student. Because fraud or better self-deception by the students is hard to prevent in such an open system, the Progress Review Framework has no countermeasures against it.

For point and single level Missions students get all the points in one step, when they have finished the Mission. In contrast, for multi point Missions the students get their points step by step for each step of the Mission they finish, which allows an ongoing evaluation and visualisation. Based on the points and the end date the Progress Review Framework calculates if a student has solved a Mission successfully or failed to do so. For point and single level Missions this decision is binary, the Mission is either solved or not and it is also a binary decision for the visualisation in the Progress Visualisation Framework. Multi level Missions get evaluated by their steps, for each step a student solves they get points and in the end a specific amount of points must be achieved to solve the Mission. This multi level concept has to be taken into account for the visualisation in the Progress Visualisation Framework.

Based on the points for the Missions of a Segment the Progress Review Framework calculates if a Segment is finished successfully. Students fail to finish a Segment successfully if they do not reach 50 percent of the total points of the Missions of the Segment. Just as its like in case of the Missions the Progress Visualisation Framework is using this calculations for his visualisation.

### 3.1.3 Interface to Backstage

The Progress Review Framework provides an interface for Backstage 2 which allows creating Reifications with Segments, creating default and individual Missions, and setting Mission points. First, a lecturer decides if they want to use the Progress Evaluation Framework and creates the Reification and default Missions. The Reification will be created automatically from the Backstage 2 course, the default Missions are individual for each lecture and have to be created manually by the teaching staff. Afterwards students can copy these default Missions or create their own individual Missions, which allows Backstage 2 to track learning progress and set points for Missions.

### 3.1.4 Interface to Progress Visualisation Framework

The interface for the Progress Visualisation Framework allows it to read the data model of the Progress Review Framework and create an individual visualisation for each student based on the Reification, its Segments, and the student's Missions.

## 3.2 Progress Visualisation Framework (PVF)

The main function of the Progress Visualisation Framework is to visualise the data from the Progress Review Framework for the students. The visualisation is course based, so every course of a term gets its own visualisation and it has to fulfil a number of functionalities:

- Present the learning progress in easily understandable and interpretable way.
- Allow the presentation of Mission rewards.
- Allow the interaction to be engaging and fun.
- Have a theme that allows decay.
- Have a narrative that is engaging, understandable, and, well-known.

First it will present the learning progress of the students: it should be possible for the students to get an overview of the progress of the whole course, but also for a single lecture. Therefore, the visualisation has easily understandable and interpretable, but also needs enough details to visualise all the different states the learning progress of the students can have. A second goal is to motivate the students to fulfil their Missions, which are based on assignments and tasks and by doing so, motivate them to go to the lecture, do their exercises, and do some extra work for the course. This is done by rewards the students earn for successfully finished Missions, with which they can develop and shape their visualisation in an individual and unique way. The third objective is that the interaction with the visualisation is engaging and fun. It should be a playful experience which leads to a connection between the students and their visualisation because this makes the visualisation to a reification that they can relate to.

The second and the third goal relate to each other and can reinforce each other. If the interaction with the visualisation is fun, students are more likely to shape the visualisation, which can lead to a bonding to their now personal visualisation. In the next step this can lead to a higher motivation of the students to do Missions, because they allow to personalise the visualisation even further. To promote this cycle, the visualisation has two separate options to shape it. One option,

which is available from the beginning (no conditions), that allows to form the basic shape of the visualisation, which shall lead to an initial involvement of the students. Then the second work based option (solving Mission) takes over, where the students can individualise their visualisation further and show their learning results.

This is the point where decay and improvement based on the ideas of loss aversion is used to further influence the students. Rewards for multi level Missions can have various conditions that represent the achieved points, for example the reward can be visualised by a flower that can grow from seedling to a full grown flower and then bloom if the Mission goals were reached or can wither if the students fail. Not only single mission rewards will represent the status of the Missions but also the overall look of the visualisation will be affected, e.g., if the overall theme of the visualisation is “arrange your own garden”, then parts of it can dry out based on the Segments and its Missions status. These effects can help to motivate students to fulfil the tasks of the Missions because they want their visualisation to grow and prosper but even if they failed to solve the Missions in the first place the decayed parts of their visualisation helps them to identify the parts of the course where they need to invest more effort. Important for this feature is to motivate the students and not frustrate them. To achieve this studies are necessary where the operating modes get adjusted.

The visualisations need a way to reach out to the students and win their interest, therefore a metaphor, an image, a story, or a narrative is helpful. EpicWin and in a smaller part Forest App show how a narrative can be attractive for users and gives the gamification a broader purpose to which the user can relate. The visualisation of the Progress Visualisation Framework also should be based on a narrative with a familiar concept and story, which allows students to envision how the visualisation can look like. A first example for an visualisation with a narrative was already mentioned in the section above, with cultivating a garden. Others are building a city, creating a landscape or path, and building a robot. Narratives like these are easy to understand; they relate to experiences that are well known. Probably no student has ever built a real city but they have an idea what it means and what is necessary in an abstract way. Many of the ideas for the visualisation have already been used in computer games as a scenario which can lead to certain expectations from the students. With all these considerations in mind the next sections will elaborate three potential visualisations and also include factors like the possibility and availability to use free graphics and the possibility to use a game engine.

### 3.2.1 Cultivating a garden

In a visualisation that looks like a garden the students have the role of the gardener, who cares for the well-being of the garden. This is a well known game concept from games like the social network game FarmVille [112] or the simulation role-playing game Stardew Valley [3] (Figure 3.8) and can be used for Progress Visualisation Framework in a simplified version.

In the “cultivating a garden” visualisation the whole garden represents the Reification of a course, Segments are represented by patches and Missions are represented by the plants. So for each Mission a student fulfils they can plant a plant on the according patch. For multi level Missions the plants grow, bloom or bear fruit and decay is simulated by the withering of the plants and growing weed in the patches. The possibilities of creation for the students are quite limited in this scenario, they can only decide where in the patch they want to crop the plants. Big design flexibility like rearranging and resizing patches will be at the expense of readability and comparability of the visualisation. So the free options to shape the garden could be only something limited like to change the design of the border of the patches. Cultivating a garden is a strong and easy to understand narrative but the necessity to limit the possibilities of creation for readability and comparability do not make it the first choice for the visualisation for the Progress Visualisation Framework.



Figure 3.8: A screen shot of Stardew Vally [3] (by [4]), showing a farm with its farmland.

### 3.2.2 Robot construction kit

“Be an engineer and build a robot with a robot construction kit.” This is the topic for a visualisation where the students build a custom robot to visualise their learning progress and success. It is inspired by Mix-and-Match or Flip-Flap books, which allow to combine different body parts, most of the time from animals, to a new animal (Figure 3.9, left) and dress-up books, where paper dolls and dresses get cut out and then combined (Figure 3.9, right).

The robot construction kit uses the mechanics from the books to allow the students to build their own individual robot avatar for a course. Similar to the garden a robot represents the Reification of one course; here the Segments get illustrated by different body parts of the robot and Missions are represented by adornments and accessories. To allow the building of a unique robot for every body part several designs are necessary and also different partitioning of the robot body are necessary for courses with different number of lectures. The adornments and accessories could be things like displays, lights, antennas, armour, bolts, and knobs and many more. One difficulty is to find adornments to represent the multi level Missions. A possible solution is to give the adornments various expansion stages, for example, an antenna can be expanded from a simple rod antenna to a satellite dish. Decay could be visualised with abrasion and rust on the according parts or a reduced energy supply in form of battery level for the according parts. The robot construction kit could be built with 2D or 3D graphics. 3D graphics have the potentials to bring the idea of reification even a step further, allowing students to 3D print their robots at the end of the term. The robot construction kit is a strong narrative and would be a good visualisation but the need to design custom graphics is raising the amount of work to realise it too much for this work.

### 3.2.3 Creating a landscape

The students create their own landscape as visualisation for their learning success. This landscape represents a path the students have to cover during this course. The narrative for this visualisation is inspired by the history of creation in the bible where God creates the world or all the cultivated landscapes mankind has created during the centuries. The students’ landscape goes from a dull and empty wasteland to a picturesque and blooming scenery. Several video games have a similar functionality in form of a level editor, where the players can create their own levels and even whole games are based on this idea like Super Mario Maker [67].





Figure 3.9: On the left the book cover from *Croc-gu-phant* [1], which illustrate what a Mix-and-Match book is. On the right the book cover from *Cut-Out Paper Dolls* [95], which illustrate what a dress-up book is.

Every course Reification gets visualised by its own landscape and the landscape gets split into parts to represent the Segments of the reification. The Missions of the Segments get visualised with objects in the landscape, like bushes, trees, rocks, flowers, clouds and so on. The form of the landscape can be shaped by the students at any time. This implements the free option of the visualisation and with every fulfilled Mission the students can decorate it more with objects until it is their unique landscape. For single level Missions the students receive small objects and for multi level Missions big objects. Trees are a good visualisation for multi level Missions, as they can grow, die and bear fruit. The decay is simulated with dried out parts of the landscape which become desert and objects like trees and plants which die. Originating from video games there are various possibilities to illustrate the landscape, the visualisation could use 2D or 3D graphics and also different perspectives or points of view [106]. Based on the availability of graphics and game engines, the Progress Visualisation Framework uses 2D and side-scrolling for its visualisation of creating a landscape. This leads to a visualisation that looks like a 2D platform game [102]. This visualisation has a view advantage; as mentioned earlier, a lot of free graphics exist for 2D platform games and the structure of these graphics with small tiles from which a level gets put together is also ideal to build a landscape. The linear structure of platform game levels makes it easy to map the different amount of Segments a course Reification can have.

Figure 3.10 shows a sketch of a landscape visualisation with 15 Segments. The upper part shows a possible default landscape, which is flat and monotonous, but in the lower part the landscape already has been partly edited and customised, which quickly leads to a unique landscape. Figure 3.10 shows Segments that are composed from four landscape tiles to give the students space to display their creativity in shaping the landscape and also have room for objects; in contrast, the left Figure 3.11 shows Segments that are three tiles wide. How many tiles are ideal for the representation of a Segment has to be tested. The left part of Figure 3.11 shows how editing the landscape works, where the users can only edit one Segment at a time and has some sort of pallet to select ground tiles to build the landscape or add objects. The right part of Figure 3.11 shows an edited Segment with a tree that represents a multi level Mission and a mushroom hat represents a

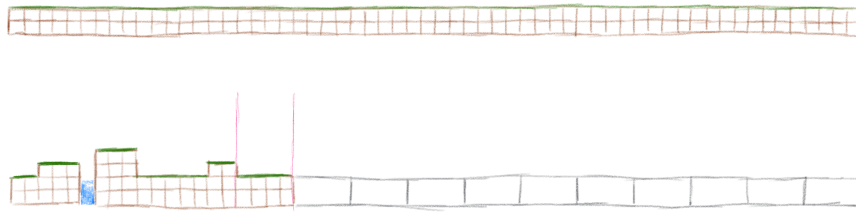


Figure 3.10: View of the whole landscape, the upper part shows a possible default landscape and the lower part a partly edited landscape which also highlights a segment.

single level Mission.

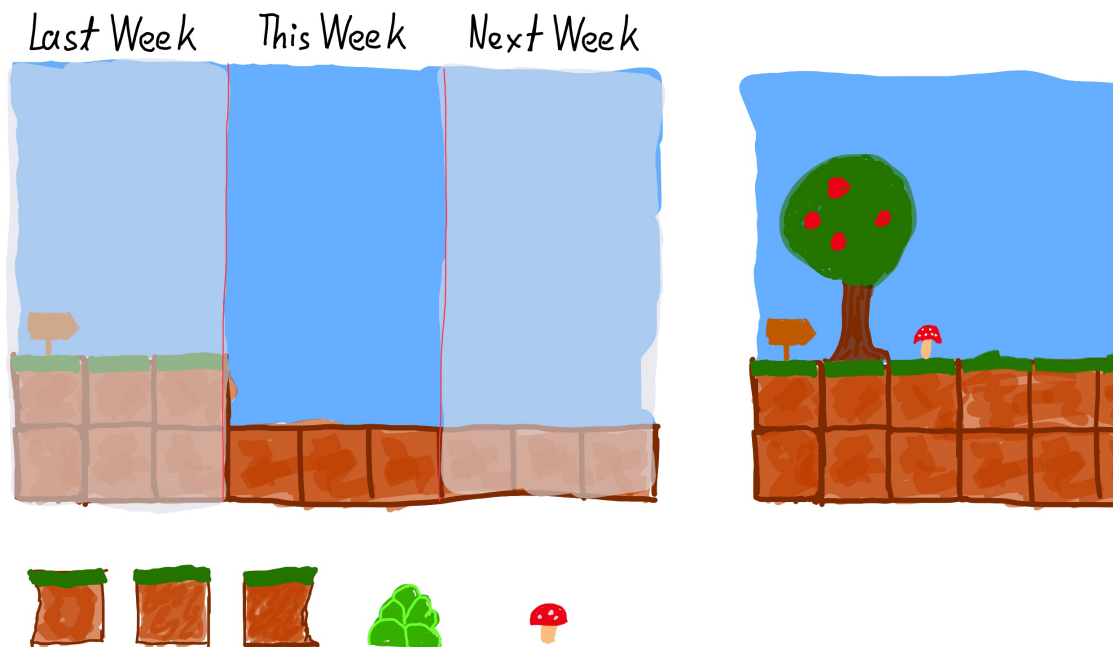


Figure 3.11: On the left a view of the landscape, that highlights the Segments and shows the edit function which allows to arrange the landscape with ground elements and place rewards for Missions. On the right a simple landscape with a tree that represents a multi level Mission and a mushroom that represents a single level Mission.



## 4 Development

This section describes the realization of the concept for Progress Evaluation Framework as a concrete implementation for Backstage 2. The Progress Evaluation Framework will be a part of Backstage 2 and is based on the same technology as Backstage 2. The following section explains the software stack, the realization of the concept into software, and the concrete implementation.

### 4.1 Software Stack

Backstage 2 is a web application [107] with a client and server component which uses RethinkDB, Express, React, and Node.js as software stack. JavaScript is used as the only programming language for both client-side and the server-side development. Backstage 2 extends its development stack with additional frameworks and libraries and the implementation of the Progress Evaluation Framework further adds the game framework Phaser CE<sup>1</sup>. The important components of the stack are introduced here briefly:

**RethinkDB** is a distributed document-oriented database with a focus on realtime web applications [73]. The backend of Backstage 2 uses RethinkDB to store its data.

**Express** is a server framework, which extends Node.js for building web applications and APIs [27]. With its help a REST API for the Progress Evaluation Framework is built.

**React** is a JavaScript library for building user interfaces [41]. The implementation of the Progress Evaluation Framework uses React for all parts of the user interface which are not rendered by Phaser CE (see below).

**Node.js** is a JavaScript run-time environment that allows to implement server-side applications in JavaScript [28]. The backend of Backstage 2 is implemented with Node.js in combination with Express and RethinkDB and just the same is true for the backend of the Progress Review Framework (see Sections 4.2.2, 4.2.3, and 4.2.4) and the Progress Visualisation Framework (see Sections 4.2.2, and 4.2.4).

**Phaser CE** is a game framework for making HTML 5 games [57]. The implementation of the Progress Visualisation Framework uses it to present its visualisation. It is a full-featured 2D game framework the Progress Visualisation Framework (see Section 3.2) only uses a fraction of.

### 4.2 Form Concept to Implementation

The development stack was the topic of the last section but the fact that the Progress Evaluation Framework will be part of Backstage 2 leads to more boundary conditions. Backstage 2 has a client-server architecture what is typical for web applications. The implementation of the Progress Evaluation Framework adopts this architecture and uses already existing components and structures from Backstage 2. On client side, the main component is a visualisation editor that gets its own view in Backstage 2. It combines the visualisation and a user interface to interact with this visualisation. This editor is complemented by a business logic which handles the server communication and does all necessary client side computation. On server side, the implementation consists of a REST API to allow the communication with the client, business logic to validate user input and storage logic to store the data in the data base.

---

<sup>1</sup><https://github.com/photonstorm/phaser-ce>

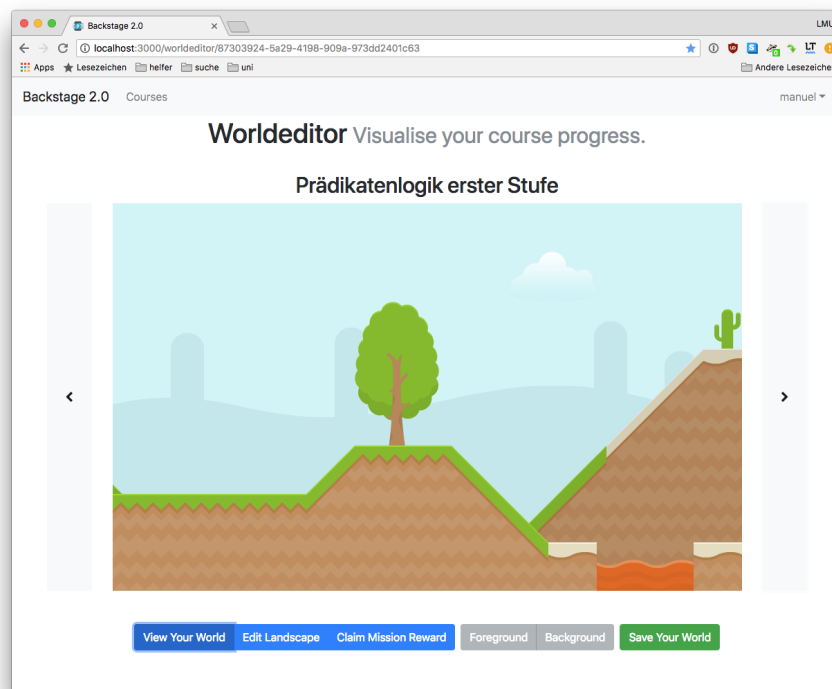


Figure 4.1: User interface of the visualisation editor, with the visualisation in the center and the other user interface elements around it.

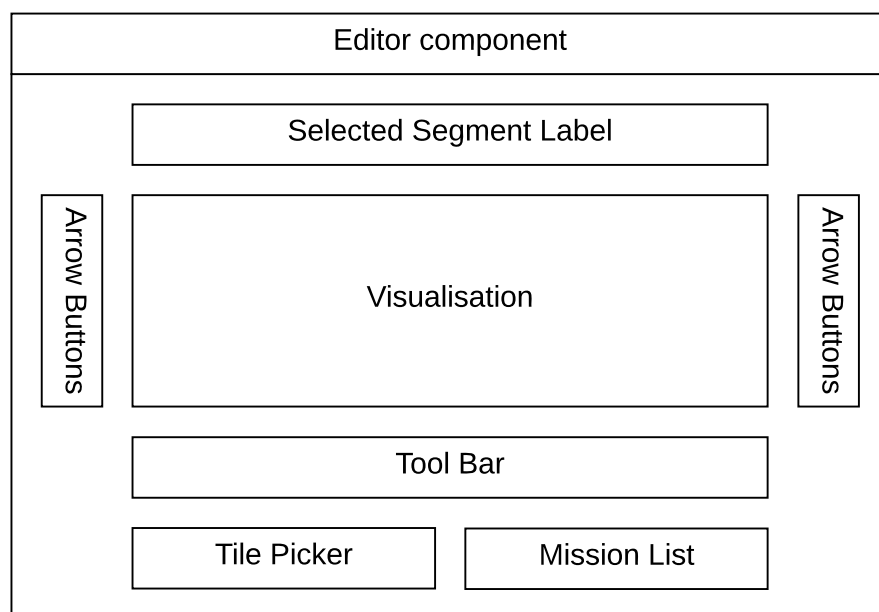


Figure 4.2: Structure of the main editor component (WorldEditor) with its nested components.

### 4.2.1 User Interface

The user interface has two parts: one for teaching staff to administrate the course reification and one for students to interact with the visualisation. This work focuses on the user interface for the students. Figure 4.1 shows the visualisation editor, which allows students to view their visualisation, shape the landscape, and claim Mission rewards. The visualisation is placed dominantly in the center, and the other UI elements are positioned around it. Above the visualisation is a label that shows the selected Segment with the title of Segment. The arrow buttons left and right of the visualisation allow users to wander through their landscape (by selecting the previous or next Segment) and below the visualisation is a tool bar which allows to change the mode of the editor and save the edited landscape.

The UI elements are implemented as React components to enclose their functionality, which helps to write compact and manageable junks of code and allows the reuse of UI elements. The visualisation editor consists of one main editor component, that encapsulates all the other components (Figure 4.2) and the variables and states that are necessary. It manages if sub components and propagates state changes from one sub component to the others.

This visualisation is rendered by the Phaser CE game framework. Using a fully fledged game framework has advantages and disadvantages.

Advantages are:

- Game frameworks have defined formats to store level data and other data structures.
- They can render levels with multiple layers (foreground, background and parallax scrolling [101]).
- They have the capability to work with sprite sheets and sprites.
- They have the capability to work with animations.

Disadvantages are:

- Game frameworks are made to build and run games, but not for editing the level while they are running.
- The size of the application can grow significantly with the use of a game framework

In the end for this work the advantages outweigh the disadvantages and Phaser CE was selected, because of the maturity of the project and the license (MIT License). The graphics in Figure 4.3 originate from Kenny.nl [51] and for simpler handling the individual graphics are combined to three sprite sheets [105].

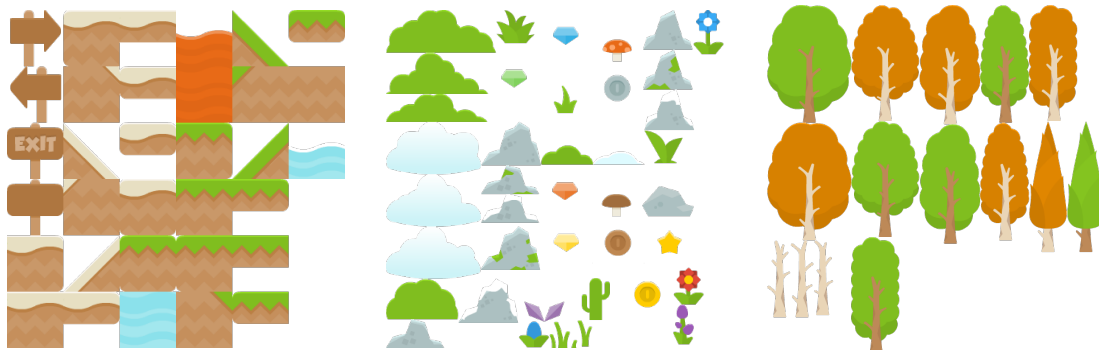


Figure 4.3: The sprite sheets used for the “creating landscape” visualisation created with graphics from Kenny.nl [51]. The graphics are licensed under creative commons (CC0 1.0). The “Platformer Pack Redux” [52] is used for landscape tiles and single level rewards and the “Foliage Pack” [50] is used for both multi level rewards and single level rewards.

**Arrow Buttons and Selected Segment Label** as seen in Figure 4.1 enable users to navigate through their landscape. The arrow to the left selects the previous Segment and the arrow to the right selects the next Segment of the course Reification. The label above the visualisations shows the associated title of the selected Segment, which is shown in the center of the visualisation. The button for the arrows is implemented as component, but the logic for selecting the Segment is implemented in the editor component together with the logic for the label.

**Tool Bar** is the central UI element. It allows users to select the different modes of the visualisation, depending on the mode to choose the visible layer, and save their changes to the visualisation. Figure 4.4 shows the tool bar with its six buttons in three groups. The first group consists of three buttons and lets users select the modes of the visualisation. Every mode allows the user to perform a different interaction with the visualisation and shows user interface elements associated with this interaction. Like the second group which is visible in the modes “Edit Landscape”, and “Claim Mission Reward”, it consists of two buttons and lets users pick the layer of the visualisation they want to edit. The last group only consists of one button that lets user save their changes to the visualisation.

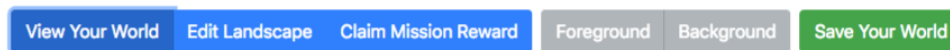


Figure 4.4: The tool bar is the central UI element. It enables to specify the modes of the visualisation (three blue buttons on the left), to select the visibility of the layers of the visualisation (two grey buttons in the middle), and to save changes to the visualisation (green button on the right).

The tool bar is implemented as a component consisting of sub components. The button groups for modes and layer selection use a component that imitates the behaviour of radio buttons, where only one button can be selected. Selecting mode or changing layer, changes states in the main editor component and then changes the condition of other components, like the visualisation. The save button calls the server side API to store the visualisation.

**Tile Picker** extends the user interface in the “Edit Landscape” mode, with an interface for selecting tiles. Tiles are images which function as building blocks for the landscape and with them users shape the landscape of their visualisation. Figure 4.5 shows the tile picker. Every tile is shown in its original size and can be selected to be used in the visualisation. Again a left and a right arrow button are available to skim through the tiles. Additionally, an eraser button exists to delete tiles from the visualisation.

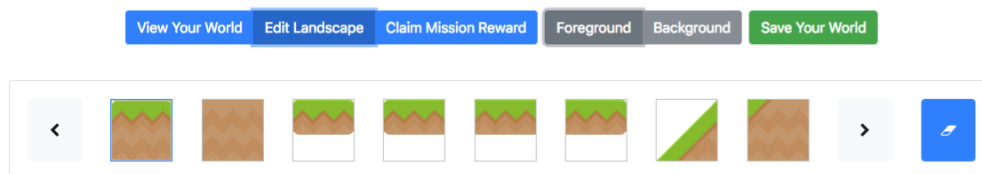


Figure 4.5: The tile picker shows the available tiles to shape the landscape of the visualisation.

The tile picker is implemented with sub components. For the arrow buttons, the same component is used as for the arrow button to select the Segment. The eraser button is a simple component, that adds logic to an HTML button element. The most complex sub component in the tile picker is the component that visualises the tiles, because it uses the same sprite sheet as the game framework and not individual graphics. This reduces the calls to the server when loading the site.

**Mission List** extends the user interface in the “Claim Mission Reward” mode, with an interface for choosing rewards for Missions. Figure 4.6 shows the mission list. It lists the students’ Missions for a Segment with the title, the description, and a button to claim a reward or an image of the claimed reward. For example the three Missions in Figure 4.6 show different states:

- The first Mission is successfully finished, but no reward has been claimed yet,
- the second Mission is a single level Mission, where a user is claiming a reward, and
- the third Mission is a multi level Mission, where a reward already has been claimed in form of a tree.

Claiming a reward by clicking the finished button shows a tile picker with all the tiles which can be placed as a reward in the visualisation and fill the landscape with life.

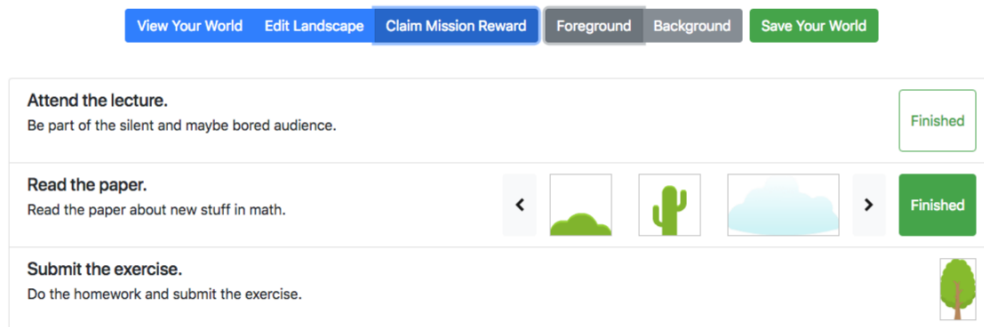


Figure 4.6: The Mission list shows the Missions for a segment and lets users claim rewards for this missions.

The Mission list component is a list of mission list item components. The mission list item components displays Mission properties and the mission tile picker. The mission tile picker has three states which can be seen in Figure 4.6 on the right:

- The first state (top button in the Figure with the title “Finished”) indicates to users that they have solved their Mission and now can choose a reward.
- The second mission list item shows the second state of the mission tile picker where it is displaying a tile picker and a button and users can pick their reward.
- The tree shows the third state, here users already have claimed a reward and instead of a button the image of the reward is shown.

The mission tile picker is the component with the most logic in this hierarchy of components.

**Visualisation** is the central user interface element of the visualisation editor. It shows the visualisation students have built and lets them edit it. Depending on the mode selected with the tool bar the visualisation looks different and allows different interaction. The three different interaction modes are “View Your World”, “Edit Landscape”, and “Claim Mission Reward”.

Figure 4.7 shows the visualisation in the mode “View Your World”, where users can look at the landscape they have already built. The possible interaction is to wander through the landscape with the arrow buttons left and right beside the visualisation. In this mode parallax scrolling is used to give the landscape a greater feeling of depth. Parallax scrolling is a 2D animation technique where background images move slower than foreground images across the view, what creates a greater feeling of depth and increases the immersion [101]. For example, the big hill on the right in Figure 4.7 is rendered on a background layer and therefore is shifted to the right by parallax scrolling.

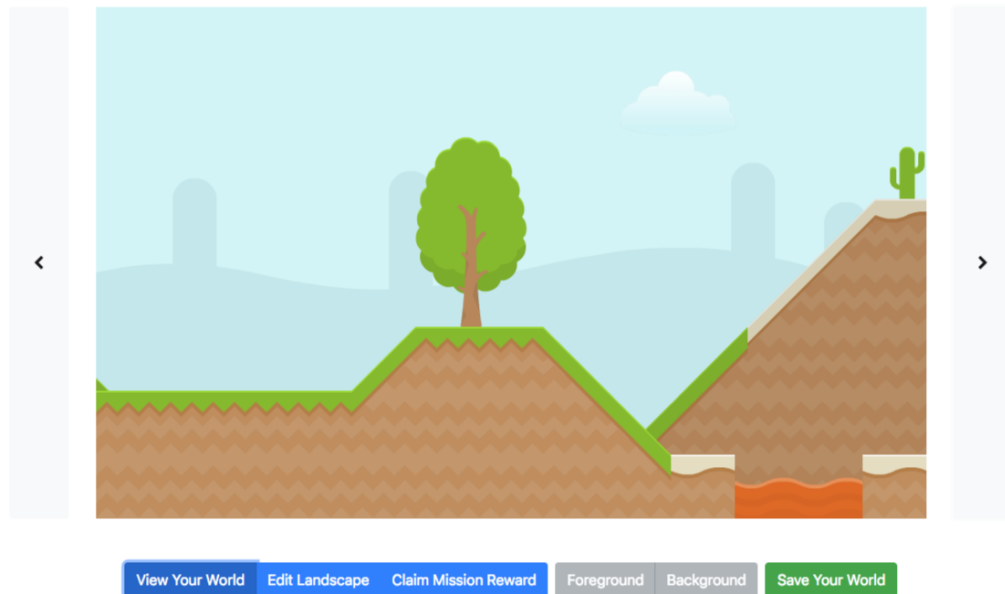


Figure 4.7: Visualisation and tool bar. The tool bar highlights the currently selected modes (View Your World), that allows wandering through the landscape.

When the user switches to the second mode “Edit Landscape” the visualisation changes its presentation. Figure 4.8 and Figure 4.9 both show the mode “Edit Landscape” with the layers button group and the tile picker visible. In Figure 4.8 “Foreground” is selected by the layers button group, this allows the user to edit the foreground and the background gets rendered translucent to lower possible distraction. In Figure 4.9 “Background” is selected and the rendering is vice versa. In both screenshots a big red outline is visible in the center of the visualisation to highlight the currently selected Segment. This is the area that users can edit. In the upper left corner of the red outline a mouse pointer gets rendered in form of a tile with a green outline. With this mouse pointer, users can edit the landscape; it allows to place, replace, and erase tiles on the selected layer. With the tile picker users select the tile graphic they want.

The third mode “Claim Mission Reward” works in almost the same manner as the second mode, but instead of shaping the landscape with tiles, users can place objects in the landscape which represent rewards for their solved Missions. Figure 4.10 and Figure 4.11 both show the

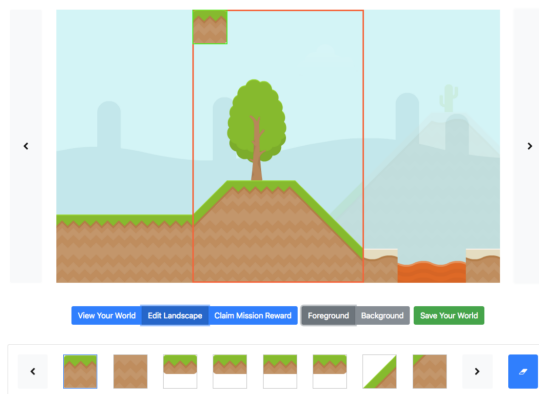


Figure 4.8: The visualisation in mode “Edit Landscape” with selected foreground.

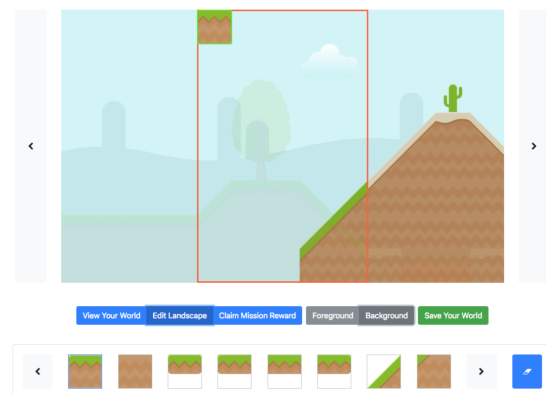


Figure 4.9: The visualisation in mode “Edit Landscape” with selected background.

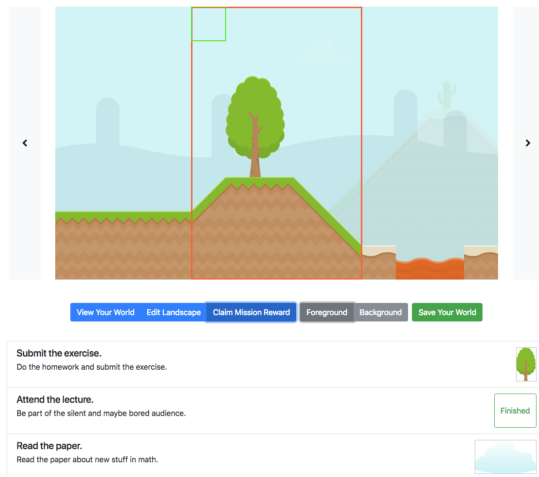


Figure 4.10: The visualisation in mode “Claim Mission Reward” with selected foreground.

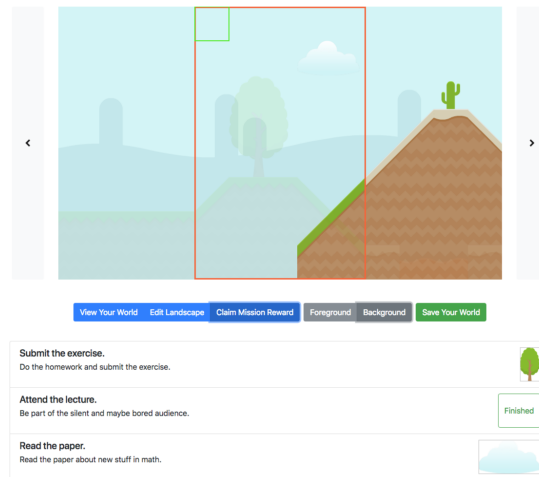


Figure 4.11: The visualisation in mode “Claim Mission Reward” with selected background.

mode “Claim Mission Reward” with the layers button group and the Mission list visible. In Figure 4.10 “Foreground” is selected by the layers button group, this allows the user to edit the foreground and the background gets rendered translucent to lower possible distraction. In Figure 4.11 “Background” is selected and the rendering is vice versa. In both screenshots a big red outline is visible in the center of the visualisation to highlight the currently selected Segment. This is the area where users could place further rewards, supplementing the already placed tree. In the upper left corner of the red outline a mouse pointer gets rendered in form of a green outline. With this mouse pointer, users can place, or replace reward objects.

All interactions with Phaser CE are wrapped in a React component, which is the most complex component of the implementation. To run a Phaser CE based game on a website a new game object has to be initialized and bound to an HTML element and then a custom state object is added. The state object implements the custom game logic by implementing callback functions which get called from Phaser CE’s game loop. To run the visualisation four callback functions need to be implemented “init”, “preload”, “create”, and “update”. The “init” function is used to prepare a set of variables and objects, among other things the reification and visualisation data objects get passed to the state object. The “preload” function loads the graphics, that are used to render the visualisation and loads the level itself. The “create” function creates all the necessary objects to render the visualisation and adds them to the game object:

- A tilemap object, with all its layers,
- a tilesprite object for the background image, and
- group objects with sprite objects for the rewards.

The last callback function is the “update” function, which is called in every cycle of the game loop, while “init”, “preload”, and “create” are only called once during the first run of the game loop. The “update” function repositions the camera if users interact with the user interface. The visualisation has five layers, from front to back: foreground rewards, foreground landscape, background rewards, background landscape, and background image. Users can edit four of these layers, the two reward layers and the two landscape layers. In the presentation to the user foreground rewards and foreground landscape, and background rewards and background landscape are combined.



---

	Internal format	Tiled Map Editor format
1		
2	[	[
3	[0, 0, 0, 0, 0],	0, 0, 0, 0, 0,
4	[0, 0, 0, 0, 0],	0, 0, 0, 0, 0,
5	[0, 0, 0, 0, 0],	0, 0, 0, 0, 0,
6	[0, 0, 0, 0, 0],	0, 0, 0, 0, 0,
7	[0, 0, 0, 0, 0],	0, 0, 0, 0, 0,
8	[0, 0, 0, 0, 0],	0, 0, 0, 0, 0,
9	[22, 22, 22, 22, 22],	22, 22, 22, 22, 22,
10	[12, 12, 12, 12, 12]	12, 12, 12, 12, 12
11	]	]

---

Listing 1: Internal and Tiled Map Editor format to store the shape of the landscape.

### 4.2.2 Business Logic

The business logic is spread between client and server. On client side, all the transformation of internal data structures to Phaser CE compatible data structures is done as well as the calculations if parts of the visualisation need to show a decay. The server side handles CRUD operations that determines and validates data before they get stored in the database.

Transforming the internal data format of the visualisation on client side into a Phaser CE compatible format is done in two steps. First, the internal data format gets transformed to the Tiled Map Editor<sup>2</sup> JSON Map Format [109] (see Listing 1 and Appendix A). This JSON map contains the shape of the foreground and background landscape. The second step is done by the component that wraps Phaser CE, which transforms reward data, stored in the internal data format of the visualisation, to sprite objects which can be rendered by Phaser CE. Both steps use decay information which are calculated by the model objects on client side.

On server side, data access object (DAO) classes handle the create, read, update, and delete (CRUD) operations. Beside the CRUD operations the data access objects provide methods to derive instances of Reification, Segment, Mission, and visualisation data structure from other Backstage 2 objects: Reification and Segment data structure get created from a given Backstage 2 course material object, default Mission can be predefined by the teaching staff and then copied by the students, and a start visualisation for the Reification, in form of a flat grass land, gets generated on the first load of the visualisation. Furthermore when students save changes to their visualisation, the visualisation data structure does not get updated direct, but first copied and then updated, to generate a timeline for the visualisation, which can be used in a future feature to show students time lapse animation of their learning progress.

### 4.2.3 Input and Output

The Progress Evaluation Framework provides various interfaces; this thesis has already discussed the user interface which is the most visible and most complex one, but the REST API and other internal interfaces are important as well. The Rest API enables the client-server communication and provides five methods to call. Those methods allow the client to load and save Reification and Visualisation objects. Internally on the server side, the DAO objects and their methods work as an interface to other parts of Backstage 2.

---

<sup>2</sup><https://www.mapeditor.org/index.html>



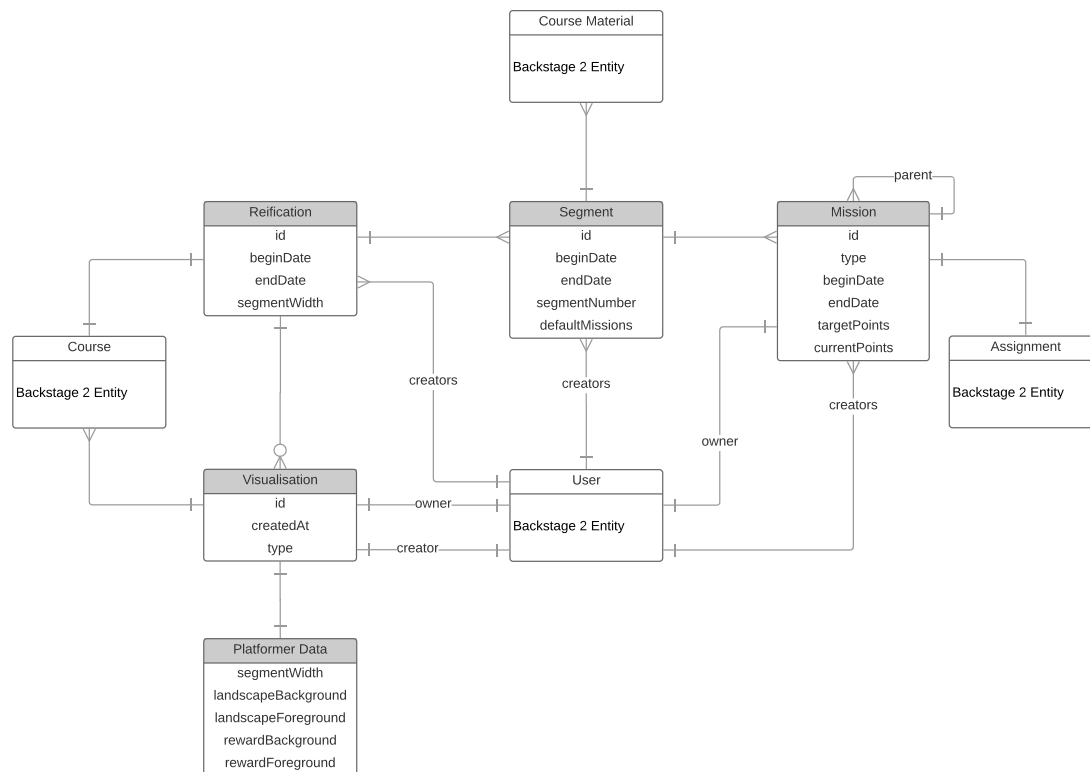


Figure 4.12: The data model of the Progress Evaluation Framework, with the relations between its entities and the relations to Backstage 2 entities.

#### 4.2.4 Data Model

The data model is used to exchange data between the different components of the Progress Evaluation Framework. On the client side it is explicitly implemented as classes and on server side it gets realised implicitly by the JSON objects which get stored in the database. Every JSON object has a unique identifier most of the times created by the database when an object gets stored for the first time. Relations between objects are implemented by storing the unique identifier of another object as property. The DAO objects allow loading objects as unfolded objects; this means the related objects get loaded from the database and replace the unique identifier in the parent object. Figure 4.12 shows the entity-relationship model of the Progress Evaluation Framework and the appendix contains listings of JSON code for all entities of the data model.

**Reification** is the root element of the Progress Evaluation Framework data model. It is the entry point to establish the complete data structure of Reification, Segments, Missions, and Visualisation for a course. The Reification gathers the base information, on the Backstage 2 course it derives from, the users who have created and edited it, the period of time this Reification reifies (beginDate, and endDate), and Segments which belong to this Reification.

**Segment** maps a lecture of a course to the Progress Evaluation Framework data model. It stores the information of one Backstage 2 course material object, the users who have created and edited it, the period of time and order this Segments reifies, and default Missions that belong to this Segment.

**Mission** maps an assignment to the Progress Evaluation Framework data model. It stores the information of one Backstage 2 assignment object, the users who have created it, the user who owns it, the period of time in which this Mission has to be solved, the received points, and additional information. The parent gets set when the Mission is a copy of an already existing Mission from another user.

**Visualisation** stores all information which is necessary to render the Visualisation with Phaser CE. The object has properties that are independent from the type of the Visualisation and properties that depend on the type of the Visualisation. All the depending properties get combined in the sub object “data”. Independent properties are references to other objects, a timestamp when the object was created, and the additional information.

**Platformer Data** for “creating a landscape” as has a simple structure. It has five properties: “segmentWidth”, “landscapeBackground”, “landscapeForeground”, “rewardBackground”, and “rewardForeground”. The “segmentWidth” stores the number that determines the number of tiles a Segment in the Visualisation has. The “landscapeBackground” and “landscapeForeground” are a structure of nested arrays, where in the root array is one array for each Segment in the visualisation and in each Segment array are eight arrays which represent a row of tiles. A row of tiles array contains as much numbers as the “segmentWidth” determines, each number represents on tile in the visualisation. This structure of nested arrays was chosen because it is easy to edit as it reproduces the structure of a Reification and its Segments and it can easily be converted into the Tiled Map Editor JSON map format. The “rewardBackground”, and “rewardForeground” are dictionaries realised with JavaScript objects. Each dictionary record consists of a unique identifier as key and an object that contains all information necessary to render the reward.

**ShoeBox sprite sheet JSON format** is the meta data format, that ShoeBox<sup>3</sup> generates when creating a sprite sheet. It contains detailed information to each graphic in a sprite sheet, for the Progress Visualisation Framework this JSON files get some extensions. The JSON format contains an array with a frame object for each graphic, this frame object contains information to the name, the size, and the position of the graphic and Progress Visualisation Framework extensions.

**Tiled Map Editor JSON map format** is the format, that Tiled Map Editor uses to store maps. It is one of the supported map formats of Phaser CE, so the Progress Visualisation Framework uses it as intermediate format. When loading a visualisation into Phaser CE the Visualisation object gets transformed into this format.

## 4.3 Implementation

This section describes the implementation of the Progress Evaluation Framework and the structure and hierarchy of the React components and the JavaScript classes. The description is from top to bottom, it starts with the main components and classes and breaks them down into their sub-components and sub-classes.

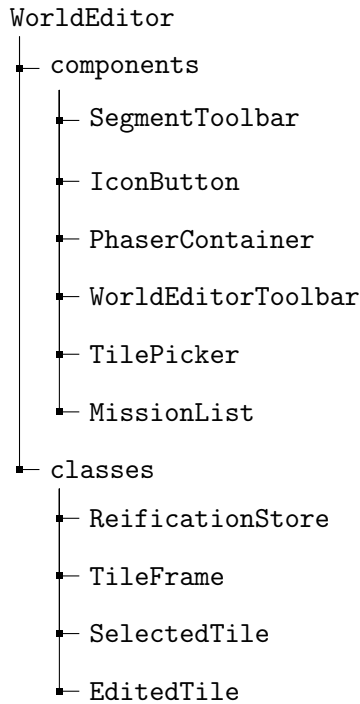
### 4.3.1 Client side

On the client side there are the React components which implement the user interface, the state class for Phaser CE, and the model and store classes.

---

<sup>3</sup><http://renderhjs.net/shoebox/>

**WorldEditor** is main React component, that renders the view with which the user interacts. Listing 2 shows the components and classes the WorldEditor uses. Its main purpose is to manage the state for the other components.

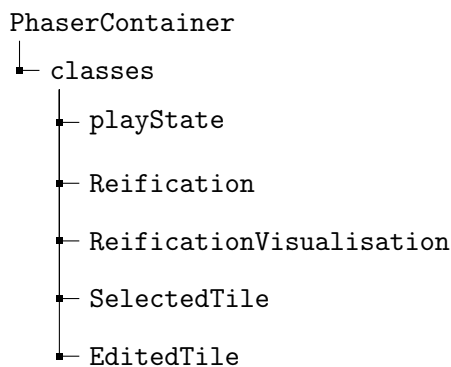


Listing 2: WorldEditor component with its sub-components and sub-classes.

**SegmentToolbar** is a React component, that renders the Segment title as headline above the visualisation.

**IconButton** is a React component, that renders a button with a icon.

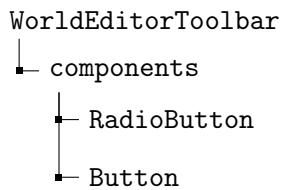
**PhaserContainer** is a React component, that wraps Phaser CE into a React component. Listing 3 shows the classes the PhaserContainer uses.



Listing 3: PhaserContainer component with its sub-class.

**playState** is a class, that implements the necessary callback methods for the usage of Phaser CE.

**WorldEditorToolbar** is a React component, that creates two groups of `RadioButton` components and a `Button` component (Listing 4).

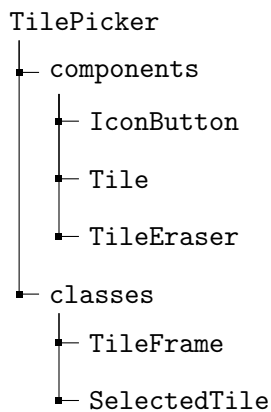


Listing 4: `WorldEditorToolbar` component with its sub-components.

**RadioButton** is a React component, that renders a button with a title and imitates the radio button behaviour where only one button of a group of buttons can be active.

**Button** is a React component, that renders a button with a title.

**TilePicker** is a React component, that creates a list of `Tile` components, with `IconButton` components to skip through this list (Listing 5).

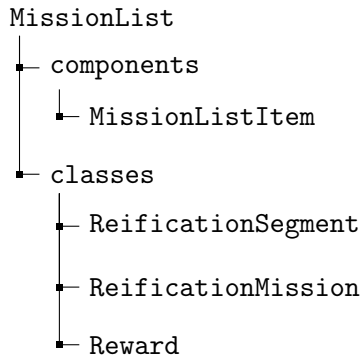


Listing 5: `TilePicker` component with its sub-components and sub-classes.

**Tile** is a React component, that renders one tile of a sprite sheet or a button. Normally a cutout of a sprite sheet gets rendered by the `Tile` component, but if it is initialised with special parameters instead a button gets rendered, this function is used by the `MissionTilePicker` component.

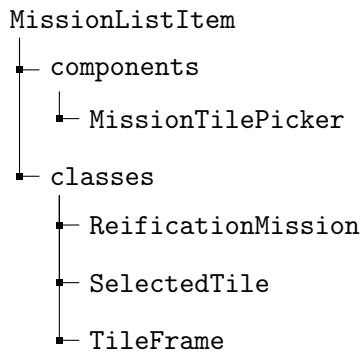
**TileEraser** is a React component, that renders a button with a eraser icon.

**MissionList** is a React component, that creates a list of MissionListItem components. For each Mission of the Segment a MissionListItem component gets rendered (Listing 6).



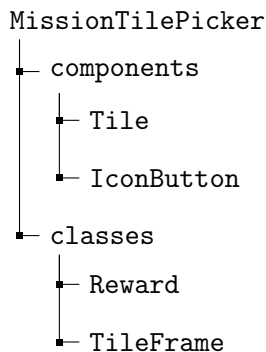
Listing 6: MissionList component with its sub-components and sub-classes.

**MissionListItem** is a React component, that renders the informations of a Mission and creates a MissionTilePicker component (Listing 7).



Listing 7: MissionListItem component with its sub-components and sub-classes.

**MissionTilePicker** is a React component, creates a list of Tile components, with IconButton components to skip through this list and a Tile component to toggle the visibility of this list (Listing 8).



Listing 8: MissionTilePicker component with its sub-components and sub-classes.

**reificationStore** is a class, that handles the communication with the server and buffers loaded objects (Listing 9).

```

reificationStore
├── classes
│   ├── Reification
│   ├── ReificationSegment
│   └── ReificationVisualisation

```

Listing 9: reificationStore class with its sub-classes.

**Reification** is a class, that implements the data model entity Reification on the client side (Listing 10).

```

Reification
├── classes
│   └── ReificationSegment

```

Listing 10: Reification with its sub-classes.

**ReificationSegment** is a class, that implements the data model entity Segment on the client side (Listing 11).

```

ReificationSegment
├── classes
│   └── ReificationMission

```

Listing 11: ReificationSegment with its sub-classes.

**ReificationMission** is a class, that implements the data model entity Mission on the client side.

**ReificationVisualisation** is a class, that implements the data model entity Visualisation on the client side (Listing 12).

```

ReificationVisualisation
├── classes
│   ├── Reification
│   ├── EditedTile
│   └── Reward

```

Listing 12: ReificationVisualisation class with its sub-components and sub-classes.

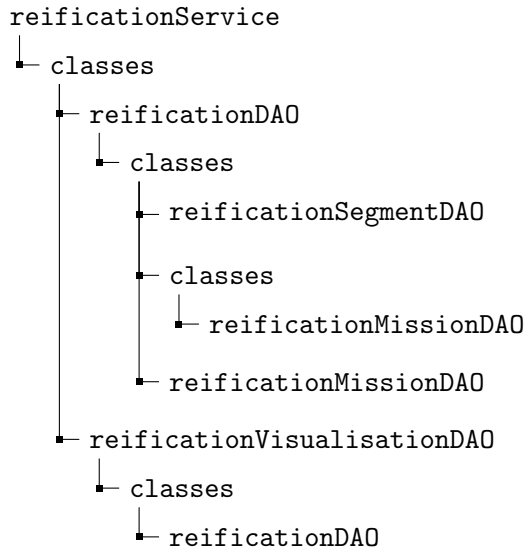
**EditedTile** is a class, that implements a data structure to store changes that users make to their visualisation.

**Reward** is a class, that implements a data structure to store changes that users make to their visualisation.

### 4.3.2 Server side

On the server side there is the service class and the DAO classes.

**reificationService** is a class, that implements the REST service for the client-server communication (Listing 13).



Listing 13: reificationService class with and sub-classes.

**reificationDAO** is a class, that implements the CRUD methods for the data model entity Reification.

**reificationSegmentDAO** is a class, that implements the CRUD methods for the data model entity Segment.

**reificationMissionDAO** is a class, that implements the CRUD methods for the data model entity Mission.

**reificationVisualisationDAO** is a class, that implements the CRUD methods for the data model entity Visualisation.

This section discussed the implementation of the Progress Review Framework as part of Backstage 2. In its two major sub sections it first outlined how to realize the Progress Review Framework concept in software and then explained the concrete implementation that is part of this work. In the next section this implementation will be tested for its usability.





### 5 Evaluation

Part of this thesis is a usability study for the implemented “creating a landscape” visualisation; ideas for a behavioural study will be have discussed at the end of this section. The usability study is based on the recommendation of the usability.gov<sup>4</sup> website. It was performed with the “Concurrent Think Aloud” technique, that means the participants are encouraged to vocalize their thoughts while doing the tasks of the study [92]. Based on the recommendation of Nielsen the study was performed with 6 participants [66]. The following evaluation is based on the “Report Template: Usability Test” from usability.gov [91].

#### 5.1 Introduction

The world editor with the “creating a landscape” visualisation is the interface for students to use the Progress Evaluation Framework for Backstage 2. It provides a central repository for students to reify, visualise, and monitor their learning progress.

A usability test is intended to determine the extent an interface facilitates a user’s ability to complete routine tasks. Typically the test is conducted with a group of potential users either in a usability lab, remotely (using e-meeting software and telephone connection) or on-site with portable equipment. Users are asked to complete a series of routine tasks. Sessions are recorded and analyzed to identify potential areas for improvement to the web site.

As part of this thesis an onsite usability test was conducted using a development version of Backstage 2 located on the test administrator’s laptop. A voice recorder captured the participants vocalized thoughts and the test administrator took notes on their behaviour. The test administrator was present in the testing room. The session captured each participant’s task completion rates, comments, questions, and feedback.

#### 5.2 Executive Summary

The test administrator conducted an onsite usability test at his flat during March 24<sup>th</sup> and April 7<sup>th</sup>, 2018. The purpose of the test was to assess the usability of the web interface design and to find out if the interaction with the visualisation is joyful and fun.

Four students from different degree programs and two former students participated in the test. Each individual session lasted approximately half an hour. All participants did the same test scenarios.

In general all participants found the world editor to be clear, straightforward, and 83% thought the web site was easy to use.

The test identified one major problem and a few minor problems including:

- The concept of fore- and background is not easy to understand and its user interface is not discoverable.
- The tile picker gets overlooked.
- Lack of feedback when clicking the save button.

This document contains the participant feedback, task completion rates, ease or difficulty of completion ratings, time on task, errors, and recommendations for improvements. A copy of the scenarios and questionnaires are included in the Appendix C section.

#### 5.3 Methodology

This section describes the procedure of the test, what a session looked like and how the participants were recruited.

---

<sup>4</sup><https://www.usability.gov/>

### 5.3.1 Sessions

The test administrator contacted and recruited participants from students and former students they know. Individual appointments were arranged with the participants. Each individual session lasted approximately half an hour. Before the session, the test administrator explained the test session and asked the participant to fill out a brief background questionnaire (see Appendix C.1). Participants read the task scenarios and tried to solve the task.

After each task, the administrator asked the participant a few open questions to explore their post-task subjective experience. The Questions included (see Appendix C.2):

- Is it hard or easy for you to find the right segment?
- Did you notice anything about this segment?
- Did you expect to be able to edit the background landscape?

After the last task was completed, the test administrator asked the participant two final questions:

- Do you think the world editor is rather difficult or rather easy to use?
- Do you know students who might be motivated by this kind of visualization?

### 5.3.2 Participants

All participant were students or former students enrolled in different degree programs. Six participants were scheduled over four testing dates. All six participants completed the test. Two participants were involved in testing on March 24<sup>th</sup>, one on March 29<sup>th</sup>, one on April 3<sup>rd</sup> and two on April 7<sup>th</sup>. Of the six participants, three were male and three were female.

All participants named their degree program (see Table 5.1) and declared to use a computer on a daily base. All of them have experience in using email, word processing, web surfing, games, and multimedia application; 83% in programming, and 67% use database applications.

Table 5.1: Degree programs the participants study or have studied.

Teaching post	Interactive Media	Psychology	Cartography and Geomedia	Fashion designer
1	2	1	1	1

### 5.3.3 Evaluation Tasks/Scenarios

Test participants attempted completion of the following tasks (see Appendix C.4 for complete test tasks):

- Free exploration
- Edit the landscape
- Redeem a Mission reward
- Tear down the landscape
- Edit the background landscape
- Save the changes

## 5.4 Results

This section outlines the results of the test in detail.

### 5.4.1 Task Completion Success Rate

All participants successfully completed Task 4 (tear down the landscape) and Task 6 (save the changes). Five of the six (83%) completed Task 2 (edit the landscape). Half (50%) of the participants were able to complete Task 3 (redeem a Mission reward) and Task 5 (Edit the background landscape). For Task 1 no completion rate was calculated because, due to the nature of the task the participants could not fail it. A detailed list which participant succeeded in which task is given in Table 5.2.

Table 5.2: Task Completion Rates

Participant	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6
1	✓	-	✓	✓	✓	✓
2	✓	✓	✓	✓	-	✓
3	✓	✓	✓	✓	✓	✓
4	✓	✓	-	✓	-	✓
5	✓	✓	-	✓	✓	✓
6	✓	-	-	✓	-	✓
Success	-	5	3	6	3	6
Completion Rates	-	67 %	50 %	100 %	50 %	100 %

### 5.4.2 Time on Task

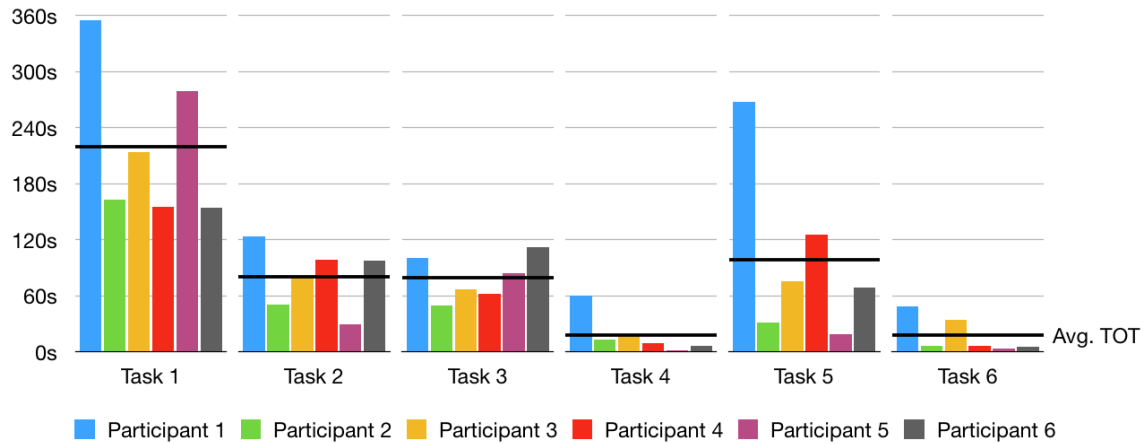


Figure 5.1: Time on task in seconds (Avg. TOT is the mean total time by task).

The test administrator recorded the time on task for each participant. Some tasks were inherently more difficult to complete than others and this is reflected by the average time on task. Figure 5.1 shows a bar chart for the measured time per task and participant, and the average time on task. The chart allows a quick analysis and shows the general trend of each task and Table 5.3 for the exact time on task per participant and task.

Task 1 required participants to freely explore the world editor and is therefore the task with the longest time on task. It shows that the participants after a mean time of 220 seconds think they found out all the functions of the world editor.

Task 5 required participants to edit the background landscape and is the task with the largest variance on time on task. This indicate that this task was perceived different by the participants.

Table 5.3: Time on task (in seconds).

	P1	P2	P3	P4	P5	P6	Avg. TOT
Task 1	355	163	214	155	279	154	220
Task 2	<b>124</b>	51	81	99	30	98	81
Task 3	101	50	67	62	84	<b>112</b>	79
Task 4	<b>77</b>	33	36	29	22	26	37
Task 5	<b>268</b>	32	76	126	19	69	<b>98</b>
Task 6	<b>49</b>	7	34	7	4	6	18

### 5.4.3 Errors

The test administrator captured the number of errors participants made while trying to complete the task scenarios. Table 5.4 shows a summary of errors by participant and task. It shows the sum of critical and non-critical errors. A critical error is an error that does prevent successful completion of the scenario and a non-critical error is an error that does not prevent successful completion of the scenario.

Task 3 required participants to redeem a Mission reward and most of the errors were made by the participants during this task. They made 3 critical errors and 2 non-critical errors. The critical error was always the same, as the participants edited a wrong segment.

Task 6 required participants to save the changes they made. No critical error was made during this task, but 4 participants made the same non-critical error, of pressing the save button several times, because of the lack of feedback of the save button.

Task 4 required participants to tear down the landscape. No critical error and only 1 non-critical error were made by the participants during this task.

Table 5.4: Task Error Rates

Participant	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6
1	2	1	1	1	0	1
2	0	0	0	0	1	1
3	0	0	0	0	0	1
4	1	0	2	0	2	1
5	1	0	1	0	0	0
6	1	1	1	0	1	0
Errors	5	2	<b>5</b>	1	4	4

### 5.4.4 Summary of Data

The Table 5.5 below displays a summary of the test data. Low completion rates and high errors and time on tasks are highlighted in fat.

Table 5.5: Summary of Completion, Errors, Time on Task

Task	Task completion	Errors	Time on Task
1	6	5	220
2	5	2	81
3	<b>3</b>	<b>5</b>	79
4	6	1	37
5	<b>3</b>	4	<b>98</b>
6	6	4	18

### 5.4.5 Overall Metrics

**Overall Ratings** After task session completion, participants were asked two questions on their overall impression of the world editor (See Attachment C). All of the participants (100%) agreed that the web application is easy to use, but also gave some ideas for improvement.

**Likes, Dislikes, Participants' Recommendations** Upon completion of the tasks, participants provided feedback on what they liked most and least about the website, and they also gave some recommendations for improving the website.

**Liked Most** The following comments capture what the participants liked most:

- Look of the landscape and rewards

**Liked Least** The following comments capture what the participants liked the least:

- Editing the background landscape

#### Recommendations for Improvement

- Feedback for the save button
- Undo function
- Better visibility of the tile picker
- Keyboard shortcuts (undo, scrolling of landscape)
- Avatar to play the level (landscape)

## 5.5 Recommendations

The recommendations section provides recommended changes and justifications driven by the participants' success rate, behaviors, and comments. Each recommendation includes a severity rating. For the severity rating multiple factors were taken into account how beneficial is the change for users, how easy is the change to implement, and how important is the change for functionality of the world editor. The following recommendations will improve the overall ease of use and address the areas where participants experienced problems or found the interface/information architecture unclear. The high error rate of Task 3 can probably be explained with a misleading task specification and therefore no recommendation to change is made.

### 5.5.1 Free exploration (Task 1)

Change	Justification	Severity
<ul style="list-style-type: none"> <li>• Support keyboard short-cuts.</li> </ul>	3 participants tried to use the keyboard.	Low
<ul style="list-style-type: none"> <li>• Add undo.</li> </ul>	3 participants ask for an undo function.	Medium

**5.5.2 Edit the landscape (Task 2)**

Change	Justification	Severity
<ul style="list-style-type: none"> <li>• Auto scroll to tile picker.</li> </ul>	2 participants needed the tip: “Scroll down the site.”	High

**5.5.3 Edit the background landscape (Task 5)**

Change	Justification	Severity
<ul style="list-style-type: none"> <li>• Hide layer buttons when not in edit mode.</li> <li>• Give layer buttons a more prominent color.</li> </ul>	50% of the participants oversaw the layer buttons.	Medium

**5.5.4 Save the changes (Task 6)**

Change	Justification	Severity
<ul style="list-style-type: none"> <li>• Show a spinner when the save button was clicked.</li> </ul>	67% of the participants clicked the save button several times to make sure their changes are saved.	High

**5.6 Usability Test Summary**

Most of the participants found the world editor to be well-organized, clean and uncluttered, and easy to use. Having a visualisation for a course and the learning progress on this course was regarded as a beneficial feature by most of the participants. Implementing the recommendations and continuing to work with users will ensure a continued user-centered web application.

Part of this thesis is a usability study of the implemented version of Progress Visualisation Framework but the complete Progress Evaluation Framework needs to be tested in a long term behavioural study, where the effect on learning behaviour and learning progress of students will be evaluated. This study should be at least with the students of one course over the period of a term. The Progress Evaluation Framework can be used in two modes, the first mode allows the students to always edit the landscape, but only to claim rewards after they completed their assignments successfully. The second mode also allows the students to always edit the landscape, but they can claim their reward as soon as get an assignment and this reward decays when they will fail to complete the assignment. Therefore, a study design needs three groups of participants, one for each mode and a control group, or the two modes get evaluated in two separate studies. The validation if the Progress Evaluation Framework has the desired effect on students, helping them to be motivated to learn and to be able to evaluate their learning progress, is the next step for future work.



## 6 Conclusion

On the basis of gamification, that has been shown to be an effective tool set to increase engagement of users, this work discusses the idea of reification as making something real or making something concrete [104]. This thesis stresses with reification, the idea of free, indefinite, and self-serving playing over competitive and comparative gaming. The purpose of reification is not to make users more effective and productive for a third party, for which gamification is used often, but to support and encourage users to self-determined and self-regulated actions. To fulfil the goal of supporting users in self-regulated actions (here students who are learning), the Progress Evaluation Framework is introduced. This framework allows users to evaluate and visualise their actions with a visualisation, which gives them a tangible and continuous feedback about their progress.

The Progress Evaluation Framework is a realization of the concept of reification. It describes how a course with its lectures and learning assignments can be mapped into software (Progress Review Framework) and introduces a reusable model for visualisations (Progress Visualisation Framework). Both frameworks are designed to be flexible and adaptable. The Progress Review Framework has a flexible course model to be able to evaluate all the different types of courses that exist at a university and it can also handle different types of assignments. The Progress Visualisation Framework is not designed for one specific visualisation, instead it can be the foundation for all kind of visualisations. This work stresses on visualisations which have a metaphor or a narrative, but the Progress Visualisation Framework could be also used for abstract visualisations, like charts and scores. A part of this work was the implementation of the Progress Evaluation Framework for Backstage 2. It enables students to reify their learning progress, calculated from solved assignments with the visualisation “creating a landscape”. This visualisation allows students to shape a landscape and fit it out with objects (like trees, bushes, and rocks) and to examine their learning progress by “wander through their landscape”. The implemented visualisation “creating a landscape” and its editor was tested for its usability and most of the participants of the study found it easy to use.

This work extends the idea of gamification towards a more user centred design where the enablement of the user is the focus of the system and tries to overcome the critic of Bogost, that gamification is “exploitationware” [5] and also addresses the problem that gamification often fosters extrinsic motivation [64]. With the idea of customisable visualisation the Progress Evaluation Framework has a unique feature that allows users to reify their learning progress. This reification in form of a visualisation can help students to achieve their goals, by giving them the possibility to overview and to evaluate their learning progress and therefore enables them to take countermeasures if they are underperforming their goals, but also encourages them to carry on if they fulfil their goals. The visualisation of the learning progress can help students to stay motivated and to improve their self-assessment.

However, some limitations are worth noting. Within this thesis it was not possible to implement a study which tested the effects of the Progress Evaluation Framework on learning manners with real students during a term. Furthermore, the implemented visualisation “creating a landscape” could be more vivid, which could have positive effects on learning behaviour and motivation if both are interconnected. At the moment the visualisation is static and only user interaction leads to change. A visualisation that by itself is vivid could be more attractive for users to interact with and shape. Possible ways to fill the visualisation with life are animated rewards (e.g. trees swaying in the wind), or taking the real weather and season into account and adjusting the ground tiles and showing weather effects. One more way to increase the vividness of the visualisation is improving the “wander through their landscape” feature by adding a playable avatar like in a platform game. For some students the feature to compare the visualisations among each other could have positive effects on their learning behaviour and motivation. Future work should therefore include a long-term study which evaluates the effects of the Progress Evaluation Framework on the learning behaviour of students and an improvement of the visualisation to make it more exciting

to use by increasing the playability and vividness.

## References

- [1] Sarah Ball. *Croc-gu-phant (Sarah Ball Books)*. Ragged Bears, 1985.
- [2] William K Balzer, Michael E Doherty, et al. “Effects of cognitive feedback on performance.” In: *Psychological bulletin* 106.3 (1989), p. 410.
- [3] Eric Barone. *Stardew Valley*. <https://stardewvalley.net/>. 2016. [Online; accessed 6-April-2018].
- [4] Eric Barone. *Stardew Valley - Screen shot*. [http://stardewvalley.net/presskit/Stardew%20Valley/images/farm\\_day\\_1920x1080.png](http://stardewvalley.net/presskit/Stardew%20Valley/images/farm_day_1920x1080.png). 2016. [Online; accessed 6-April-2018].
- [5] Ian Bogost. “Gamification is bullshit”. In: *The gameful world: Approaches, issues, applications* (2011), pp. 65–80.
- [6] Ian Bogost. *Persuasive games: The expressive power of videogames*. Mit Press, 2007.
- [7] Deborah L Butler and Philip H Winne. “Feedback and self-regulated learning: A theoretical synthesis”. In: *Review of educational research* 65.3 (1995), pp. 245–281.
- [8] Roger Caillois. *Man, play, and games*. University of Illinois Press, 1961.
- [9] Yu-Kai Chou. *Gamification Research: How FarmVille uses Game Mechanics to become Winning & Addicting*. <http://yukaichou.com/gamification-examples/farmville-game-mechanics-winning-addicting/>. 2013. [Online; accessed 08-February-2018].
- [10] Mia Consalvo. “There is no magic circle”. In: *Games and culture* 4.4 (2009), pp. 408–417.
- [11] Richard DeCharmes. “Personal causation”. In: *New York: Academic* (1968).
- [12] Edward L Deci. “Effects of externally mediated rewards on intrinsic motivation.” In: *Journal of personality and Social Psychology* 18.1 (1971), p. 105.
- [13] Edward L Deci, Richard Koestner, and Richard M Ryan. “Extrinsic rewards and intrinsic motivation in education: Reconsidered once again”. In: *Review of educational research* 71.1 (2001), pp. 1–27.
- [14] Edward L Deci and Richard M Ryan. “Human autonomy”. In: *Efficacy, agency, and self-esteem*. Springer, 1995, pp. 31–49.
- [15] Edward L Deci and Richard M Ryan. “Intrinsic Motivation and Self-Determination in Human Behavior”. In: *Plenum Press* (1985).
- [16] Sebastian Deterding. “Gamification: designing for motivation”. In: *interactions* 19.4 (2012), pp. 14–17.
- [17] Sebastian Deterding, Dan Dixon, Rilla Khaled, and Lennart Nacke. “From game design elements to gamefulness: defining gamification”. In: *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments*. ACM. 2011, pp. 9–15.
- [18] Sebastian Deterding, Miguel Sicart, Lennart Nacke, Kenton O’Hara, and Dan Dixon. “Gamification. using game-design elements in non-gaming contexts”. In: *CHI’11 extended abstracts on human factors in computing systems*. ACM. 2011, pp. 2425–2428.
- [19] Aaron Dignan. *Game frame: Using games as a strategy for success*. Simon and Schuster, 2011.
- [20] Adrián Domínguez, Joseba Saenz-De-Navarrete, Luis De-Marcos, Luis Fernández-Sanz, Carmen Pagés, and José-Javier Martínez-Herráiz. “Gamifying learning experiences: Practical implications and outcomes”. In: *Computers & Education* 63 (2013), pp. 380–392.

- [21] Tao Dong, Mira Dontcheva, Diana Joseph, Karrie Karahalios, Mark Newman, and Mark Ackerman. “Discovery-based games for learning software”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. 2012, pp. 2083–2086.
- [22] P Christopher Earley, Gregory B Northcraft, Cynthia Lee, and Terri R Lituchy. “Impact of process and outcome feedback on the relation of goal setting to task performance”. In: *Academy of Management Journal* 33.1 (1990), pp. 87–105.
- [23] Rosta Farzan, Joan M DiMicco, David R Millen, Beth Brownholtz, Werner Geyer, and Casey Dugan. “When the experiment is over: Deploying an incentive system to all the users”. In: *symposium on persuasive technology*. 2008.
- [24] Rosta Farzan, Joan M DiMicco, David R Millen, Casey Dugan, Werner Geyer, and Elizabeth A Brownholtz. “Results from deploying a participation incentive mechanism within the enterprise”. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM. 2008, pp. 563–572.
- [25] John Ferrara. *Playful design: Creating game experiences in everyday interfaces*. Rosenfeld Media, 2012.
- [26] forestapp.cc. *Forest - Stay focused, be present*. <https://www.forestapp.cc/en/>. [Online; accessed 19-March-2018].
- [27] Node.js Foundation. *Express - Node.js web application framework*. <https://expressjs.com/>. 2017. [Online; accessed 17-April-2018].
- [28] Node.js Foundation. *Express - Node.js web application framework*. <https://nodejs.org/en/>. 2018. [Online; accessed 17-April-2018].
- [29] Google. *Fit - Google*. <https://www.google.com/fit/>. [Online; accessed 19-March-2018].
- [30] Google. *Google Fit – Fitness-Tracking*. <https://play.google.com/store/apps/details?id=com.google.android.apps.fitness&hl=de>. [Online; accessed 19-March-2018].
- [31] Merci Victoria Grace and Justin Hall. “Projecting surveillance entertainment”. In: *Presentation, ETech, San Diego: CA* (2008).
- [32] Fabian Groh. “Gamification: State of the art definition and utilization”. In: *Institute of Media Informatics Ulm University* 39 (2012).
- [33] Lasse Hakulinen, Tapio Auvinen, and Ari Korhonen. “Empirical study on the effect of achievement badges in TRAKLA2 online learning environment”. In: *Learning and Teaching in Computing and Engineering (LaTiCE), 2013*. IEEE. 2013, pp. 47–54.
- [34] Juho Hamari. “Perspectives from behavioral economics to analyzing game design patterns: loss aversion in social games”. In: *the CHI 2011 Social Games Workshop*. 2011.
- [35] Juho Hamari. “Transforming homo economicus into homo ludens: A field experiment on gamification in a utilitarian peer-to-peer trading service”. In: *Electronic commerce research and applications* 12.4 (2013), pp. 236–245.
- [36] Juho Hamari, Jonna Koivisto, and Harri Sarsa. “Does gamification work?—a literature review of empirical studies on gamification”. In: *System Sciences (HICSS), 2014 47th Hawaii International Conference on*. IEEE. 2014, pp. 3025–3034.
- [37] Birgit Harks, Katrin Rakoczy, John Hattie, Michael Besser, and Eckhard Klieme. “The effects of feedback on achievement, interest and self-evaluation: the role of feedback’s perceived usefulness”. In: *Educational Psychology* 34.3 (2014), pp. 269–290.

- [38] John Hattie and Helen Timperley. “The power of feedback”. In: *Review of educational research* 77.1 (2007), pp. 81–112.
- [39] Fritz Heider. *The Psychology of Interpersonal Relations*. Psychology Press, 1958.
- [40] Kai Huotari and Juho Hamari. “Defining gamification: a service marketing perspective”. In: *Proceeding of the 16th International Academic MindTrek Conference*. ACM. 2012, pp. 17–22.
- [41] Facebook Inc. *React - A JavaScript library for building user interfaces*. <https://reactjs.org/>. 2018. [Online; accessed 17-April-2018].
- [42] JH Jung, Christoph Schneider, and Joseph Valacich. “Enhancing the motivational affordance of information systems: The effects of real-time performance feedback and goal setting in group collaboration environments”. In: *Management science* 56.4 (2010), pp. 724–742.
- [43] Jesper Juul. *Half-real: Video games between real rules and fictional worlds*. MIT press, 2011.
- [44] Jesper Juul. “The game, the player, the world: Looking for a heart of gameness”. In: *PLURAIIS-Revista Multidisciplinar* 1.2 (2010).
- [45] Daniel Kahneman, Jack L Knetsch, and Richard H Thaler. “Anomalies: The endowment effect, loss aversion, and status quo bias”. In: *The journal of economic perspectives* 5.1 (1991), pp. 193–206.
- [46] Daniel Kahneman and Amos Tversky. “Choices, values, and frames”. In: *American Psychologist* 39.4 (1984), pp. 341–350.
- [47] Daniel Kahneman and Amos Tversky. “Prospect Theory: An Analysis of Decision under Risk”. In: *Econometrica* 47.2 (1979), pp. 263–292.
- [48] Karl Kapp. *Three Training Technology Predications*. <http://elearnmag.acm.org/featured.cfm?aid=1925057>. 2011. [Online; accessed 15-May-2018].
- [49] Karl M Kapp. *The gamification of learning and instruction: game-based methods and strategies for training and education*. John Wiley & Sons, 2012.
- [50] Kenney. *Kenney - Foliage Pack*. <http://kenney.nl/assets/foilage-pack>. [Online; accessed 26-April-2018].
- [51] Kenney. *Kenney - Home*. <http://kenney.nl/>. 2010. [Online; accessed 26-April-2018].
- [52] Kenney. *Kenney - Platformer Pack Redux*. <http://kenney.nl/assets/platformer-pack-redux>. [Online; accessed 26-April-2018].
- [53] Julius Kuhl and Thomas Goshke. “A theory of action control: Mental subsystems, modes of control, and volitional conflict-resolution strategies”. In: *Volition and personality: Action versus state orientation* 5 (1994), pp. 93–124.
- [54] Raymond W Kulhavy. “Feedback in written instruction”. In: *Review of educational research* 47.2 (1977), pp. 211–232.
- [55] E Lawley. “Games as an alternate lens for design”. In: *Interactions* 19.4 (2012), pp. 16–17.
- [56] Wei Li, Tovi Grossman, and George Fitzmaurice. “GamiCAD: a gamified tutorial system for first time autocad users”. In: *Proceedings of the 25th annual ACM symposium on User interface software and technology*. ACM. 2012, pp. 103–112.
- [57] Photon Storm Ltd. *Phaser CE*. <https://github.com/photonstorm/phaser-ce>. 2018. [Online; accessed 17-April-2018].

- [58] Thomas W Malone. “Heuristics for designing enjoyable user interfaces: Lessons from computer games”. In: *Proceedings of the 1982 conference on Human factors in computing systems*. ACM. 1982, pp. 63–68.
- [59] Thomas W Malone. “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*. ACM. 1980, pp. 162–169.
- [60] Marc McDonald, Robert Musson, and Ross Smith. “Using productivity games to prevent defects”. In: *The Practical Guide to Defect Prevention*, Microsoft Press, Redmond 7 (2008).
- [61] Jane McGonigal. *Reality is broken: Why games make us better and how they can change the world*. Penguin, 2011.
- [62] Markus Montola, Timo Nummenmaa, Andrés Lucero, Marion Boberg, and Hannu Korhonen. “Applying game achievement systems to enhance user experience in a photo sharing service”. In: *Proceedings of the 13th International MindTrek Conference: Everyday Life in the Ubiquitous Era*. ACM. 2009, pp. 94–97.
- [63] Cristina Ioana Muntean. “Raising engagement in e-learning through gamification”. In: *Proc. 6th International Conference on Virtual Learning ICVL*. 42. 2011, pp. 323–329.
- [64] Scott Nicholson. “A user-centered theoretical framework for meaningful gamification”. In: *Games+ Learning+ Society* 8.1 (2012), pp. 223–230.
- [65] David J Nicol and Debra Macfarlane-Dick. “Formative assessment and self-regulated learning: A model and seven principles of good feedback practice”. In: *Studies in higher education* 31.2 (2006), pp. 199–218.
- [66] Jakob Nielsen. *How Many Test Users in a Usability Study?* <https://www.nngroup.com/articles/how-many-test-users/>. 2012. [Online; accessed 16-May-2018].
- [67] Nintendo. *Super Mario Maker*. <http://supermariomaker.nintendo.com/>. 2015. [Online; accessed 9-April-2018].
- [68] Paul R Pintrich and Akane Zusho. “Student Motivation and Self-Regulated Learning in the College Classroom”. In: *Higher Education: Handbook of Theory and Research*. Springer, 2002, pp. 55–128.
- [69] Alexander Pohl, Francois Bry, and Yingding Wang. *Backstage - A Digital Backchannel for Large Class Lectures*. <http://backstage.pms.ifi.lmu.de/>. 2014. [Online; accessed 08-March-2018].
- [70] Seth Priebatsch. *Seth Priebatsch: The game layer on top of the world*. Ted, 2010.
- [71] Nola Purdie, John Hattie, and Graham Douglas. “Student conceptions of learning and their use of self-regulated learning strategies: A cross-cultural comparison.” In: *Journal of educational psychology* 88.1 (1996), p. 87.
- [72] Redbox and SuperMono. *Epic win - Level-Up Your Life*. <http://www.rexbox.co.uk/epicwin/>. [Online; accessed 8-March-2018].
- [73] *RethinkDB: the open-source database for the realtime web*. <https://www.rethinkdb.com/>. 2018. [Online; accessed 17-April-2018].
- [74] Margaret Robertson. “Can’t play, won’t play”. In: *Hide & Seek* 6 (2010), p. 2010.
- [75] Richard M Ryan. “A motivational approach to self: Integration in personality edward l, deci and”. In: *Perspectives on motivation* 38 (1991), p. 237.
- [76] Richard M Ryan and James P Connell. “Perceived locus of causality and internalization: examining reasons for acting in two domains.” In: *Journal of personality and social psychology* 57.5 (1989), p. 749.



- [77] Richard M Ryan and Edward L Deci. "Intrinsic and extrinsic motivations: Classic definitions and new directions". In: *Contemporary educational psychology* 25.1 (2000), pp. 54–67.
- [78] Richard M Ryan and Edward L Deci. "Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being." In: *American psychologist* 55.1 (2000), p. 68.
- [79] Richard M Ryan, Edward L Deci, and Wendy S Grolnick. "Autonomy, relatedness, and the self: Their relation to development and psychopathology". In: *Ariel* 128.151.189 (1995), p. 155.
- [80] Katie Salen and Eric Zimmerman. *Rules of play: Game design fundamentals*. MIT press, 2004.
- [81] Roger Saljo. *Learning in the learner's perspective. I. Some common-sense conceptions*. ERIC Clearinghouse, 1979.
- [82] Katie Seaborn and Deborah I Fels. "Gamification in theory and action: A survey". In: *International Journal of Human-Computer Studies* 74 (2015), pp. 14–31.
- [83] Kennon M Sheldon, Richard M Ryan, Laird J Rawsthorne, and Barbara Ilardi. "Trait self and true self: Cross-role variation in the Big-Five personality traits and its relations with psychological authenticity and subjective well-being." In: *Journal of personality and social psychology* 73.6 (1997), p. 1380.
- [84] ShoeBox. <http://renderhjs.net/shoebox/>. [Online; accessed 8-May-2018].
- [85] Burrhus Frederic Skinner. *Science and human behavior*. Simon and Schuster, 1953.
- [86] Nando Stöcklin. *Who coined the term "gamification"?* <https://www.quora.com/Who-coined-the-term-gamification>. 2012. [Online, accessed November 21, 2017].
- [87] Dean Takahashi. "Funware's threat to the traditional video game industry". In: *venturebeat.com, en ligne: http://goo.gl/09lSq* (2008).
- [88] TL Taylor. "The assemblage of play". In: *Games and Culture* 4.4 (2009), pp. 331–339.
- [89] Jennifer Thom, David Millen, and Joan DiMicco. "Removing gamification from an enterprise SNS". In: *Proceedings of the acm 2012 conference on computer supported cooperative work*. ACM. 2012, pp. 1067–1070.
- [90] Amos Tversky and Daniel Kahneman. "Loss aversion in riskless choice: A reference-dependent model". In: *The quarterly journal of economics* 106.4 (1991), pp. 1039–1061.
- [91] U.S. Department of Health & Human Services. *Report Template: Usability Test*. <https://www.usability.gov/how-to-and-tools/resources/templates/report-template-usability-test.html>. [Online; accessed 16-May-2018].
- [92] U.S. Department of Health & Human Services. *Running a Usability Test*. <https://www.usability.gov/how-to-and-tools/methods/running-usability-tests.html>. [Online; accessed 16-May-2018].
- [93] Luis Von Ahn. "Games with a purpose". In: *Computer* 39.6 (2006), pp. 92–94.
- [94] David Watkins and Murari Regmi. "How universal are student conceptions of learning? A Nepalese investigation." In: *Psychologia: An International Journal of Psychology in the Orient* (1992).
- [95] Fiona Watt. *Cut-Out Paper Dolls*. Paper dolls and models. Usborne Publishing, 2017.
- [96] Robert W White. "Motivation reconsidered: The concept of competence." In: *Psychological review* 66.5 (1959), p. 297.



- [97] Wikipedia contributors. *Feedback* — *Wikipedia, The Free Encyclopedia*. <https://en.wikipedia.org/w/index.php?title=Feedback&oldid=823132060>. 2018. [Online; accessed 27-January-2018].
- [98] Wikipedia contributors. *Illusory superiority* — *Wikipedia, The Free Encyclopedia*. [https://en.wikipedia.org/w/index.php?title=Illusory\\_superiority&oldid=828002817](https://en.wikipedia.org/w/index.php?title=Illusory_superiority&oldid=828002817). 2018. [Online; accessed 8-March-2018].
- [99] Wikipedia contributors. *Loss aversion* — *Wikipedia, The Free Encyclopedia*. [https://en.wikipedia.org/w/index.php?title=Loss\\_aversion&oldid=815192611](https://en.wikipedia.org/w/index.php?title=Loss_aversion&oldid=815192611). 2017. [Online; accessed 21-January-2018].
- [100] Wikipedia contributors. *Overconfidence effect* — *Wikipedia, The Free Encyclopedia*. [https://en.wikipedia.org/w/index.php?title=Overconfidence\\_effect&oldid=822626061](https://en.wikipedia.org/w/index.php?title=Overconfidence_effect&oldid=822626061). 2018. [Online; accessed 8-March-2018].
- [101] Wikipedia contributors. *Parallax scrolling* — *Wikipedia, The Free Encyclopedia*. [https://en.wikipedia.org/w/index.php?title=Parallax\\_scrolling&oldid=824405047](https://en.wikipedia.org/w/index.php?title=Parallax_scrolling&oldid=824405047). 2018. [Online; accessed 26-April-2018].
- [102] Wikipedia contributors. *Platform game* — *Wikipedia, The Free Encyclopedia*. [https://en.wikipedia.org/w/index.php?title=Platform\\_game&oldid=833150424](https://en.wikipedia.org/w/index.php?title=Platform_game&oldid=833150424). 2018. [Online; accessed 9-April-2018].
- [103] Wikipedia contributors. *Prospect theory* — *Wikipedia, The Free Encyclopedia*. [https://en.wikipedia.org/w/index.php?title=Prospect\\_theory&oldid=818707079](https://en.wikipedia.org/w/index.php?title=Prospect_theory&oldid=818707079). 2018. [Online; accessed 22-January-2018].
- [104] Wikipedia contributors. *Reification* — *Wikipedia, The Free Encyclopedia*. <https://en.wikipedia.org/w/index.php?title=Reification&oldid=783820717>. 2017. [Online; accessed 8-March-2018].
- [105] Wikipedia contributors. *Texture atlas* — *Wikipedia, The Free Encyclopedia*. [https://en.wikipedia.org/w/index.php?title=Texture\\_atlas&oldid=828576106](https://en.wikipedia.org/w/index.php?title=Texture_atlas&oldid=828576106). 2018. [Online; accessed 8-May-2018].
- [106] Wikipedia contributors. *Video game graphics* — *Wikipedia, The Free Encyclopedia*. [https://en.wikipedia.org/w/index.php?title=Video\\_game\\_graphics&oldid=827939311](https://en.wikipedia.org/w/index.php?title=Video_game_graphics&oldid=827939311). 2018. [Online; accessed 9-April-2018].
- [107] Wikipedia contributors. *Web application* — *Wikipedia, The Free Encyclopedia*. [https://en.wikipedia.org/w/index.php?title=Web\\_application&oldid=835981007](https://en.wikipedia.org/w/index.php?title=Web_application&oldid=835981007). 2018. [Online; accessed 17-April-2018].
- [108] Peter C Wright, K OVERBEEKE, AF MONK, and MA BLYTHE. “Funology: from usability to enjoyment”. In: *Human-Computer Interaction Series 3* (2003).
- [109] Tiled Documentation Writers. *JSON Map Format* — *Tiled 1.1.0 documentation*. <http://doc.mapeditor.org/en/stable/reference/json-map-format/>. 2017. [Online; accessed 2-May-2018].
- [110] Gabe Zichermann and Christopher Cunningham. *Gamification by design: Implementing game mechanics in web and mobile apps*. O’Reilly Media, Inc., 2011.
- [111] Gabe Zichermann and Joselin Linder. *Game-based marketing: inspire customer loyalty through rewards, challenges, and contests*. John Wiley & Sons, 2010.
- [112] Zynga. *Grow Your Own Farm Online with FarmVille*. <https://www.zynga.com/games/farmville>. 2009. [Online; accessed 6-April-2018].

## A Data Model

### A.1 Progress Evaluation Framework

#### A.1.1 Reification Object

Reification object is the root element of the Progress Evaluation Framework data model; actually, it is the complement to the course object from the Backstage 2 data model (see Listing 14).

---

```

1 {
2   "id": "db0954d2-9df9-4aea-b30f-5ae22cb2aaa5",
3   "creatorIds": ["eca6b366-22cf-4265-98a9-b23849439c11"],
4   "courseId": "87303924-5a29-4198-909a-973dd2401c63",
5   "courseTitle": "Logik und diskrete Strukturen",
6   "courseType": "default",
7   "beginDate": 1517410333569,
8   "endDate": 1517410334569,
9   "segmentWidth": 5,
10  "segments": [
11    "585f8e1f-96c2-42c0-b85f-ecf8c8271dcb",
12    "c5956bef-2f18-4d3e-a8b9-755d1b7ccbc4",
13    "9e3c6cdb-a83c-4a41-84c9-30f37476a587",
14    "cdb2dfc9-7557-4c0d-a075-c4237e526312",
15    "c1d82148-7788-4a8a-85b1-9b49c11dd1eb",
16    "7e496af2-c885-47a9-8652-7cb7edebeecd7",
17    "d2188aec-40b7-4450-95b7-9b2806c44638",
18    "c1fcbb6f-c91e-41cf-8b98-c9947b0b9842"
19  ]
20 }
```

---

Listing 14: Reification object

- “id”, a unique identifier for this set of data, generated by the data base.
- “creatorIds”, a list of unique identifiers from the users table of the Backstage 2 data model.
- “courseId”, a unique identifier from the courses table of the Backstage 2 data model.
- “courseTitle”, the title of the course copied from the courses table.
- “courseType”, the type of the course copied from the courses table.
- “beginDate”, the date of the start of the Reification.
- “endDate”, the date of the end of the Reification.
- “segmentWidth”, the size of the segment.
- “segments”, a list of unique identifiers from the Segments table.

### A.1.2 Segment Object

Segment object maps a lecture of a course to the Progress Evaluation Framework data model (see Listing 15).

---

```

1 {
2   "id": "6ff22c61-61e9-4c08-9aae-53acba637b9a",
3   "creatorIds": ["eca6b366-22cf-4265-98a9-b23849439c11"],
4   "courseMaterialId": "e41119ff-d5a6-404b-9569-2dc8b2948a00",
5   "courseMaterialTitle": "Natürliche Zahlen",
6   "courseMaterialType": "simple",
7   "beginDate": 1519772400000,
8   "endDate": 1520377199000,
9   "segmentNumber": 5,
10  "defaultMissions": ["lecture", "exercise", "readpaper"]
11 }
```

---

Listing 15: Segment object

- “id”, a unique identifier for this set of data, generated by the data base.
- “creatorIds”, a list of unique identifiers from the users table of the Backstage 2 data model.
- “courseMaterialId”, a unique identifier from the course material table of the Backstage 2 data model.
- “courseMaterialTitle”, the title of the course material copied from the course material table.
- “courseMaterialType”, the type of the course material copied from the course material table.
- “beginDate”, the date of the start of the Segment.
- “endDate”, the date of the end of the Segment.
- “segmentNumber”, the position of the Segment, in the order of all Segments of a Reification.
- “defaultMissions”, a list of default Missions.

### A.1.3 Mission Object

Mission object maps an assignment to the Progress Evaluation Framework data model (see Listing 16).

---

```

1 {
2   "id": "2b3a9ff7-94ac-404d-9589-28d0322d1afb",
3   "creatorIds": ["eca6b366-22cf-4265-98a9-b23849439c11"],
4   "ownerId": "eca6b366-22cf-4265-98a9-b23849439c11",
5   "segmentId": "585f8e1f-96c2-42c0-b85f-ecf8c8271dcb",
6   "assignmentId": null,
7   "parentMissionId": null,
8   "beginDate": 1517353200000,
9   "endDate": 1517957999000,
10  "currentPoints": 0,
11  "targetPoints": 100,
12  "type": "point",
13  "pattern": null,
14  "title": "Attend the lecture.",
15  "description": "Be part of the silent and maybe bored
16    audience.",
17  "note": "Important stuff!"
18 }
```

---

Listing 16: Mission object

- “id”, a unique identifier for this set of data, generated by the data base.
- “creatorIds”, a list of unique identifiers from the users table of the Backstage 2 data model.
- “ownerId”, a unique identifier from the users table of the Backstage 2 data model (identifies the user who has to do the Mission).
- “segmentId”, a unique identifier from the Segments table.
- “assignmentId”, a unique identifier from the assignments table of the Backstage 2 data model.
- “parentMissionId”, a unique identifier from the missions table (identifies the Mission, that has been copied to create this Mission).
- “beginDate”, the date of the start of the Mission.
- “endDate”, the date of the end of the Mission.
- “currentPoints”, the amount of points already received for this Mission.
- “targetPoints”, the total amount of possible points.
- “type”, the type of the Mission (point, single level, or multi level).
- “pattern”, property for future use to select a reward before the mission is solved.
- “title”, the title of the Mission.
- “description”, the description of the Mission.
- “note”, user notes for the Mission.

### A.1.4 Visualisation Object

Visualisation object stores all information which is necessary to render the Visualisation with Phaser CE (see Listing 17). The object has properties that are independent from the type of the Visualisation and properties that depend on the type of the Visualisation.

---

```

1 {
2   "id": "09e49f4f-24b9-4cef-8cd6-b1e1b60fa934",
3   "ownerId": "eca6b366-22cf-4265-98a9-b23849439c11",
4   "reificationId": "db0954d2-9df9-4aea-b30f-5ae22cb2aaa5",
5   "courseId": "87303924-5a29-4198-909a-973dd2401c63",
6   "createdAt": 1524743413295,
7   "createdBy": "user",
8   "type": "platformer",
9   "data": {}
10 }
```

---

Listing 17: Visualisation object

- “id”, a unique identifier for this set of data, generated by the data base.
- “ownerId”, a unique identifier from the users table of the Backstage 2 data model.
- “reificationId”, a unique identifier from the reifications table.
- “courseId”, a unique identifier from the courses table of the Backstage 2 data model.
- “createdAt”, a timestamp when this Visualisation object was created.
- “createdBy”, identifies if this Visualisation object was created by a user or the system.
- “type”, the type of the Visualisation (at the moment always “platformer”).
- “data”, a object that contains all the data necessary to render the visualisation. The structure of data depends on the type of the visualisation.

**Data object** for “creating a landscape” as shown in the code Listing 18 has a simple structure. It has five properties: “segmentWidth”, “landscapeBackground”, “landscapeForeground”, “rewardBackground”, and “rewardForeground”.

---

```

1 {
2   "segmentWidth": 5,
3   "landscapeBackground": [
4     [
5       [0, 0, 0, 0, 0],
6       [0, 0, 0, 0, 0],
7       [0, 0, 0, 0, 0],
8       [0, 0, 0, 0, 0],
9       [0, 0, 0, 0, 0],
10      [0, 0, 0, 0, 0],
11      [22, 22, 22, 22, 22],
12      [12, 12, 12, 12, 12]
13    ]
14  ],
15  "landscapeForeground": [
16    [
17      [0, 0, 0, 0, 0],
18      [0, 0, 0, 0, 0],
19      [0, 0, 0, 0, 0],
20      [0, 0, 0, 0, 0],
21      [0, 0, 0, 0, 0],
22      [0, 0, 0, 1, 0],
23      [22, 22, 22, 22, 22],
24      [12, 12, 12, 12, 12]
25    ]
26  ],
27  "rewardBackground": {},
28  "rewardForeground": {
29    "6c8552ba-a964-4901-b437-708446cc86c7": {
30      "missionId": "6c8552ba-a964-4901-b437-708446cc86c7",
31      "data": {
32        "segment": 1,
33        "x": 420,
34        "y": 70,
35        "tileSheet": "items.png",
36        "tileFrame": {
37          "filename": "cloud1.png",
38          "frame": { "x": 0, "y": 294, "w": 128, "h": 71 },
39          "spriteSourceSize": { "x": 0, "y": 0, "w": 128, "h": 71 },
40          "sourceSize": { "w": 128, "h": 71 },
41          "reificationInfo": {
42            "visible": true,
43            "index": 0
44          }
45        }
46      }
47    }
48  }
49 }

```

---

Listing 18: Data object for the visualisation “creating a landscape”

- “segmentWidth” stores the number that determines the number of tiles a Segment in the Visualisation has.
- “landscapeBackground” and “landscapeForeground” are a structure of nested arrays, where in the root array there is one array for each Segment in the visualisation and in each Segment array there are eight arrays which represent a row of tiles. A row of tiles array contains as many numbers as the “segmentWidth” determines, each number represents one tile in the visualisation. This structure of nested arrays was chosen because it is easy to edit as it reproduces the structure of a Reification and its Segments.
- “rewardBackground”, and “rewardForeground” are dictionaries realised with JavaScript objects. Each dictionary record consists of a unique identifier (id of a Mission) as a key and an object that contains all information necessary to render the reward.
  - “segment”, the Segment this rewards belongs to as number.
  - “x”, the x coordinate where in the visualisation the reward gets rendered.
  - “y”, the y coordinate where in the visualisation the reward gets rendered.
  - “tileSheet”, the name of the tile sheet the reward graphic is part of.
  - “tileFrame”, detailed information to the tile based on the ShoeBox sprite sheet JSON format.



## A.2 Third Party Tools

### A.2.1 ShoeBox sprite sheet JSON format

ShoeBox sprite sheet JSON format is the meta data format that ShoeBox [84] generates when creating a sprite sheet (see Listing 19). It contains detailed information to each graphic in a sprite sheet, for the Progress Visualisation Framework this JSON files get some extensions. The JSON format contains an array with a frame object for each graphic; this frame object contains information to the name, the size, and the position of the graphic and Progress Visualisation Framework extensions.

---

```

1 {
2   "frames": [
3     {
4       "filename": "grass.png",
5       "frame": { "x": 140, "y": 280, "w": 70, "h": 70 },
6       "spriteSourceSize": { "x": 0, "y": 0, "w": 70, "h": 70 },
7       "sourceSize": { "w": 70, "h": 70 },
8       "reificationInfo": {
9         "visible": true,
10        "index": 27,
11        "bloomIndex": 27,
12        "decayIndex": 21,
13        "deadIndex": 21
14      }
15    }
16  ],
17  "meta": {
18    "image": "grounds.png",
19    "size": { "w": 420, "h": 420 },
20    "scale": "1"
21  }
22 }
```

---

Listing 19: ShoeBox sprite sheet JSON format

- “filename”, the name of the source graphic.
- “frame”, the position and size the graphic has in the sprite sheet.
- “spriteSourceSize”, irrelevant data for the Progress Visualisation Framework.
- “sourceSize”, irrelevant data for the Progress Visualisation Framework.
- “reificationInfo”, extended information for the Progress Visualisation Framework.
  - “visible”, boolean flag if the the graphic is visible in the TilePicker component.
  - “index”, the index of the graphic (count from left to right and start with one).
  - “bloomIndex”, “decayIndex”, and “deadIndex”, are the indexes for replacement graphics for the decay system.

### A.2.2 Tiled Map Editor JSON map format

Tiled Map Editor JSON map format [109] is the format, that Tiled Map Editor uses to store maps (see Listing 20). It is one of the supported map formats of Phaser CE, so the Progress Visualisation Framework uses it as an intermediate format. When loading a visualisation into Phaser CE the Visualisation object gets transformed into this format.

---

```

1  {
2    "height": 8,
3    "infinite": false,
4    "layers": [
5      {
6        "data": [
7          0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
8          0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
9          0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
10         0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
11         0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
12         0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
13         22, 22, 22, 22, 22, 22, 22, 22, 22, 22,
14         12, 12, 12, 12, 12, 12, 12, 12, 12, 12
15       ],
16       "height": 8,
17       "name": "landscapeForeground",
18       "opacity": 1,
19       "type": "tilelayer",
20       "visible": true,
21       "width": 10,
22       "x": 0,
23       "y": 0
24     }
25   ],
26   "nextobjectid": 1,
27   "orientation": "orthogonal",
28   "renderorder": "right-down",
29   "tiledversion": "1.1.3",
30   "tileheight": 70,
31   "tilesets": [
32     {
33       "columns": 6,
34       "firstgid": 1,
35       "image": "grounds.png",
36       "imageheight": 420,
37       "imagewidth": 420,
38       "margin": 0,
39       "name": "grounds",
40       "spacing": 0,
41       "tilecount": 36,
42       "tileheight": 70,
43       "tilewidth": 70
44     }
45   ],

```

```
46  "tilewidth": 70,  
47  "type": "map",  
48  "version": 1,  
49  "width": 10  
50 }
```

---

Listing 20: Tiled Map Editor JSON map format

## B File Listing

### B.1 Client Side

```

client/components/reification/
├── missionlist/
│   ├── MissionList.jsx
│   ├── MissionListItem.jsx
│   └── MissionTilePicker.jsx
├── tilepicker/
│   ├── Tile.jsx
│   ├── TileEraser.jsx
│   └── TilePicker.jsx
├── worldeditortoolbar/
│   ├── Button.jsx
│   ├── IconButton.jsx
│   ├── RadioButton.jsx
│   ├── SegmentToolbar.jsx
│   └── WorldEditorToolbar.jsx
├── PhaserContainer.jsx
└── WorldEditor.jsx

```

Listing 21: File listing of client side files containing the component classes.

**MissionList.jsx** file contains the MissionList React component. It creates a list of MissionListItem components. The MissionList component needs as input a Segment object with its Missions, a list of Reward objects, and sprite sheets and their meta data. For each Mission of the Segment a “MissionListItem” gets rendered.

**MissionListItem.jsx** file contains the MissionListItem React component. It renders the informations of a Mission (title and description) and creates a MissionTilePicker component. The MissionListItem component needs as input a Mission object, a Reward object (optional), and sprite sheets and their meta data.

**MissionTilePicker.jsx** file contains the MissionTilePicker React component. It creates a list of Tile components, with IconButton components to skip through this list and a Tile component to toggle the visibility of this list. The MissionTilePicker component needs as input a sprite sheet and its meta data, and a Reward object.

**Tile.jsx** file contains the Tile React component. It renders one tile of a sprite sheet or a button. The Tile component needs as input a sprite sheet and its size, and a TileFrame object. Normally a cutout of a sprite sheet gets rendered by the Tile component, but if the TileFrame object has special parameters instead a button gets rendered, this function is used by the MissionTilePicker component.

**TileEraser.jsx** file contains the TileEraser React component. It renders a button with an eraser icon.

**TilePicker.jsx** file contains the TilePicker React component. It creates a list of Tile components, with IconButton components to skip through this list. The TilePicker component needs as input a sprite sheet and its meta data.

**Button.jsx** file contains the Button React component. It renders a button with a title. The Button component needs as input a title.

**IconButton.jsx** file contains the IconButton React component. It renders a button with an icon. The IconButton component needs as input the name of an icon.

**RadioButton.jsx** file contains the RadioButton React component. It renders a button with a title and imitates the radio button behaviour where only one button of a group of buttons can be active. The RadioButton component needs as input a title, a number (value), and a number (selected). Value is used by the RadioButton component to identify itself and selected to determine if it is active.

**SegmentToolbar.jsx** file contains the SegmentToolbar React component. It renders a headline. The SegmentToolbar component needs as input a Segment title.

**WorldEditorToolbar.jsx** file contains the WorldEditorToolbar React component. It creates two groups of RadioButton components and a Button component. The WorldEditorToolbar component needs as input a number (selectedLayer), and a number (editorModus).

**PhaserContainer.jsx** file contains the PhaserContainer React component. It wraps the Phaser CE into a React component and contains the playState object. The PhaserContainer component needs as input sprite sheets and their meta data, a Reification object and a Visualisation object. It contains the playState is a class, that implements the necessary callback methods for the usage of Phaser CE.

**WorldEditor.jsx** file contains the WorldEditor React component. It is the main component and renders the view with which the user interacts. The WorldEditor component needs as input the unique identifier of a Backstage 2 course object. It manages the states between all the other components.

```
client/stores/  
├── models/  
│   ├── Reification.js  
│   ├── ReificationHelper.js  
│   ├── ReificationMission.js  
│   ├── ReificationSegment.js  
│   └── ReificationVisualisation.js  
└── reificationStore.js
```

Listing 22: File listing of client side files containing the store and model classes.

**Reification.js** file contains the Reification class. It implements the Reification data model on client side.

**ReificationHelper.js** file contains the ReificationHelper class.

**ReificationMission.js** file contains the ReificationInfo, the TileFrame, the SelectedTile, the EditedTile, the PlatformerRewardData, and the Reward classes. These classes help to implement a type system.

**ReificationSegment.js** file contains the ReificationSegment class. It implements the Segment data model on client side.

**ReificationVisualisation.js** file contains the ReificationVisualisation class. It implements the Visualisation data model on client side.

**reificationStore.js** file contains the reificationStore class. It implements the communication with the server and caches Reification and ReificationVisualisation objects.

**B.1.1 Assets**

```
client/components/reification/assets/  
├─ bggrasslands.png  
├─ exampletiledmap.json  
├─ grounds.json  
├─ grounds.png  
├─ items.json  
├─ items.png  
├─ transparent.png  
├─ trees.json  
└─ trees.png
```

Listing 23: File listing of the image files and their meta data files.

**bggrasslands.png** is the background image of the “creating a landscape” visualisation.

**exampletiledmap.json** is an example map in the Tiled Map Editor JSON map format.

**grounds.json** contains the meta data in the ShoeBox sprite sheet JSON format for the grounds.png sprite sheet.

**grounds.png** is a sprite sheet that contains all images for the landscape.

**items.json** contains the meta data in the ShoeBox sprite sheet JSON format for the items.png sprite sheet.

**items.png** is a sprite sheet that contains images for the rewards.

**transparent.png** is a transparent image.

**trees.json** contains the meta data in the ShoeBox sprite sheet JSON format for the trees.png sprite sheet.

**trees.png** is a sprite sheet that contains images for the rewards.

## B.2 Server side

```
server/
├── dao/
│   ├── reificationDAO.js
│   ├── reificationMissionDAO.js
│   ├── reificationSegmentDAO.js
│   └── reificationVisualisationDAO.js
└── rest/
    └── reificationService.js
```

Listing 24: File listing of the server side files containing the DAO and service classes.

**reificationDAO.js** file contains the `reificationDAO` class. It implements the CRUD methods for the Reification data model.

**reificationMissionDAO.js** file contains the `reificationMissionDAO` class. It implements the CRUD methods for the Mission data model.

**reificationSegmentDAO.js** file contains the `reificationSegmentDAO` class. It implements the CRUD methods for the Segment data model.

**reificationVisualisationDAO.js** file contains the `reificationVisualisationDAO` class. It implements the CRUD methods for the Visualisation data model.

**reificationService.js** file contains the `reificationService` class. It implements the REST service for the Progress Evaluation Framework.



### B.3 Share

```
share/
├── enumerations/
│   ├── defaultMissions.js
│   └── missionTypes.js
└── reification/
    └── reificationconfig.js
```

Listing 25: File listing of the shared files.

**defaultMissions.js** file contains the DefaultMissions enumeration.

**missionTypes.js** file contains the MissionTypes enumeration.

**reificationconfig.js** file contains the ReificationConfig enumeration.

## C Usability Test

### C.1 Background Questionnaire

- How old are you?
- What is your gender?
- What is your study / Profession?
- How often do you use a computer?
  - Never
  - Occasionally
  - Monthly
  - Weekly
  - Daily
- What application do you use? (please check all that apply)
  - Email
  - Word Processing
  - Web Surfing
  - Games
  - Database
  - Multimedia
  - Programming

### C.2 Post-Task Questionnaire

- Task 2
  - Was it hard or easy for you to find the right segment?
  - Was it hard or easy for you to find the water tiles?
  - What has been hard for you?
- Task 3
  - Did you notice anything about this segment?
  - Why do you think the landscape looks different?
  - What do you associate with this landscape?
- Task 4
  - Did it take you a rather long or a rather short time to discover the eraser?
  - Why do you think you have been looking a long time for it?
  - What did you expect how the erasing of the tiles would look like?
- Task 5
  - Did you expect to be able to edit the background landscape?
  - Why did you expect it?
  - Why did you not expect it?

- Task 6
  - Did it take you a rather long or a rather short time to discover the save button?
  - Why do you think you have been looking for it a long time?
  - Where did you expect the save button to be?

### **C.3 Post-session Overall Subjective Questionnaire**

- Do you think the world editor is rather difficult or rather easy to use?
- Do you know students who might be motivated by this kind of visualization?

### **C.4 Task Scenarios**

- Task 1
  - Free exploration, familiarize yourself with the world editor.
- Task 2
  - Edit the landscape of the segment “Narural numbers”, by building a moat
- Task 3
  - Redeem a Mission Reward in the “Resolution” segment, for the Mission “Submit the exercise” a yellow star.
- Task 4
  - Edit the landscape of the “Modular Arithmetic” segment, delete the top elements of the hill.
- Task 5
  - Edit the background landscape of the segment “Statement Logic”, build something that is higher as the foreground landscape.
- Task 6
  - Save your changes.