



Julian Philipp Wolf, BSc

# User Assessment in 3D Environments

## Master's Thesis

to achieve the university degree of

Master of Science

Master's degree programme: Computer Science

submitted to

**Graz University of Technology**

Supervisor

Assoc.Prof. Dipl.-Ing. Dr.techn. Christian Gütl

Co-Supervisor

Dipl.-Ing. Dr.techn. Johanna Pirker, BSc

Mag. Joachim Maderer

Institute of Interactive Systems and Data Science

Head: Univ.-Prof. Dipl.-Inf. Dr. Stefanie Lindstaedt

Graz, January 2019

This document is set in Palatino, compiled with pdfL<sup>A</sup>T<sub>E</sub>X<sub>2</sub><sup>ε</sup> and Biber.

The L<sup>A</sup>T<sub>E</sub>X template from Karl Voit is based on KOMA script and can be found online:  
<https://github.com/novoid/LaTeX-KOMA-template>



Julian Philipp Wolf, BSc

# Benutzerbewertung in 3D Umgebungen

## Masterarbeit

zur Erlangung des Akademischen Grades

Diplom Ingenieur

Masterstudium: Computer Science

an der

**Technischen Universität Graz**

Betreuer

Assoc.Prof. Dipl.-Ing. Dr.techn. Christian Gütl

Mitbetreuer

Dipl.-Ing. Dr.techn. Johanna Pirker, BSc

Mag. Joachim Maderer

Institut für Informationssysteme und Computer Medien

Leitung: Univ.-Prof. Dipl.-Inf. Dr. Stefanie Lindstaedt

Graz, Jänner 2019

## **Affidavit**

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master's thesis.

---

Date

---

Signature

## **Eidesstattliche Erklärung**

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Dissertation identisch.

---

Datum

---

Unterschrift



# Abstract

Virtual Learning Environments gain more and more attraction. Nevertheless, most of these systems are not able to ensure constant quality, by providing individual immediate feedback according to the skills. This holds especially true for systems, which are not directly attended by human tutors. In particular with immersive environments this shows to be a challenging problem.

Automatic assessment systems can provide immediate feedback which has been proven to be very important during the learning process. Furthermore, traditional feedback strategies such as multiple-choice tests help to deepen the newly gained knowledge. Although there is a lot of research going on in regards on either immersive learning environments or assessment systems, the amount of research of combining these two fields seems rather untouched, especially targeting physics experiments.

This work proposes a technique to combine the fields immersive environment and assessment system. This assessment system is capable of semantic event tracking and is able to communicate with multiple virtual environment of any type at any time. As a proof of concept, a pendulum experiment was created and the assessment system enhanced with an assignment sheet. This lead to a supervised learning environment for students.

To ensure the goals were achieved, a survey including 8 children of middle school age was conducted. The results were promising and provided a lot of information for further improvements. Students liked visual appearance and the feedback and instruction given to them. Nevertheless, it has been shown, that the model needs to be very versatile to gain the acknowledgment necessary. An adaptive assessment system seems to be crucial to be used in a practical way.

# Kurzfassung

Virtuelle Lernumgebungen gewinnen immer mehr an Attraktivität. Trotzdem sind die meisten dieser Systeme nicht in der Lage, eine konstante Qualität zu gewährleisten, indem sie eine individuelle sofortige Rückmeldung entsprechend den Fähigkeiten geben. Dies gilt insbesondere für Systeme, an denen keine menschliche Tutoren direkt teilnehmen.

Automatische Bewertungssysteme können eine sofortige Rückmeldung geben, die sich im Lernprozess als sehr wichtig erwiesen hat. Darüber hinaus tragen traditionelle Feedback-Strategien wie Multiple-Choice-Tests dazu bei, das neu gewonnene Wissen zu vertiefen. Zwar wird in Bezug auf virtuelle Lernumgebungen oder Bewertungssysteme viel geforscht, doch scheint der Forschungsaufwand für die Kombination dieser beiden Bereiche ziemlich unberührt zu sein, insbesondere für Physikexperimente.

In dieser Arbeit wird ein Ansatz vorgeschlagen, um die Bereiche immersives Umfeld und Bewertungssystem zu kombinieren. Dieses Bewertungssystem ist für semantische Ereignisverfolgung geeignet und kann jederzeit mit mehreren virtuellen Umgebungen jeglicher Art kommunizieren. Als Proof-of-Concept wurde ein Pendelexperiment erstellt und das Bewertungssystem mit einem Aufgabenblatt ergänzt. Dies führt zu einer betreuten Lernumgebung für Studenten.

Um sicherzustellen, dass die Ziele erreicht wurden, wurde eine Umfrage mit 8 Kindern im Alter von 10 bis 14 durchgeführt. Die Ergebnisse waren vielversprechend und lieferten viele Informationen für weitere Verbesserungen. Die Schüler mochten das visuelle Erscheinungsbild sowie das Feedback und die Anweisungen, die sie erhielten. Es hat sich jedoch gezeigt, dass das Modell sehr vielseitig sein muss, um die notwendige Bestätigung zu erhalten. Ein adaptives Bewertungssystem scheint für die praktische Anwendung entscheidend zu sein.

# Contents

<b>Abstract</b>	<b>v</b>
<b>Kurzfassung</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Problem Definition . . . . .	2
1.3 Structure of the work . . . . .	2
<b>2 Background and Related Work</b>	<b>4</b>
2.1 E-Learning . . . . .	4
2.1.1 History of Virtual Learning Environments . . . . .	5
2.1.2 Learning Activities and E-Learning Support . . . . .	6
2.1.3 COSE . . . . .	7
2.1.4 Learning Settings for Physic Experiments . . . . .	9
2.2 Simulated Virtual Learning Environments . . . . .	10
2.2.1 Second Live . . . . .	11
2.2.2 Minecraft Education . . . . .	12
2.2.3 Open Wonderland . . . . .	13
2.2.4 Maroon based on Unity . . . . .	15
2.2.5 SAVE framework . . . . .	16
2.3 Assessment Systems . . . . .	18
2.3.1 Feedback and Learning . . . . .	19
2.3.2 Types of Assessment Systems . . . . .	19
2.3.3 Rule-Based Systems . . . . .	20
2.3.4 Example-Based Systems . . . . .	23
2.4 Summary . . . . .	26
<b>3 Requirements and Design</b>	<b>28</b>
3.1 Motivation . . . . .	28
3.2 Requirements . . . . .	28
3.3 Conceptual Architecture . . . . .	30

## Contents

3.4	Design of the Assessment System . . . . .	31
3.4.1	Feedback . . . . .	31
3.4.2	Evaluation Engine . . . . .	32
3.4.3	Assessment Model . . . . .	33
3.4.4	Web Service . . . . .	33
3.5	Design of the Immersive Environment . . . . .	34
3.5.1	Experiment Workplace . . . . .	34
3.5.2	Communication Interface . . . . .	35
3.6	Proof of Concept . . . . .	35
3.6.1	Immersive Environment . . . . .	35
3.6.2	Assessment System . . . . .	38
3.7	Summary . . . . .	41
<b>4</b>	<b>Development</b>	<b>43</b>
4.1	Pendulum Experiment . . . . .	43
4.1.1	Structure of Maroon . . . . .	43
4.1.2	Structure of the Pendulum Experiment . . . . .	44
4.1.3	Background and Environment . . . . .	45
4.1.4	Pendulum . . . . .	50
4.1.5	Stopwatch . . . . .	51
4.1.6	Calculator . . . . .	52
4.1.7	Summary . . . . .	53
4.2	Assessment System . . . . .	53
4.2.1	Evaluator . . . . .	54
4.2.2	Feedback . . . . .	58
4.2.3	Logging . . . . .	61
4.2.4	Model . . . . .	61
4.3	Usage . . . . .	63
4.3.1	Pendulum Experiment . . . . .	63
4.3.2	Unity Interface for Assessment . . . . .	65
4.3.3	Logger . . . . .	67
4.4	Summary . . . . .	69
<b>5</b>	<b>Evaluation</b>	<b>71</b>
5.1	Aim and Method . . . . .	71
5.2	Pendulum Experiment Assignment . . . . .	71
5.3	Target group . . . . .	73
5.4	Instruments and Setup . . . . .	73
5.5	Findings . . . . .	74
5.5.1	General Response . . . . .	77
5.5.2	Comparison A/B Group . . . . .	79

## Contents

5.5.3	Emerged Problems . . . . .	79
5.6	Summary . . . . .	81
<b>6</b>	<b>Lessons Learned</b>	<b>84</b>
6.1	Research . . . . .	84
6.2	Development . . . . .	84
6.3	Evaluation . . . . .	85
<b>7</b>	<b>Future Work and Summary</b>	<b>86</b>
7.1	Summary . . . . .	86
7.2	Future Work . . . . .	87
	<b>Bibliography</b>	<b>88</b>
	<b>Appendix</b>	<b>93</b>

# List of Figures

2.1	General Overview of E-Learning . . . . .	5
2.2	Architecture of Open Wonderland . . . . .	14
2.3	Conceptual Architecture of Maroon . . . . .	17
2.4	SOFIA Assessment Components . . . . .	21
3.1	Conceptual architecture of the generalized flexible assessment approach	30
3.2	Design of the Prototype . . . . .	41
4.1	UML Pendulum Experiment . . . . .	45
4.2	Pendulum Experiment . . . . .	46
4.3	GUI . . . . .	48
4.4	Assignment Sheet Prefab . . . . .	51
4.5	Basic UML of the Sofia EE Structure . . . . .	54
4.6	UML of the Engine . . . . .	55
4.7	Finish/Activate Assessment . . . . .	56
4.8	UML of the Feedback Model . . . . .	58
4.9	UML of the Model . . . . .	59
4.10	Question Assessment 2 . . . . .	60
4.11	Logger . . . . .	61
4.12	Hint Text of Assessment 1 . . . . .	62
4.13	Different Scenes during Play-Through . . . . .	64
4.14	Language Settings . . . . .	65
4.15	Assessment Components . . . . .	67
4.16	Assessment Components Time Line . . . . .	68
5.1	Overall CES Result . . . . .	76

# List of Tables

4.1	Controls . . . . .	47
4.2	Color Scheme for Feedback . . . . .	49
4.3	Inpput Fields . . . . .	60
4.4	Filter Key Words . . . . .	69
5.1	Demographic Data . . . . .	73
5.2	Free Questions . . . . .	75
5.3	Emotion Constructs . . . . .	75
5.4	Summary of SUS . . . . .	77
5.5	Positive Free Text Answers . . . . .	78
5.6	Free Measurement Results . . . . .	80
5.7	Negative Free Text Answers . . . . .	82
5.8	Suggestive Free Text Answers . . . . .	83

# 1 Introduction

## 1.1 Motivation

In secondary and tertiary schools, physics experiments always seem to fascinate students. Moreover, hands-on experiments seem to have a much deeper impact in understanding and learning effect than simple drawings. Nonetheless, some experiments tend to be far too dangerous, expensive or too time consuming to do in real life. These experiments might involve high voltages, acids or heavy machinery to name just a few. Other experiments may be about physical principals, which are not involving visible light at all, such as magnetic waves or experiments involving electrons or other very small particles.

The above stated problems can be solved by using 3D simulated environments. Especially with virtual reality, dangerous physic experiments may be executed without endangering anyone. Furthermore, abstract visualization techniques may be utilized to visualize otherwise invisible happenings. The impressiveness of VR is furthermore enhancing not only the interest of students but supporting the embedding of the learned content.

This master thesis picks up the masters thesis of Joachim Maderer (2013) by extending the proposed assessment system prototype and adjusting the virtual learning environment (VLE) "Maroon" (Pirker, Lesjak, & Guetl, 2017) to interact with it. The prototype will be extended to support whole assignments rather than single assessment per experiment. Furthermore, a questionnaire feedback system with the possibility of a sophisticated analysis system is introduced. This greatly reduces the time necessity of the experiment expert, when modeling new experiment assignments. As a last step, the code is optimized and enhanced with multiple new features, such as different mathematical operations or simple boolean comparators.

As Maroon is a multi platform VLE with multiuser support, it suggests outsourcing the assessment part to an external system. There are couple of reasons for that. Firstly, resources on the client devices are spared. This holds true especially for mobile VR devices. Secondly, when outsourcing the assessment to a separate service, not only Maroon as VLE may be utilized, but also other different Moodles or VLEs



may be assessed, as the interface of the assessment system is very generic through its XML structure. As a third reason, why to outsource the assessment system is, to provide one unique point for the assessment expert to access all the data created by the users. The educational aspect of assessment in learning can be observed through the positive results mentioned in Pirker, Lesjak, and Guetl (2017) and Maderer and Gütl (2013), if the system implements the feedback part as seamless as possible (most time in form of dialogues).

### 1.2 Problem Definition

It is a vital part of any e-learning system, to provide an assessment or even assignment component. The student needs to be able to self-evaluate his progress by receiving feedback from the system about his performance. As of right now, the prototype has the ability to assess the behavior of a student. However, to provide a working system which is not only able to assess the behavior, but also the gained knowledge, it needs to be enhanced with more functionality, such as assignments. These provide the re-usability of single assessments and the possibility of evaluating the gained knowledge with questionnaires.

Furthermore, technology has evolved since the implementation of the prototype. New Virtual learning environments arose, which means an improvement in usability and visual appearance. To enhance the intrinsic motivation of students these are key factors as well to gain acceptance for the system. This holds also true for the experiment expert, as the creation of new experiments should be as time effective as possible. That means, a change of virtual learning environment and the providence of an easy-to-use interface is one of the key problems which needs to be tackled.

As a third step, a 3D simulated environment needs to be provided. This environment should provide an as-easy-as-possible to use, feature rich possibility for the experiment expert to create new experiments. This means, the work load needs to be as little as possible by simultaneously providing the most feature rich export functionality. That should include not only desktop virtual reality, but also the communication interface for the assessment system.

### 1.3 Structure of the work

The introduction is followed by chapter 2, Background and Related Work. Here, a basic overview is given over e-learning systems and their types. Furthermore,

## 1 Introduction

their history is considered and thus their development as well as the importance is discussed. Last but not least, the five main goals for e-learning systems in general are described. Moreover, it presents different assessment systems as well as different virtual learning environments. Their similarities and differences are discussed. As a conclusion, multiple reasons are mentioned, why some of these systems seem not to be the ideal approach, and thus are not chosen.

Design and Requirements, chapter 3, states a detailed list of the requirements and explains the thus evolving conceptual architecture. However, the main focus lies on the design decisions. Accordingly, the prototype planning is discussed by choosing two frameworks and discussing the alterations needed, to achieve the discussed requirements beforehand.

The realization of this planned prototype is discussed in chapter 4, Development. A very detailed description of all classes and enhancements can be found here, as the technical details of the chosen frameworks. Last but not least, the realization of the model for the experiment is described.

As the prototype and the realized experiment needs to be evaluated, chapter 5, Evaluation, summarizes not only the findings, but also how they have been achieved. This includes the questions asked and the target group for which the experiment was made for. The evaluation methodologies are described as well as the results of the feedback and their interpretation.

The last two chapters are chapter 6 and chapter 7, which are Summary, Lessons Learned and Future Work. These recapitulate the aforementioned chapters, implementations, findings, and results. In lessons learned, take-away findings are discussed as well as problem solutions to implementation specific obstacles. Future Work shows a list of possible enhancements for the used Frameworks.

## 2 Background and Related Work

E-learning has gained outstanding importance in the last couple of years. Times seem to shift from Students, just sitting in classes at fixed time tables, to a computer-based interactive scene. In order to provide feedback and assessments to students during e-learning sessions, different kinds of feedback and assessment systems have been developed to combine the advantages of direct interaction with tutors and the freedom of time and space. The aim of this chapter is to provide an overview of the background of the proposed work, as well as show state of the art techniques for assessment systems in virtual learning environments.

To start with, a general overview of e-learning is given. Then, the principle of STEM education (Science, Technology, Engineering, and Mathematics) is introduced and approaches to integrate e-learning in this field are discussed. Moving on, Simulated Virtual Environments with focus on STEM education are explained and different approaches and categorizations are discussed with regard to their advantages and disadvantages. Last but not least, different assessment systems are introduced and the challenges involved are outlined.

### 2.1 E-Learning

The “E” in E-Learning stands for “electronic” (Zhang, Zhao, Zhou, & Nunamaker, 2004). It refers to electronic or digital media that aid in classes. This does not restrict to local or remote usage, on- or offline utilization or the type of device being used. Figure 2.1 shows an overview of the structure of the principal concept of E-Learning. This overview of concepts is inspired by combining the information of all the mentioned papers, but especially from Littlejohn and Pegler (2007), Potkonjak et al. (2016), Yamagata-Lynch (2014). The learning content, in this case STEM education can be taught either Face-to-Face or Experiment Based. These two Learning activities in combination with the STEM education content are discussed in subsection 2.1.2. When taught in a “blended” environment, synchronous and asynchronous implementations exists. Because the proposed work is focusing on asynchronous blended learning, subsection 2.1.4 compares these two techniques. Last but not least, the

## 2 Background and Related Work

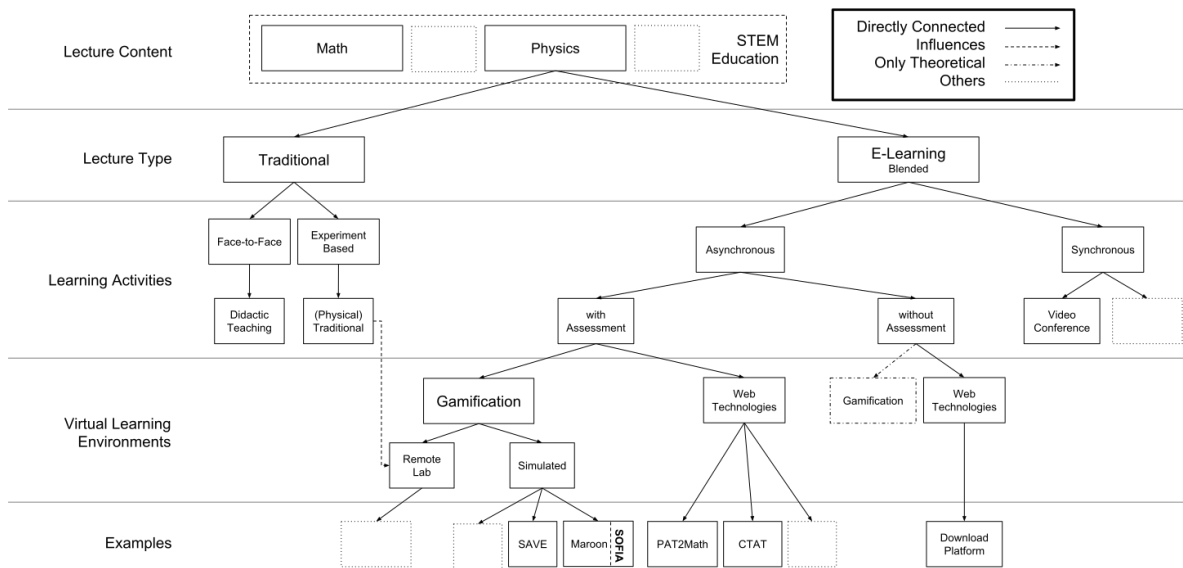


Figure 2.1: General Overview of the structure of E-Learning for this Work.

different approaches of virtual learning environments are discussed thoroughly in the following sections, starting with section 2.2.

### 2.1.1 History of Virtual Learning Environments

A very short milestone-like summary is provided, to give a glimpse over the evolution of E-Learning systems:

- **1956** First versions of virtual learning environments as we know them today (mostly relate able to today's "MOOCs") were created by the University of Houston with their Project KUHT, one of the first public television stations in Texas. This station broadcast about 38% of the courses held by professors at the university mostly at night time to provide the lessons to students, who had to work during the day (Wendt, 1956).
- **1962** One of the first on-line virtual learning environments was PLATO (Programmed Logic for Automated Teaching Operations) developed at the University of Illinois (Berkeley, McGovern, Berlin, & MacDonald, 1962). It featured on-line notes, access points for instructors to audit the students data, as well as communication. Furthermore authors were able to create new lessons – basically everything which can be found nowadays too. The transmission was mainly done over a non standardized channel which had a throughput of up

## 2 Background and Related Work

to 60 ASCII characters per second. The storage was done on film tapes and punched cards.

- **1969** start of Internet by US Department of Defense ARPANET and thus paving the way for all Internet-based and collaborative learning frameworks.
- **1991** Datacloud (Johnson-Eilola, 2005) is one of the first virtual reality, multiuser virtual learning environment with collaboration in mind.
- **1998 - now** hundreds of different open web platforms, university lecture collections for all kinds of content were created as we know them today, as for example UniSANet (King, 1998), the web platform “Moodle” (Dougiamas & Taylor, 2003) or the TU Graz Teach-Center (Scerbakov, 2015).

As the proposed work focuses on simulated 3D Environments, a selection of similar frameworks and engines are discussed in the next subsections. Before diving into this matter, the different learning activities and e-learning support will be discussed in the following section with emphasis on STEM-education and related e-learning strategies.

### 2.1.2 Learning Activities and E-Learning Support

As the focus of the proposed work is on physical experiments, it is part of the STEM education. STEM stands for *Science, Technology, Engineering and Mathematics*. Sanders (2009) describes the needed of change of STEM educational software to fit the needs of the current time. Integrative STEM education, as it is described by the author, needs to follow the criteria of the non-profit, non-governmental ABET<sup>1</sup> organization, which are: “(a) an ability to apply knowledge of mathematics, science, and engineering, (b) an ability to design and conduct experiments, as well as to analyze and interpret data, and (d) an ability to function on multidisciplinary teams” (Felder & Brent, 2003). Integrative STEM education is inherently learner-centered and implements integrative instructions. According to Hartzler (2000) this leads to thorough better results in comparison to students of traditional classes, when conducting state-wide tests. Furthermore, studies have shown that it increases interest, motivation, and self-efficacy. Multiple attempts have been made to realize these principles in simulated virtual environments.

In general, there are three main lecture types: “Traditional”, “Blended Education” and solely online based learning courses. The traditional is the didactic teaching exercised since decades: a teacher is standing in front talk and chalk while the students are listening and taking notes. Blended Education means, to use multimedia material during lectures. Dede, Eisenkraft, Frumin, and Hartley (2016) talk about

---

<sup>1</sup><https://www.abet.org/>

## 2 Background and Related Work

the different approaches and development in blended education and the usage of online courses. Furthermore, they discuss the upside and the possible advantages for teachers. Especially the usage of MOOCs (Massive Open Online Courses) is described in the paper and the positive effects emphasized. A very important term in this context as well is self-directed learning. This not only means, the learners decides on the speed and material by themselves, but the system as well adjusts accordingly to the skills of the individual person.

A basic distinction between synchronous and asynchronous e-learning can be made. Synchronous e-learning means, one or multiple learners are connected to each other in any way and have direct interaction with an instructor. The big advantage of this approach is to be able to respond immediately with feedback and adjust accordingly to the skills of the individual learner. In contrast, asynchronous means, that there is no direct interaction with the instructor. The content lies on a central point to be provided 'just-in-time'. This is inherently a big advantage of this approach, as the learner can access the content at any time (Yamagata-Lynch, 2014).

An example of a virtual learning environment is the COSE framework. It supports everything needed for collaboration. The assessment is done by a human tutor. They also create the structure of the learning content.

### 2.1.3 COSE

The COSE virtual learning environment (Stiles, 2000) was funded by the 'UK Joint Information Systems Committee' with the goal in mind to create a learner centered system. To achieve that, a thorough analysis has been undergone about why a lot of VLEs are not really supporting students but rather overwhelming them with a graphical appealing but functionality lacking system. With functionality lacking meaning basically copying all the mistakes done by typical face-to-face teaching which leaves the learner in a passive notion of absorbing without understanding. Furthermore, the focus seems to be often directed on the content rather than the outcome which leads to the fact, that learners might not be intrinsically motivated.

Stiles (2000) finds the causes of these problems mostly because of the failure of recognizing the social nature of learning. Because of that, one main goal of COSE was to include social aspects into the learning process. The second main goal was to introduce a discussion component, as it considers discourse as the primary collaborative form, according to Klemm and Snell (1996). As a third goal, multi-media-rich material should be provided but still keeping in mind, to focus the multi-media on the content requirements and not to excite with "richness".

## 2 Background and Related Work

The system itself is organized into groups. The learning material and courses are not connected but rather set into relationship with learning opportunities. These learning opportunities are managed by tutors who are forming groups and any number of subgroups to assign opportunities to the groups for the students or provide the possibility for educators to work collaboratively on the course material. The content creation is handled via a versioning and monitoring system. If an author creates new material it is set to the state "Work-in-progress" and needs to be reviewed by a so called "Supertutor". If the work is accepted the state changes to "Published", otherwise, the author has to redo the content and thus automatically creates a new version without changing the original. Only published material can be used for learning opportunities (Stiles, 2000).

Students inside a subgroup or group can communicate with each other or the assigned tutor via notice boards and email directly inside the System. Furthermore, the system provides three techniques for tutors to support the student: coaching (prompts and hints for guidance), scaffolding (learner adapted intensity of guidance) and fading (reducing the guidance). Primarily, to achieve the learner based system concept, the learner can organize the learning material in any way as pleased. This constructs can be placed into context with other provided resources in the system and reused by other tutors or students (Stiles, 2000).

The COSE system focuses primarily on the "how" of representing the content and trying to implement a new way of providing learning material. It seems to be a quite unique approach on virtual learning environments for problem-based learning. The collaborative aspect is well implemented as well as the learner based focus, although the system seems to lack some features needed for the defined requirements of this work:

Firstly, the graphical system seems to be very simple with a tree view of all the structured learning material on the left hand side and the actual content on the right hand side, as can be found in Windows Explorer for example. Furthermore, it is structured with tabs for communication, organization, creation to name just a few. This approach might seem effective for organizing data, but is definitely not usable to represent real world experiments.

Secondly, the assessment part, although taken into consideration, is completely done by a human which raises the problem of real individual assessment during an experiment or assignment. Learner based assessment and adaption to the skill of the learner is discussed, the realization of this very function seems to be done manually though. There is no automatic or intelligent mechanism supporting the tutor on this behalf. With that said, this system definitely gave a very good insight of how to achieve a learner based virtual learning environment which will be taken into

consideration during the creation of the showcase model of the defined requirements for this work.

### 2.1.4 Learning Settings for Physic Experiments

When focusing in particular on physic based experiments, three different approaches can be found on how to realize them and how to interact with it. The following list summarizes and compares these different approaches:

- **Physical:** This is the typical set up, where the experiment is physically assembled in real life. The advantages are obviously full realism and accuracy. However, the big downsides are that there is no other support and assessment except of the teacher. Furthermore, most times schools cannot provide resources such that every student can work on the experiment individually. That is because the materials needed can be very expensive, or would take up too much space. Moreover, it is hard for a teacher to assess multiple groups of students at once and provide individual feedback for every single person. These disadvantages are the reason for implementing physical experiments in virtual reality with assessment components (Gustavsson, Nilsson, & Lagö, 2009).
- **Remote Labs:** With the use of the Internet, financial support for laboratories declined because the access to physical labs was made easier then ever. This has been achieved by providing interfaces such that students are able to interact with a physical experiment assembly by using a interface provided via a remote server. There are multiple studies showing the qualitative equivalence to hands-on experiment settings, but basically only one problem mentioned before is tackled: reduced costs per individual person (see Corter, Nickerson, Esche, & Chassapis, 2004; Scanlon, Morris, Paolo, & Cooper, 2002; Sicker, Lookabaugh, Santos, & Barnes, 2005)
- **Simulated:** These types of labs typically tackle most of the before mentioned problems: The actual experiment is completely simulated and sometimes augmented with additional information on current states but most times do not provide any assessment or assignment feature. Obvious advantages are, that there are no additional costs with increasing number of users, which means every student is able to utilize the lab as long as a computer is provided. Furthermore, even dangerous and expensive experiments can be executed by anyone. A very famous example are the PhET Simulations (Wieman, Adams, & Perkins, 2008) of the University of Colorado.



## 2.2 Simulated Virtual Learning Environments

As the primary focus of the proposed work is on simulated laboratories, the different techniques used to represent them are discussed in more detail. Basically, there are basically three technologies to represent a laboratory:

- **Augmented Reality:** Augmented or Mixed Reality is achieved by special Headsets, with one of the most prominent example being the Microsoft HoloLens (Kress & Cummings, 2017). These headsets enhance the real world image with simulated elements. This is achieved by measuring and recreating the current surrounding on the fly. With this information augmented 3D objects may be placed context sensitive in the environment. With that additional information, simulate furniture or a simulated object may be represented to the user. As the current position in relation to these virtual object is tracked, extreme immersion can be achieved. Utilizing this technique for physic experiments results in a mixture between hands-on physical experiments and simulated experiments. Kaufmann and Meyer (2008) show a study on how to use this technique for physics experiments and foresee a promising future. Current development in the Maroon framework addresses this approach too, although there are no publications out yet. Head Mounted Display suffer one big issue, which is very hard to overcome. The field-of-view must be kept low due to strong distortion according to the displacement of head- and eye- direction. This leads to a very narrow productive window (Azuma, 1997).
- **Virtual Reality Headsets:** With VR Headsets like the Oculus Rift (DJSCOE, Vile - Parle (W), Mumbai, Rajesh Desai, Nikhil Desai, Deepak Ajmera, & Mehta, 2014) or the HTC Vive (Dempsey, 2016), a full immersion into virtual reality can be achieved. The headset completely covers the eyes with a monitor, in contrast to augmented reality headsets. Two special lenses inside the headset, as well as a special rendering technique distort the image so it feels naturally sound and covers even the eye movement inside the headset. Most times, the environment is interacted with two controllers and their multiple buttons and track pads. Because of the wide field-of-view and the rather good image quality, not to mention the exceptional precise input mappings, it is a unique way to simulate any experiment and create full immersion and "hands-on-feeling". Beside full fledged devices like the HTC Vive with dependence on a computer, there are implementations which utilize the processing power of mobile devices for this purpose. Prominent examples are Samsungs GearVR (Moro, Stromberga, & Stirling, 2017) and Google Cardboard (MacIsaac, 2015). As nowadays flagship mobile phones provide resolutions up to 440 PPI, these devices gain more and more potential. Furthermore, due to the built in accelerometer in every phone a surprisingly accurate movement detection of the user is possible.

## 2 Background and Related Work

One of the biggest downside for this devices is, that quite a lot of people seem to have problems with a special motion sickness, as the body undergoes different external influences than the eyes report to the brain. This is especially a problem at mobile VR, as the processing power of mobile phones is often not sufficient to render more than 60 frames per second, whereas 120 frames are recommended for a smooth experience (Jing, Behzad, Jeff H, Sarah Farukhi, & S Khizer, 2018).

- **3D Simulations for Desktop:** 3D Simulation for desktop computers is basically the same technique as in VR or AR headsets. The main difference is the render technique, as it is optimized for desktop or laptop computers. Thus, it is very probably the most wide spread technology with the nearly uncountable amount of 3D games and software available. With desktop computer targeted 3D simulation it is possible, especially since the technical evolution of the last few years, to use nearly any device to simulate a virtual environment. Nearly all devices provide a dedicated graphic processing unit (GPU) to efficiently represent the animated objects. This makes it particularly interesting as the accessibility is very high. Unity utilizes the fact of similarity quite well by providing the possibility to export to any device: stand-alone desktop application, web-based, VR, AR and mobile VR (performance optimized), to name just a few (Pirker, Lesjak, & Guetl, 2017).

The proposed work focuses on desktop targeted 3D simulation and thus discusses different approaches in this matter. The goal is, to create a stand-alone desktop application which simulates physical experiments which are enhanced with feedback of an assessment system. Because of that, four VLEs are introduced, two of which already implement an assessment and feedback component, which will be discussed in the following sections.

### 2.2.1 Second Live

Warburton (2009) describes the wide range of educational applications implemented in Second Life. It is built up in a client-server infrastructure. The rendering is done on the client side, without storing pre-rendered object in the memory, which leads to great customization capabilities at the cost of performance. The focus of this framework is massive multi-player and cooperation. Furthermore, it provides full user freedom in terms of creation and modeling of the own character as well as the environment.

There are multiple advantages of using Second Live as a educational framework, for example (Warburton, 2009):

## 2 Background and Related Work

- Rich social interaction possibilities
- great immersion due to realistic environment and free character model customizability
- big community in either cooperation as well as content creation

But comes with some disadvantages(Warburton, 2009):

- One of the biggest problems is efficiency and performance, as everything needs to be sent to the client and rendered locally. This comes with high cost on bandwidth and hardware usage
- as in every virtual environment with free selectable characters, the correctness of data personal data is not and can not be enforced. This can lead to high confusion.
- The freedom of (nearly) unrestricted alteration of the environment for every player very often leads to so called grieving, a process where players intentionally destroy the doings of other players. Mostly, creating something new is very time consuming, which leads to the fact that a lot of time is lost because of that.

According to Baker, Wentz, and Woods (2009) over 100 University of the United States use Second Life for courses, learning content access or live streams. An example is the University of Derby which holds psychology courses as well as a virtual laboratory called “virtual hallucinations project” to teach several aspects of schizophrenia.

In a similar manor, Minecraft, a very popular game, is also used in education. It is mostly used in environmental awareness, storytelling and programming skills. The next section introduces this game in more detail and the utilization for education.

### 2.2.2 Minecraft Education

Minecraft<sup>2</sup> is often compared to a “digital Lego” game. In the base game, the user is able to place different blocks of different material anyway wanted. A very important introduced block is so called “red stone” and corresponding acting blocks like lights, pistons, extenders and code blocks. With these blocks, it is possible to create logical wirings in-game and thus move blocks, light up rooms or even execute in-game server related commands (Nebel, Schneider, & Rey, 2016).

For education, Minecraft has been used already in its early stages for spatial geometry or social skills. For experiment based learning, structural integrity was introduced

---

<sup>2</sup><https://education.minecraft.net>

## 2 Background and Related Work

and represented by different colored blocks. Often, the collaboration component has the main focus. Very often, the nonetheless quite copious game is very often extended with modifications (“mods”) to provide additional features, such as programming blocks as can be found in scratch (Nebel et al., 2016).

There are some limitations though (Nebel et al., 2016):

- Experienced players, as the game is available for a very small price for everyone, can outperform unexperienced players a lot in competitive tasks
- Although basically easy to use, the learning curve of mastering the red stone logical constructions is quite steep and not easy to master for everyone
- Architectural correct lightning or structures are not achievable with the default game, as the lightning calculations are quite simplified and structural integrity does not exist at all
- the main idea of the game is, that goals are self-set, which means there are no limitations or goals set per default. This leads to the problem, that the actions taken by the student need to be very carefully supervised, such that the students stay on the teachers intended

Minecraft is utilized in multiple fields of studies like biology, ecology, physics, chemistry and geography. An outstanding example for ecology is the ‘Tragedy of the commons’- map which teaches students the principle of sustainability of a society. For chemistry, a free modification is available called ‘MineChem’ to explore chemical elements in more detail (Short, 2012).

The last two applications (Second Live and Minecraft for education) both evolved from a game not directly intended to be educational platforms but rather entertaining computer games. In contrast, there are applications and/or frameworks, especially developed to be used as an educational software. Open Wonderland is one example. It was the framework of choice by Maderer (2013) and will be discussed in the following chapter.

### 2.2.3 Open Wonderland

Open Wonderland is a virtual learning environment developed with three main goals in mind (Kaplan & Yankelovich, 2011):

- Synchronous Collaboration  
To achieve this goal multiple research has been done to find out, what really supports an immersive collaboration. According to Yankelovich et al. (2004) especially audio is extremely important to support collaborative usability. Because of that, the 3D space is taken into account too, to organize conversations,

## 2 Background and Related Work

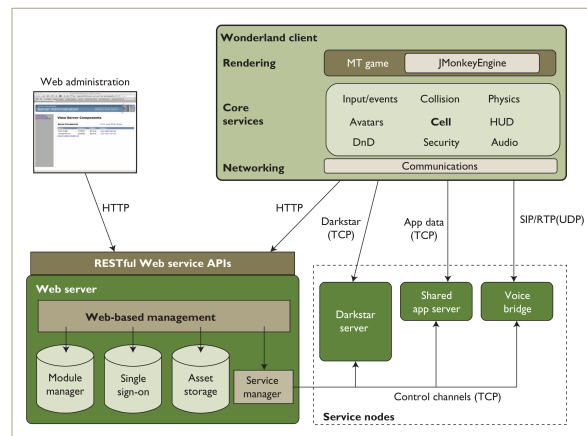


Figure 2.2: The Architecture of Open Wonderland (Kaplan & Yankelovich, 2011).

which means proximity to each other is a factor of hearing the conversation of others. Last but not least, security was taken into consideration too with unauthorized users not being able to edit documents or disrupt ongoing work of other groups.

- **Extensibility**

Extensibility primarily means to enhance the system with more features easily without changing the core code. This was achieved on multiple levels, by adding the possibility to extend the authentication methods and more game objects like avatars, sounds, menu options and shared application. To achieve this goal while keeping the system simple at the same time, the application was split up into multiple small form factor servers, as described in the next point.

- **Federalization of Infrastructure** The third important goal was to be able to spread the workload onto multiple different servers if needed by creating a heterogeneous server infrastructure consisting of multiple small form factor servers with very specific tasks, like the voice bridge, shared app server or the Darkstar server (a project developed by Sun under the name RedDwarf, a gaming server with low latency). Further details about the structure of Open Wonderland can be observed in Figure 2.2.

Another interesting feature of Open Wonderland is, that it supports legacy 2D X11 applications inside the 3D environment. This is achieved as every object in the scenery is organized into a so called cell which inherits from a base cell object and follows the purpose of rendering one specific item. Furthermore, the cells can be saved as XML, which renders it especially useful for user-created objects.

Although the chosen architecture seems promising, there are multiple trade offs

## 2 Background and Related Work

discussed as well. Besides the reasonable fact that there needs to be a trade off between scalability and intractability (as a largely dynamic systems needs far more updates for all users than a highly static one), one of the biggest down sides of the system might be the mentioned point, that the spreading of workload over multiple different servers means also spreading the code. This leads to a certain degree of overview loss of the code in comparison to a core-implemented experiment.

“Maroon”, implemented in the game engine “Unity”, tries to overcome this problem, mostly by being much more specific for physic related experiments. Unity is an open game engine which was used to develop the Maroon framework. Maroon was designed with scalability and compatibility in mind. The game engine as well as the framework are discussed in the next section.

### 2.2.4 Maroon based on Unity

Maroon (Pirker, Lesjak, & Guetl, 2017) is implemented in Unity (Haas, 2014; see Unity, n.d.), a powerful 3D game engine, with capabilities to export the application in multi platform manner like PC, Web, Android, iPhone and even Virtual Reality headsets like Oculus Rift (DJSCOE, Vile - Parle (W), Mumbai et al., 2014) and HTC Vive (Dempsey, 2016) as well as mixed reality devices like Microsoft HoloLense (Kress & Cummings, 2017). Unity provides a lot of useful features, like concept of mass and gravitation, support for GUI interfaces and user defined C# scripting to name just a few. This leads to the fact, that a lot of necessary components for the prototype are already provided by the game engine itself.

As Unity uses C# as its back end language, it works perfectly with the current prototype of the assessment service. To enhance usability for the experiment expert, an interface will be created, which does provide helper functions for an easy-to-use communication with the assessment system. This interface will be integrated into the experiment.

Maroon emerged because of a study on if and how a virtual learning environment can provide support for physics experiments, mostly because STEM (science, technology, engineering, and mathematics) in the usual class room routine are often described as boring. Thus, the goal with Maroon is to provide an extensible multi-platform multi-user laboratory to enhance and deepening learning material from school with the technique of virtual reality immersion. Maroon was also optimized to be used on a desktop basis, Web Based as well as Headset VR and Mobile VR devices (Pirker, Lesjak, & Guetl, 2017).

## 2 Background and Related Work

The extensibility is provided by structuring every experiment in room a separate room. By starting every experiment separately the upside of performance gain emerges. This is especially useful in a virtual reality set up. Figure 2.3 shows the conceptual architecture of Maroon. With this architecture in mind, a system with as much real live handling as possible has been created to deliver an immersive surrounding for physics experiments. The major advantages of virtualization of physics experiments are that firstly, dangerous experiments can be executed without any risk, secondly, otherwise invisible phenomenon can be visualized and thirdly additional hints may be provided through animations and graphical representations (Pirker, 2017). An example for these techniques is the falling coil experiment, where the magnetic fields and interaction between magnet and coil during the fall is animated with arrows and lines. Furthermore, the “van de Graaf” experiments involving Tesla coils are an excellent example of experiments, which would be too dangerous and costly to execute in real life.

Multiple studies have shown, that virtualization of STEM experiments indeed shows positive results. The survey performed on Maroon resulted in very promising feedback and overall very positive response (Pirker, 2017; Pirker, Lesjak, & Guetl, 2017; Pirker, Lesjak, & Gütl, 2017; see also Settgast, Pirker, Lontschar, Maggale, & Gütl, 2016). Especially (Pirker, 2017) describes very positive results with Maroon, especially in combination with a VR Headset. This is especially interesting, as it is stated that it is one of the first surveys taken for physics experiments in combination with Virtual Reality Headsets.

### 2.2.5 SAVE framework

Greuel, Myers, Denker, and Gervasio (2016) describe the SAVE framework, which supports the creation and learning of procedural tasks. The main goal was to create an environment which supports the learner as well as the content creator to define procedural tasks. The framework aims for an easy-to-use approach such that domain experts rather than content creators are able to define even complex tasks.

This is achieved by extending the 3D model of an object with annotations stored separately to describe the object in the necessary detail. This leads to the fact that the ontology of the object can be transferred to other 3D models (for example higher detail model). The implementation was done in a stand alone application which seems to lack the features and possibilities known in video games (Greuel et al., 2016).

To define the procedural task, the expert is recording the actions needed in the program to define a base model of the task. This task may then be extended in

## 2 Background and Related Work

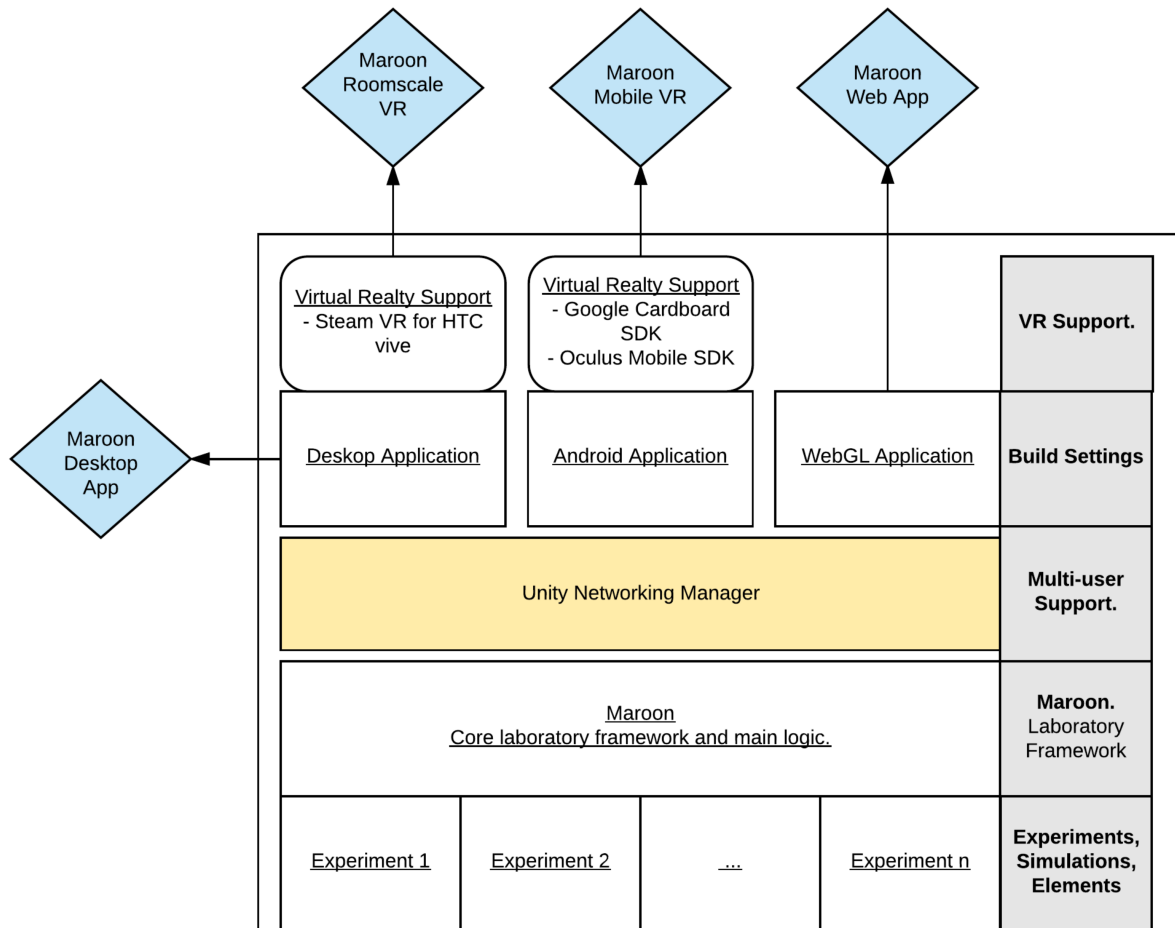


Figure 2.3: Conceptual Architecture of Maroon (Pirker, 2017).



## 2 Background and Related Work

a separate graphical editor to add annotations and possible different approaches (such as the possibility of performing certain steps in a different order). From this definition a user interaction graph can be calculated. The user executed steps are also translated into a graph and matched to the model. The graph with the lowest distance cost is considered to be the path taken by the user, from which the feedback may be generated (Alevan, McLaren, Sewall, & Koedinger, 2009).

Although the implementation, description and screen shots look promising, there seems to be multiple downsides:

Firstly, despite the fact, that all object are semantically enhanced and thus can be used in the editor to create the program flow, the approach seems rather stiff in view of the possibilities, which types of learning methodologies can be modeled. The examples shown in the paper tend to be rather linear, as the assembly of a machine gun or a cooking receipt is a fixed, barely changeable process. For these kind of problems, this approach seems to be well working. To model problems which do not have a fixed outcome, the work for the expert seems to be rather difficult or tedious. This assumption is mainly supported by the fact, that all parametric values are coming from the semantically enhanced 3D object and thus are only discrete.

Secondly, hand-in-hand with the first problem, there seems to be no possibility to perform calculations inside the assessment model. This leads to the problem, that a continuous problem can neither be checked by a equal-with-delta approach, nor can be checked, if an action performed by the user was theoretically correct, backed by a formula influenced by the settings of the user.

Thirdly, the GUI the user is using seems to be particularly stiff, as it is directly implemented into the assessment system. This leads to not only the problem, that the view seems to be quite hardly separable from the assessment system, but rather multi-user assessment seems to be rather difficult to implement.

Because of these four reasons (discrete editor, no calculations, no multi-user, no easy adoptable view) the approach of the SAVE framework seems not to be suitable for the discussed goals and requirements in chapter 3.

### 2.3 Assessment Systems

Since the focus of the proposed work is on asynchronous blended virtual learning environments, the need on an assessment system arises which is able to provide immediate feedback to the student. Furthermore, the feedback should be able to adapt to the users needs to provide just enough feedback to help but not enough to

spoiler. At first, an overview of the pedagogical impact of feedback during learning is shown, as well as the two major methods of assessments. In the following subsections, the two main types of assessment systems are discussed. These two types, namely rule-based and example-based are thoroughly discussed and examples are shown.

### 2.3.1 Feedback and Learning

To examine and support students during learning, two main methods are distinguished:

- **Knowledge-Based:** Knowledge-Based feedback aims to support during the knowledge-gain phase. That means, this method is used to learn something completely new or check for gained knowledge (DiSessa, 1982).
- **Skill-Based:** If gained already known knowledge should be accessed and used in a practical manor, this method comes into play. Students are confronted with a practical example which should be exercised. The aim of this method is deepening of already existing knowledge (Collis, De Boer, & Slotman, 2001).

This work focuses on the skill-based approach, in particular on problem solving. For skill based based learning, it is crucial to provide constant feedback. Collis et al. (2001) emphasizes the importance of feedback to enforce self directed learning for students. This primarily engages learners to participate and thus creates excitement for the topic.

### 2.3.2 Types of Assessment Systems

Within skill-based assessment systems, mainly two types of implementation can be found. On top of that, two different realizations are shown. This leads to four different approaches, as every permutation may be implemented:

- **Internal assessment system:** The Assessment system is implemented directly into application. Most times, this leads to a very specialized assessment system which is just able to assess one kind of problem. (see Greuel et al., 2016; Heffernan & Heffernan, 2014; Jaques et al., 2013, for the ASSISTment, PAT2Math and SAVE Frameworks respectively).
- **External assessment system:** When the Assessment System is decoupled from the virtual learning environment, it is called external assessment system. This technique has the positive impact, that multiple different systems may be assessed at the same time. This technique is further discussed by Alevan, McLaren, and Sewall (2009) and Maderer and Gütl (2013).

Beside the type of implementation, there are two main techniques:

- **Rule-Based systems:** These systems describe the model which describes the experiment with conditionals. If the conditions are met, a consequences block is executed. To improve performance, there are several techniques to determine, when to suspend rules to counter weight avoidable interpretations of rules (Jaques et al., 2013)
- **Example-Based Systems:** Example-based systems are created by a play-through from the expert. Because of that, a graph representation is created of the model. Most times, this model can then be modified and generalized (Aleven, McLaren, & Sewall, 2009)

This leads to four different possible implementations of assessment systems. Examples of all of them are discussed in the following subsections, grouped by their technique. For rule-based systems these are SOFIA (external) and PAT2Math (internal), for example-based CTAT (external) and ASSISTment (internal).

### 2.3.3 Rule-Based Systems

As described in subsection 2.3.2 discussed, rule-based systems are created by defining conditionals which invoke a consequences block if fulfilled. These systems often are optimized by suspending rules which are not able to be fulfilled. Because of that, performance can be drastically increased. The down side of these systems is mostly, that it is a time consuming process to define the model. Because of that, most systems provide an editor, to decrease that time significantly (Jaques et al., 2013; Maderer & Gütl, 2013).

The chosen framework of the proposed work is SOFIA (Maderer & Gütl, 2013; Al-Smadi, Wesiak, & Guetl, 2012). It was picked up by Maderer (2013) and showcased with Open Wonderland (see subsection 2.2.3). SOFIA is an external rule-based assessment system, discussed in the next section.

### SOFIA

As SOFIA (Maderer & Gütl, 2013) was the chosen framework for the proposed work, it will be discussed in more detail. In contrast to CTAT and PAT2Math, SOFIA follows the “Service-Oriented Flexible and Interoperable e-Assessment” approach as mentioned by Al-Smadi et al. (2012) (hence the name SOFIA). The architectural concept can be seen in Figure 2.4.

## 2 Background and Related Work

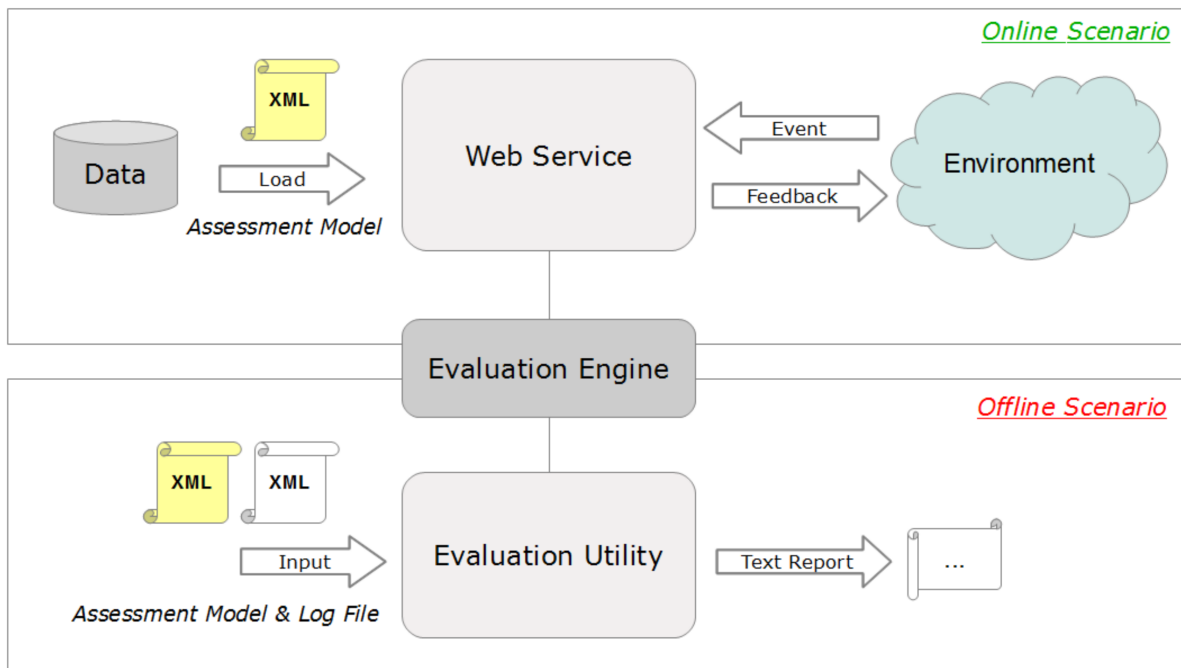


Figure 2.4: The conceptual architecture of SOFIA (Maderer, 2013).

SOFIA utilizes a modular built up, which allows to exchange Model as well as the Interpreter for the Model by using the “Visitor pattern” (Gamma, 1995, see). This allows for a very easy extendability and ingenious simple adaptability to nearly any experiment and experiment type imaginable. The model is structured with multiple behavior patterns, which are checked in the particular order they are stated. A behavior pattern consists of two parts: a conditional and a consequences part (Maderer & Gütl, 2013).

The conditional part may be any number of “match” operations (match-action, match-condition, match-group) to define when and if the consequences should be triggered. By this technique it is, for example, possible to chain events together by setting the condition to match a variable to a certain value, which is set by another action in the consequences part. The consequences part does provide a variety of possibilities as well. Beside providing a conditional block as well to even further add conditionality actions or values of variables, the acquiring of any value can be done with a huge variety of operations, such as reading variables, all sorts of mathematical operations or logical operations. Furthermore, feedback can be sent which provides the interface for communication if the virtual learning environment. At the moment, only a Text Feedback is present, which does provide the type “question” though, but it is not sufficient to provide a whole assignment possibility by itself (Maderer, 2013).

## 2 Background and Related Work

The implemented showcase has been realized with Open Wonderland (see subsection 2.2.3). In this showcase a simple pendulum has been realized and tested. The task for a student was to measure the frequency of a given pendulum by taking the time of how long a period takes. Maderer (2013) discusses about the findings and problems with this prototype. To mention just one example of the highlighted points was the missing calculator inside the scenario. These points are taken into consideration in the proposed work. Nevertheless, one of the goals is to remodel this prototype experiment in Maroon and add additional features to it, like the calculator or the discussed assignment part, mentioned in the outlook section of the paper.

This proposed work picks up the concept and prototype of Maderer (2013), enhances it with the features discussed in chapter 3 and adapts it to the new virtual learning environment “Maroon”. Some described mathematical operations are missing in the actual implementation of the prototype, as well as there are some minor bugs present which need to be addressed. Nevertheless, other approaches give good insights on which problems might occur and how to tackle them. Because of that, other frameworks, like PAT2Math, are discussed. PAT2Math was created to help students with mathematical equations by providing step-by-step feedback during solving an equation.

### **PAT2Math**

PAT2math is an Intelligent Tutoring System (ITS) which supports students at solving equations and text exercises. It was developed in Brazil since the early 2000 and has been tested thoroughly up to now. Vanlehn (2006) describes the detailed structure of the system. The basic principle although is an outer loop which chooses the exercise according to the skill of the student, and an inner loop which overwatches the student input (called student model) and compares it to a reference model created by the expert system (ES) model (Jaques et al., 2013).

The expert system is a rule-based system which acts according to two main steps as described in Polya (2003): Firstly, the Problem Representation step is performed. This step transforms the textual description of the algebraic problem into a mathematical equation. Secondly, the Problem Solution step, as the name infers, is the process to solve the derived algebraic equation. Jaques et al. (2013) split actions for the purpose of inferring the algebraic equation as well as the solution into Knowledge Components. These components describe simple inferences, such as addition or multiplication on both sides of the equation.

A tree-based data representation was used to represent the equation in the expert system. In order to find which action were taken by the student, a step generator

reproduces all possible steps and then tries to find the steps taken by the student through a model tracing technique. This so called step analyzer tries to infer a linked list between the current equation in the step and the student model. If the result is a 0 length list, the steps are the same. As this is achieved with backtracking, a meta-rule system was introduced to greatly reduce the complexity of the algorithm. This meta-rule system works by not considering all possible steps but rather only meaningful steps which could have been taken by the student (Jaques et al., 2013).

The expert system of PAT2Math and thus the feedback component seem to be very well optimized to provide fast and reasonable response to the inputs of the user. Nevertheless, for the proposed system this is not enough, as it lacks the possibility of adjusting to other environments than mathematical equations. It seems to be of unreasonable amount of work to adjust this system to a 3D environment and assessing user behavior. With that said, the adjustment of the difficulty and amount of feedback according to the skill of the user will be kept in mind during the development of the model.

### 2.3.4 Example-Based Systems

In contrast to rule-based systems (see subsection 2.3.3) example-based systems describe the model through a graph representation. This means, that an example play-through is performed by the experiment expert, which is recorded in a step-by-step state machine. This “straight line” of intermediate actions can then be parameterized and edited to create multiple different possible paths, beside the one which was played through. Furthermore, the feedback is attached to every node. This model is then interpreted by a state-machine-similar implementation to check the students doings and provide feedback (Alevén, McLaren, et al., 2009). The following sections discuss different frameworks utilizing this approach.

#### CTAT

The CTAT framework (Alevén, McLaren, & Sewall, 2009), also known as Mathtutor, is a closed source interactive tutoring system for Mathematical problems as well as the afore mentioned PAT2Math system (section 2.3.3). Mathtutor is a freely accessible web page with performance in mind. The main difference between the CTAT and the PAT2Math framework is, that CTAT is an example-based system in contrast to the rule-based system PAT2Math. That means, as described in the SAVE framework subsection 2.2.5, that the experiment expert defines the experiment by executing it himself. This results in a graph representation of the actions that have

## 2 Background and Related Work

been performed during the experiment. The graph can be edited to redefine values to parameters or set relative values instead of absolute. Moreover, it provides the possibility to define multiple possibilities to solve the problem (Alevén, McLaren, & Sewall, 2009).

The most differencing part of CTAT framework although happens in the design choices of implementation. Firstly, as in SOFIA (see section 2.3.3), the choice was made to use a server-client architecture, whereas the client is handled as a thin-client. That means, that there is no real logic executed on the client, only on the server. The communication is realized as a socket with small messages being sent. There are multiple advantages listed, such as:

- Multi-User support with no restriction to the used front-end technologies
- Minimum resource requirements on the client side, which means possible performance gain for the GUI
- Easy extendability of the system itself with other external programs such as natural language processors or image analyzer

Also, first tests have shown, that the server-client communication with 60 simultaneously working clients had no real performance impact on neither the server nor the clients. With that said, stress tests have not been performed up to now, but it is believed to be a suitable approach for even unreasonable amounts of clients and workload (Alevén, McLaren, & Sewall, 2009). The realization of the system itself follows the same strategy as the PAT2Math framework, as both follow the ITS definition of Vanlehn (2006). That means, that this system also is built up with an inner and outer loop, whereas the inner loop performs the immediate feedback component and thus influences the outer loop, which chooses the next task.

As the basic understanding and underlying architecture is the same in CTAT and PAT2Math, the way of resolving the steps of the student and comparing it to the target model are basically similar, which is why it is not discussed in detail anymore. Although, a key difference is, that the example based approach of CTAT makes it a bit easier to resolve the steps: if a student chooses one of the possible multiple paths to solve the problem, the system does not accept a change of strategy anymore. This means in addition that all correctly performed and accepted steps remain accepted until the task is done. This automatically leads to the fact, that only the correct feedback and hints are shown, as they are attached to a step of the task (Alevén, McLaren, & Sewall, 2009).

As the PAT2Math system, CTAT is a very well implemented and performance optimized system for mathematical problems only. The fact, that the client-server-based approach seems to work well is definitely an important point for the proposed work. As for the system and its approach it is not feasible to adapt it to a 3D environment

## 2 Background and Related Work

and use it for physical problems, as it just assesses mathematical calculations, not distinct doings of the student. The fact, that the server based approach works well with immediate feedback is an important factor for the realization for the proposed work.

### **ASSISTment Ecosystem**

Heffernan and Heffernan (2014) describe the implementation and utilization of the ASSISTment Ecosystem, a very well known, and broadly used system in the United States. Its goal is to support students in different algebraic problems and is based on SQL-Tutor (Mitrovic, 1997), which was created to analyze the submissions of students to check their answers.

Over the years the system evolved to an on-the-fly assessment system. It is called ecosystem as teachers are directly involved in the building process of the system. Hence, teachers are using a built-in tool called "Skill Builder" to create the rule-based model (see Ohlsson, 1994). It has been shown, that creating one hour of content for the student, at least 100 hours of a very skilled LISP constraint programmer and expert of the topic is needed. The provided Skill Builder reduces these costs significantly (Heffernan & Heffernan, 2014).

With a model of that size, performance seemed to be of great concern as well. Ohlsson (1994) (as cited by Heffernan & Heffernan, 2014) describes, that relevance settings are used on the rules itself to improve performance, as not all of them need to be evaluated all the time. This is achieved by compiling all rules into two networks on start: a relevance and a satisfaction network, based on Mitrovic (1997) approach. The study showed, that only about 20% of the rules need to be permanently evaluated, whereas all others are dependent on external influences.

As an additional feature, the system records a history of the actions and rules performed by each student. It measures how often a student violates or fulfills a rule. With this information, it calculates how relevant a rule is for each user, and assigns a "relevant" indicator to the rule. (Mitrovic & Ohlsson, 1999). Although the ASSISTment Ecosystem proved to be very effective and very well implemented, it does not completely fulfill the requirements for the proposed work. Especially because the feedback is not provided immediately, but when requested by the student. Thus, it does not have an immediate feedback component.

Instead of creating a complete assessment system, it drifted off to being a question-answer system with feedback functionality. Its goal is to improve the solution finding for students. It relies on a very big catalog of questions with possible answers and their feedback responses. These are the reasons why it seems rather unfeasible



to utilize this system to assess a student in virtual reality or a virtual learning environment. Especially tracking of actions and providing immediate feedback unintuitive to model in such a system.

### 2.4 Summary

E-Learning systems seem to be a topic of discussion and research since computers and especially the world wide web evolved. There are hundreds of different systems with dozens of techniques to be found, public accessible via the Internet and also in research. Particularly interesting are these systems, which provide motivating techniques for the student, like virtual reality, smart assessment systems and the possibility to motivate by providing manageable yet interesting topics to be discovered.

Especially for STEM education and in particular physics experiments they carry a huge potential. The possibility to enhance simulated physical facts with additional information and provide a personal tutor for every student creates high motivation for that topic. Multiple 3D virtual learning environments have been introduced. Two of them, namely Second Life and Minecraft, are massive open online games used for education. Both of them introduce extensions to better adapt to the course material, with Second Life showing much more potential in that regard.

In contrast to these two approaches Open Wonderland and Maroon were specifically developed to provide collaboration on experiments in a room spaced virtual world. Open Wonderland is developed in Java with multiple libraries in the background to provide seamless integration and utilization of multiple servers. Maroon on the other hand is implemented in Unity and thus C# and does not need a server. It runs autonomously on the client.

As its own topic, the SAVE framework was introduced as well. As a 3D virtual learning environment with its internal example-based assessment system it is one of few examples. Its main motivation is to provide an editor for objects and their meta classes to address them in the model-graph. As the research is motivated by the U.S. Military, the showcase was the assembly of a M4 rifle.

Except of the SAVE framework, none of the discussed virtual environment systems provide an intelligent system for tutoring and assessment of the student. Because of that, different approaches on assessment systems have been discussed too, with the two main techniques “rule-based” and “example-based”. Examples for example-based systems are the ASSISTment Ecosystem, a very well tested and deployed system with an extremely broad user audience and lifetime record. It has evolved

## 2 Background and Related Work

to a very basic nonetheless very well implemented and deployed question-answer system with simple feedback on a web based front-end. The ASSISTment Ecosystem is an internal implemented assessment system.

The second introduced example-based system was the CTAT framework. The conceptual architecture follows the same idea as of the PAT2Math system, but with two main differences: it is an externally implemented system and it is, as already mentioned, example-based. The goal of the CTAT framework is to provide an open-source alternative to PAT2Math.

PAT2Math, one of the two introduces rule-based system, is an internally implemented rule-based system to help students to solve mathematical equations. Last but not least, SOFIA was discussed. It was introduced as an external rule-based system but with a very sophisticated yet easy-to-use implementation, which allows for simple extendability as well as adaption to various types of experiments. It communicates with any external System (which is referred to the GUI in this work) via an XML interface over a web service, which provides incredible adaptability to other systems.

## 3 Requirements and Design

The aim of this chapter is to define the requirements in detail. Accordingly, the conceptual architecture may be planned, which is realized in more detail in the design definition. All these decisions are backed up by the findings described in chapter 2. The final step is to design a working prototype and thus choose appropriate frameworks to accomplish all goals desired.

### 3.1 Motivation

This work is motivated by two main factors. Firstly, the experiment described by Maderer and Gütl (2013) needs to be reassembled. The experiment is a simple pendulum experiment, which means, a pendulum, a stopwatch and a calculator need to be modeled in VR. The Student should measure one period and calculate the correct frequency, as well as find out about the relation between the rope length and mass of the pendulum and the resulting frequency.

Secondly, the assessment system should be as generic as possible, whereas maintaining consistency as good as possible. It needs to provide the necessary tools, to be able to model any experiment which might come in the future. This does not only involve the interface from the experiment to the assessment system, but also the core of the assessment system itself. This means, that the assessment system needs to be able to interact with any number of virtual environments at any given time possible. There should be no differentiation between type or platform.

Considering the knowledge gained during the research survey described in chapter 2 and keeping the motivation in mind, the requirements can be defined.

### 3.2 Requirements

Based on the literature survey and the previous experiences by Maderer and Gütl (2013), the requirements for the virtual learning environment and the assessment

### 3 Requirements and Design

systems are defined as follows:

- Virtual Learning Environment
  - Experiments
    - \* All new objects need to raise events
    - \* Represent the assignments tasks in an immersively sound way, and make it as dynamic as possible
    - \* Immediate feedback and assignment representation should follow an intuitive color scheme
  - Non-Functional Requirements
    - \* An easy-to-use interface for the experiment expert for the connection to the assessment system.
    - \* Provide the possibility to export to as many different platforms (mobile, browser, desktop, VR headsets, etc.) as possible
    - \* If possible, keep all newly created models as generic as possible to ensure reusability
- Assessment System
  - Feedback
    - \* Possibility to define immediate text based feedback
    - \* Possibility to define an assignment sheet containing not only text but answerable questions
    - \* Questions can have different ways of input fields, like text, buttons, check boxes
    - \* Integrate the definition of Feedback into the current model to reduce clustering
  - Model
    - \* Possibility to perform calculations like addition, multiplication or sine
    - \* Separate assignments and assessments to provide better structure and reusability of assessments in different assignments
    - \* Weave definition of feedback into the same structure - immediate text feedback and assignment questionnaire sheet should be definable inside the model
  - Service / Web Service
    - \* Provide a simple way to interact with multiple different learning environments and thus clients
    - \* Provide the support for multiple different experiment definitions
    - \* Provide an overview (logger) over all happenings inside the evaluation engine for simpler debugging of the experiment definition
  - Non-Functional

### 3 Requirements and Design

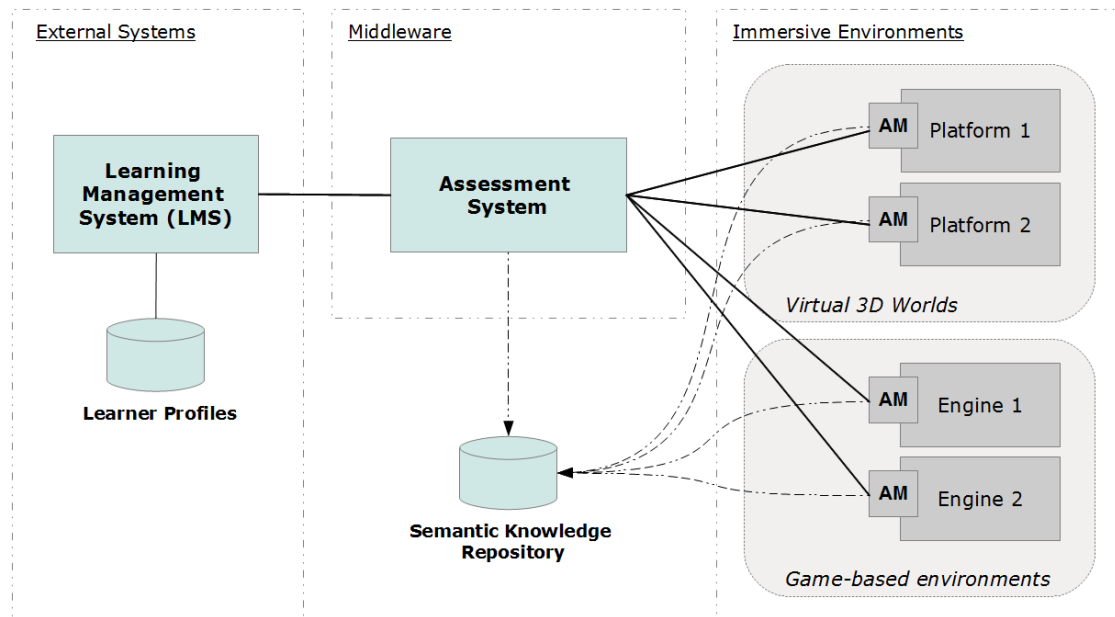


Figure 3.1: Conceptual architecture of the generalized flexible assessment approach (Maderer, Gütl, & AL-Smadi, 2013; as cited by Maderer, 2013)

- \* Support different virtual learning environments of any type (Moodle, Immersive, etc) at once
- \* Support different experiments for multiple users on different clients at once
- \* The communication interface needs to be as generic as possible

With these requirements in mind, the conceptual architecture emerges.

### 3.3 Conceptual Architecture

The basic conceptual architecture of the assessment system and the immersive environment corresponds to Maderer (2013). In general, the concept is built upon a three-tier architecture, differentiated between immersive environments, middleware and background (see Figure 3.1). Such a flexible assessment system should provide multiple components (Maderer, 2013):

- The ability to interact with multiple immersive virtual environments via a distinct protocol. This is realized by the communication interface, in the figure

referenced as “AM” (Assessment Module)

- The necessary logic to evaluate the user behavior
- A link to a semantic knowledge repository, which holds meta data for objects
- Tracking of the state of the environment and user behavior
- Pre-processing of events to provide useful information
- Incorporating a learning management system which adapts the feedback intensity based on preferences and foreknowledge which saves a learner profile about individual learners to keep track of achievements and assessment outcomes

In order to prove the credibility of the conceptual architecture, a working prototype with an immersive 3D simulated environment with immediate feedback is developed.

### 3.4 Design of the Assessment System

Having a closer look at Maderer (2013, section 5.5ff), the design of the assessment module already describes very detailed how the conceptual architecture should be realized. It is used as the foundation for the enhancements described in the following sections. The design of Maderer (2013) defines how the model is structured. The engine interprets so called behavior patterns, which match against events and updates of the environment. This is achieved by different match expressions to compare values, groups of values, or negations of values. Furthermore, it described how feedback is created and defined. In the course of this work all of these points are extended as discussed in the following sections.

#### 3.4.1 Feedback

Up to this point, only textual feedback existed. To achieve the questionnaire feedback (also called the assessment sheet), the feedback is split up in two parts: the textual immediate feedback and the questionnaire knowledge-based assignment sheet. Both of them follow the same classification:

- **Success:** positive feedback for things done correctly.
- **Hint:** if something might be wrong or should be performed differently.
- **Mistake:** if something was done wrong.

### 3 Requirements and Design

The immediate feedback and the assignment sheet are represented differently. Simple text feedback consist of a text variable and is not able to await user input. In contrast, the assignment sheet consists of text fields, questions and multiple different input fields. These input fields include:

- **Text field:** A simple text field for answering the question with free text. A hint text can be defined as well as a default value.
- **Check Box:** A box which may be toggled on or off without any restrictions to other check boxes.
- **Radio Button:** A box which may be checked but not unchecked by clicking. It is part of a group, hence the definition of a group name is needed. Exactly one element of this group typically needs to be selected
- **Drop Down:** A field of predefined values which may be selected by the user. On preselected value may be defined
- **Button:** The primary aim of the button is to send the form. The form is submitted via an event and environment change actions.

All of these inputs are organized in rows to define a structure at representation. Every input field must carry a distinct name, which is used for the environment alternation. Furthermore, this assignment sheet is sent from the assessment system to gain the benefit, that the experiment expert can define multiple assessment sheets, such that the same experiment setup can have multiple different assignment sheets.

Moreover, the idea of the assignment sheet is to be persistent for the user, like it would be in real life. All the default and hint values described before can be sent via the model provided commands, such as simple value, mathematical-, arithmetical or logical operations, which are evaluated, when the feedback needs to be sent. This is done by the evaluation engine.

#### 3.4.2 Evaluation Engine

The engine can be best described as the heart of the controller. It contains the evaluators and renderer needed to reflect the current model and be able to evaluate the environment changes correctly. This part of the assessment system undergoes the most changes.

The evaluators are rendering the correct actions for all expressions. Expressions are all mathematical and logical operations in the model, such as add, subtract, AND, OR, compare, to name just a few. The renderer is creating the correct object to be sent to the view.

For the assignment part, the engine needs to be adapted, to evaluate assignments with multiple assessments, not only a single assessment. That means, initially a list of assessments needs to be loaded, their dependencies checked and the correct ones initialized and loaded. With dependencies, it is possible to create a step-by-step evaluation. Furthermore, it needs to be possible, to reuse assessments in other assignments. The idea is, to only load the behavior patterns which are able to activate dependency-met assessments, and all behavior patterns of all active assessments.

Last but not least, to ensure a correct behavior of the assignment evaluation, the engine needs to provide a way to activate and finish certain assessments. This is needed, to not accidentally evaluate multiple assessments at once, just because they have overlapping behavior patterns.

The model XML files are imported as a template, whereas the renderer creates a working representation. This fact has not changed, but was enhanced with more objects as described above. Nevertheless, the model needs to undergo some changes as well, as the structure has been changed.

### 3.4.3 Assessment Model

The model can be seen as a template which needs to be rendered by the renderer to be fully functional. The model is changed from a simple assessment XML file to a structure, where an assignment definition (also a separate XML file) is at the top level. This assignment definition then references assessment definitions. This leads to the fact, that multiple assignments may contain the same assessments.

The focus of the XML files is to be as human readable as possible, as an editor is not (yet) provided. The usage of attributes was avoided as much as possible and only used to define the behavior of an expression, but not the type.

Because of the extension of the functionality of the engine, the model needs to be adapted too. All described mathematical operators and feedback objects (see subsection 3.4.2) have to be represented in the model as well. The same goes for the aforementioned activate and finish functions.

### 3.4.4 Web Service

The service provides the interface for communication between the 3D environment and the assessment tool (in this case via the communication interface). It is utilized by the web service to define a uniform interface, usable by any application, as long



as these programs are conform to the parametric structure, which is implemented with XML encoding. Furthermore, all the needed methods for context creation, event processing and logging are provided by the service.

As creating models for the assessment system can get very complex, a thorough logging information during this process is of great help. For this reason, the web service provides a logging interface, which not only show all the logging information created by the assessment system, but does provide filter functionality and a structural sound tree view for object. This tree view is able to represent any parsable object in a resizable and movable collapsing view (see Figure 4.11).

As opposed to the engine, the web service itself stays unchanged for the most parts, as it does already provide everything which is needed for a flawless communication between the virtual learning environment and the assessment system. With the web service as the last piece of the assessment systems conceptual architecture discussed, the requirements for the virtual learning environment changed too.

### **3.5 Design of the Immersive Environment**

The 3D immersive environment of the proposed proof of concept is going to be renewed. Therefore, the design of the immersive environment needs to be redone, as not only the technical background changes, but the environmental conditions dictated by the framework too.

#### **3.5.1 Experiment Workplace**

The experiment workplace should replace the physical deployment of a physics experiment. This means, that the environment should be structural sound for the user. Furthermore, every created object should be stand alone, which means it does not depend on any other object or even experiments. This automatically leads to the fact that all object have to raise events and be able to register event listeners in order to interact with each other.

It makes sense for the experiment expert to keep in mind to create one central object which handles all the event registrations between the objects. This reduces maintenance time drastically. Moreover, all object could be defined, such that they are reusable in other experiments. To communicate with the assessment system, a single point is defined: the communication interface.

### 3.5.2 Communication Interface

As an additional abstraction layer for the communication between the experiment and the assessment service (see subsection 3.4.4), the communication interface is introduced. It handles the inclusion of the external web service encapsulated in extended error detection. Thus, it safely prevents the experiment from crashing, when the web service is not reachable. All the requests for sending an event are still accepted in this case, but are not processed.

Every time, the user executes an action, it is considered to be an event. If such an event is registered with a listener, the assessment system needs to be updated. When sending the actions to the assessment system, all monitored environmental variables are sent as well to keep the system up-to-date. As discussed in the conceptual architecture (see section 3.3), the learning environment communicates with the assessment system via a strict protocol. To ease the task for the experiment export, conforming to this protocol, a generator for event objects is provided.

If the need arises, that the update of a watched value needs to be as frequent as possible (not only on events), all monitors provide an optional feature to update the value of the variables on a per-frame basis. As Maderer (2013) argues, this is in general not a feasible method as it creates tremendous communication load and may slow down not only the assessment system but the client too.

## 3.6 Proof of Concept

The aim of this section is to design a proof of concept. It should show that the before mentioned design decisions are feasible in practical applications. Creating the whole requirements in one step would go beyond the scope of this master's thesis. Because of that, suitable frameworks are selected and discussed why they have been chosen. After that, a more detailed overview is provided, how the general design is applied in this particular case.

### 3.6.1 Immersive Environment

Given the discussed papers in section 2.2, only two reasonable candidates for the immersive environment evolve: the framework of choice by Maderer (2013) Open Wonderland, or the rather new framework Maroon, built in the Unity game engine. When comparing Open Wonderland and Unity, the latter provides a very well optimized physics engine as well as a huge free assets store. These are the reasons

### 3 Requirements and Design

why Maroon was chosen for the proposed work. Needless to say that the pendulum experiment fits perfectly into the ecosystem of experiments of Maroon. As a third crucial point, building new physics experiments in Maroon is particularly easy, as there are a lot of pre-fabricated objects which can be reused.

#### Requirements

When faced with the general design discussed in section 3.5, the requirements for this particular framework need to be discussed. Firstly, the pendulum experiment needs to be implemented completely new in Unity and adjusted to the environment of Maroon. That includes new 3D models for the pendulum, a stopwatch and, as discussed by Maderer and Gütl (2013), the missing calculator. Moreover, the logic needs to be redone, hence a manager script for the experiment needs to be created. This script is going to handle all events and input of the user as well as correct behavior of the interacting visible components.

Secondly, the experiment expert needs to be able to utilize the assessment system easily, a new and easy-to-use interface needs to be provided, to handle the environment changes as comfortable as possible. This interface needs to provide all necessary functions to send events to and receive feedback from the assessment system easily. Furthermore, there should be no code needed to handle the connection to the assessment system, just a simple configuration interface.

#### Experiment Workplace

As a showcase, a simple pendulum experiment is created inside of the Maroon framework. As mentioned in subsection 2.2.4 every experiment is encapsulated from all other experiments. This makes development of new experiments easy as there is no dependency.

The experiment itself consists of 3 main parts: The pendulum itself, a stop watch to measure the time for one period and, as discussed by Maderer and Gütl (2013), a pocket calculator with basic calculation operators. As this experiment is implemented for the desktop environment, all interactable objects need to be operable via mouse or via keystrokes on the keyboard. Especially with Virtual Reality in mind, interaction with mouse is very important.

To lighten up the ambiance, some free assets from the Unity Asset Store<sup>1</sup> are added. These assets are plants, shelves, books, boxes and couches to name a few. For the

---

<sup>1</sup><https://assetstore.unity.com/>

### 3 Requirements and Design

communication with the assessment system, the prototype itself needs to be further discussed, before diving into one of its sub-parts.

The big improvement in contrast to Open Wonderland is, that the physics behind the pendulum do not have to be implemented by the expert anymore. Unity provides a very rich and detailed physics engine, which handles this tasks entirely. This leads to the complete obsolescence of actual C#-code in this part. To realize a pendulum like this one, all the necessary components are provided by the Unity engine itself.

The calculator as well as the stopwatch are objects which may be reused at any experiment as well, as these objects get encapsulated in a so called prefab and provide their own stand alone code with event handling. The calculator throws an event on every single button press, the stopwatch on start, stop and reset of the watch. Furthermore, the elapsed time of the stopwatch is accessible at any point in time. As it is part of the major change of the assessment system, the representation of the knowledge feedback is discussed in more detail.

#### **Feedback**

The feedback is split up into two parts: the textual immediate feedback and the questionnaire knowledge-based feedback. Both of them follow the same color scheme according to their class:

- **Green:** Success
- **White:** Hint
- **Yellow:** Mistake

The immediate feedback is integrated into the current existing GUI. Its purpose is to show short information about the current doings of the user. Thus it is shown temporarily. In contrast, the questionnaire feedback is represented on a white board. Hereby, the goal is to be persistent throughout all of the actions of the user and other feedback. Furthermore, this assignment sheet is sent from the assessment system to gain the benefit, that the experiment expert can define the tasks, such that the same experiment setup can have multiple different assignment sheets.

To send all the events and receive the feedback, the communication interface has to be defined as well, as the assessment system is communicating through a web server. This is discussed in the next section.

### Communication Interface

As aforementioned, a special interface is introduced to provide a failsafe possibility to communicate with the web server. This is done by providing a DLL file, which can be included inside Unity. This library provides all classes and constructs which are needed to communicate with the web server.

Moreover, a simple way to create the events is provided. In combination with the interfaces presented to the experiment expert to define values which need to be monitored, it creates a very easy-to-use basis. These values are able to provide continuous updates that lead to a per-frame update of the assessment system. As described by Maderer (2013), this method is not desired as it normally results in extreme overhead and drastic efficiency downgrades. Nevertheless, the option is given as there might be certain circumstances where this is a desired behavior.

The communication interface literally paves the way to the discussion of the assessment system prototype, as the only communication happens through its web service.

### 3.6.2 Assessment System

If an immersive environment, like Maroon, does not provide an integrated assessment system, considerations need to be taken if an external or internal system should be implemented. According to Alevin, McLaren, and Sewall (2009), external assessment systems provide the possibility to enhance performance of the clients drastically by outsourcing the interpretation task of the user assessment to a different device. Furthermore, it is much easier to provide the flexibility of interacting with multiple learning environments (and thus clients), as well as multiple experiments at once. As some learning environments address VR and AR devices too, it is especially crucial for them to save computational resource and thus keep up a steady frame rate.

The decision for an external assessment system narrows the choice to SOFIA or CTAT. CTAT was implemented to be an intelligent tutoring system for mathematics. To implement a general approach seems hardly feasible. Furthermore, although considered to be easier, example based assessment systems narrow the varieties of implementation strongly and thus paving rather fixed paths for the student on how to tackle a problem. CTAT is a graph-based framework in contrast to SOFIA. As one of the objectives is greatest possible adaptability for the assessment system, it leaves only SOFIA as the framework of choice (Alevin, McLaren, & Sewall, 2009; Maderer & Gütl, 2013). With the two frameworks chosen, the adaptations can be discussed on how to achieve the symbioses.

#### Requirements

With the decision made to combine the virtual learning environment “Maroon” in Unity as an example for the prototype with the assessment system “SOFIA Evaluation Engine” the design of the adaptations need to be done. When faced with the general design discussed in section 3.4, some additional requirements need to be considered to achieve the desired goal:

Firstly, as described in section 3.4, multiple adjustments need to be done on the assessment system in order to provide the desired functionality of feedback with questionnaire. It is desired to not only be able to assess the doings of the student, but also his gained knowledge. This is done by presenting questions with input fields to the user. This feature of questions with input fields of different kind needs to be implemented in Unity as well. Moreover, more sophisticated mathematical operators need to be implemented inside the assessment system, such that it is possible to check the calculations of the student inside the assessment system. These operations are addition, subtraction, division, multiplications, as well as logical operators like AND, OR and XOR. Last but not least, the *Value* class needs to be able to handle constants like  $\pi$  or  $e$ .

Secondly, the system should not only be able to assess a single experiment, but rather a set of exercises and/or even multiple settings of experiments in one go. That means, that it needs to be able to accept assignments which contain of multiple assessments, defined with certain dependencies. When implemented, this means, it is possible to assign one assessment to multiple assignments but individually evaluated. This may result in a tremendous reduction of work when similar assessment need to be performed during one assignment.

Going hand-in-hand with the last two points, technology evolved since the implementation of the prototype of SOFIA, which means some code and techniques may be re-factored and simplified with new Technologies like Linq (Meijer, Beckman, & Bierman, 2006). An example is implicit getter and setter implementation for fields in C# 6.0 or higher.

#### Assessment Prototype

The prototype, as described by Maderer (2013), consists of multiple parts: Engine, (Web-) Service, and Model, beside the testing projects. The whole structure of the assessment system is best described as a Model-View-Controller (MVC) concept (Bucanek, 2009). The view is handled by any external program communicating via the web service with the engine. This means, that the Service in the prototype is the

### 3 Requirements and Design

interface for the view and thus part of the MVC concept. With the visitor pattern, the Engine (Controller) decides how to handle the fragment of the current Model. The model is represented as XML files.

#### **Web Service**

The service represents the View in the MVC, thus providing the interface for communication between the 3D environment and the assessment tool (in this case via the communication interface). It is utilized by the web service to define a uniform interface, usable by any application, as long as these programs conform to the parametric structure, which is implemented with XML encoding. Furthermore, all the needed methods for context creation, event processing and logging are provided by the service.

As creating models for the assessment system can get very complex, a thorough logging information during this process is of great help. For that reason, the web service provides a logging interface, which not only shows all the logging information created by the assessment system, but does provide filter functionality and a structural sound tree view for object. This tree view is able to represent any parsable object in a resizable and movable collapsing tree view (see Figure 4.11).

As opposed to the engine, the web service itself stays unchanged for the most parts, as it does already provide everything which is needed for a flawless communication between the virtual learning environment and SOFIA Evaluation Engine.

#### **Refactoring**

During the analyze phase, some code refactoring possibilities showed up. First of all, the .NET-Framework (Wigley, Sutton, Wheelwright, Burbidge, & Mcloud, 2002) used was 2.0. The current version now supported by C# is 4.6.2. Basically, VisualStudio (Martin & Banks, 2014) provides a simple functionality to switch the desired .NET framework. In this case, it was no problem, as all the used functions and data sets are still supported. Caution is needed for the unity interface though, as unity is working with C# version 4.0 and .NET 3.5.

Furthermore, C# got improved too, as the used version was C# 2.0, the current version is C# 7.0. Again, this is just a matter of the chosen compiler inside VisualStudio and had basically no effect on the code, as all commands are backward compatible. The reason to do this is, as there are new features which improve readability of the code by dramatic reduction of code (Albahari & Albahari, 2017).

### 3 Requirements and Design

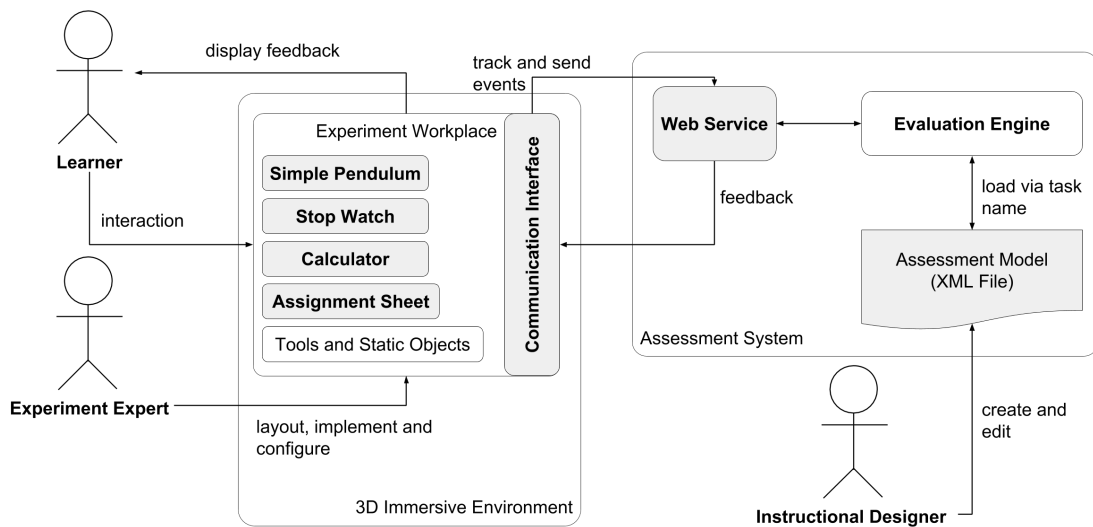


Figure 3.2: The design of the assessment system interacting with the 3D immersive environment (in this case “Maroon”). Gray squares are interactable and/or configurable objects. The Evaluation Engine loads the assessment model and communicates via a web service with the virtual environment. On interaction by the user, it tracks and sends these events via the communication interface to the web service (based on Maderer, 2013)

A very good example are automatic implemented properties, which were introduced in C# 3.5. This very useful feature handles the default case of a private variable with public getter and setter function in one single line – the private variable does not need to be mentioned and the getter and setter function aren’t explicitly implemented. Other changes according to the code are just minor bug fixes or good-practice changes, for example the way how tasks are created and initialized or naming convention on variable names.

### 3.7 Summary

In Figure 3.2 the design of the proof of concept prototype is illustrated. It shows how the components are interacting with each other. The most important part is, that the communication between Maroon and SOFIA Evaluation Engine happens through the web service which creates an easy-to-adapt interface for other immersive environments.

The analysis of the current prototype gave a very good insight on the existing program. It revealed that the implementation was very well made by following exactly the rules of the patterns and thus providing a very good basis for improvements and



### 3 Requirements and Design

enhancements. Fortunately, the used platform was C#, although Open Wonderland is based on Java, which even further proves the modularity of SOFIA Evaluation Engine.

The engine itself is enhanced with the feature of assignment parsing, instead of a single assessment parser. This means, that according to the settings given in the assignment, only parts of the behavior patterns of inactive assessments are loaded, in contrast to the active assessments, which are fully loaded. Furthermore, the model can activate and deactivate assessments at any given time. Going hand in hand with this improvement, the expressions are enhanced with multiple mathematical as well as logical operators and two new commands for activating and finishing assessments. Last but not least, with code refactoring the anyway very maintainable code is improved even further by reducing code with Linq or by using newly provided features by a newer C# version.

Comparing the newly designed pendulum experiment to the Open Wonderland version mentioned by Maderer (2013), not much has changed in terms of usability for the user, except of the desired calculator. With that said, the rendering, maintainability for the expert and the seamless integration in the game-like framework Maroon are big improvements nonetheless. Especially because the pendulum is calculated by the Unity physics engine and has not to be programmed by the experiment expert anymore.

## 4 Development

This chapter discusses the concrete development of the proof of concept. The UML diagrams and the exact realizations are provided and the reasons explained, why these decisions were made. As described in section 3.6.2, the used technology will be the latest C# 7.0 with .NET framework 4.6.2, except for the Unity interface, which will be C# 7.0 but with .NET framework 3.5 for compatibility reasons.

The first step was to create the experiment, as it can be designed completely independent from the assessment system. Then, the communication interface was implemented in order to create the connection to the assessment system. As a final big step, the assessment system was enhanced with the defined features such as the assignment sheet and improvements like more mathematical operations.

### 4.1 Pendulum Experiment

As mentioned before, the pendulum experiment will be developed first as it basically works without the assessment system. To provide a good overview on how it is done, and which parts are provided by Maroon, the structure of Maroon is discussed first.

#### 4.1.1 Structure of Maroon

As described in subsection 2.2.4, Maroon is a virtual learning environment (VLE) to provide an immersive educational suite for various experiments. The simple yet ingenious concept makes it a perfect front end for prototyping the new assessment system, as experiments are treated separately and thus do not interfere with each other (Pirker, 2017). That means, it is possible to implement the assessment module in one experiment, whereas other experiments can be used without it. This results in the huge benefit of seamless testing and improving the assessment system in combination with the experiment for future work, without any dependency interference from

and with other experiments, while at the same time gaining all the benefits of the Maroon framework.

These benefits include shared classes, as can be seen in Figure 4.1. These classes are used by all experiments and contain the gamification manager and the language manager. The gamification manager defines the theme of the GUI by providing the necessary prefab. A prefab (short for prefabrication) is an entity in Unity to reuse objects. This technique is utilized multiple times during the development, for example with the stopwatch and the calculator. This way, this object can be instantiated in different experiments without having to rebuild them.

The language manager handles the translation of the texts for the multi language support. Currently, it supports German, English, and Italian. The texts are chosen by a key word, which is resolved to the text according to currently selected language. The default language is set during start up of the application according to the system language.

### 4.1.2 Structure of the Pendulum Experiment

The pendulum experiment, which will be implemented as a showcase of the defined conceptual architecture, consists of five subsections:

1. **Background and Environment:** Describes how the center piece, the *Pendulum-Manager*, and the ambient decoration is developed. Furthermore, discusses how the communication interface is implemented and how it corresponds with the feedback visualizations.
2. **Assignment Sheet:** Describes in more detail, how the assignment sheet is represented structurally sound in the environment and how the questions with their input fields are visualized.
3. **Pendulum:** Describes the realization of the pendulum itself.
4. **Stopwatch:** The provided stopwatch should be used to measure the time of one period. It provides a start, stop and reset action.
5. **Calculator:** The calculator provides all basic mathematical operations such that the calculation of the frequency can be fulfilled inside the environment.

## 4 Development

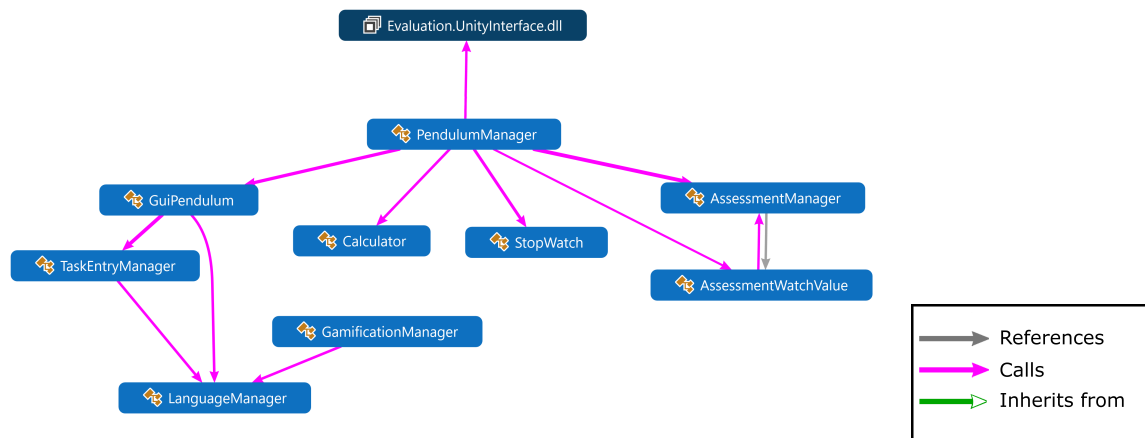


Figure 4.1: The UML diagram of the classes concerning the pendulum experiment. The UnityInterface.dll is solely used by the PendulumManager class by creating all the event listeners and handlers. The library is an interface provided by the assessment system to abstract the web service connection. The Gamification Manager and Language Manager classes are provided by the Maroon framework to create the GUI representations.

In Figure 4.1 the UML diagram can be observed, which states, how the classes interact with each other. Furthermore, Figure 4.2 provides an overview of the end result of the scene.

### 4.1.3 Background and Environment

Every experiment in Maroon is created in its own scene. Thus, it is independent of all other experiments, but does share all the resources and some global scripts with the rest of the laboratory. The scene is built up like a big room, as can be observed in Figure 4.2. Nearly all of the ambient object, like the table, the sofa, the bookshelf with its content, the plants and the white board are freeware in the Unity Asset store<sup>1</sup>. The blue help character is a prefab from the Maroon framework and represents the character which follows the student through the whole Laboratory as a helping guide if needed.

To be able to fully utilize these provided components, two base classes are created: *PendulumManager* and *GuiPendulum*.

<sup>1</sup><https://assetstore.unity.com/>

## 4 Development

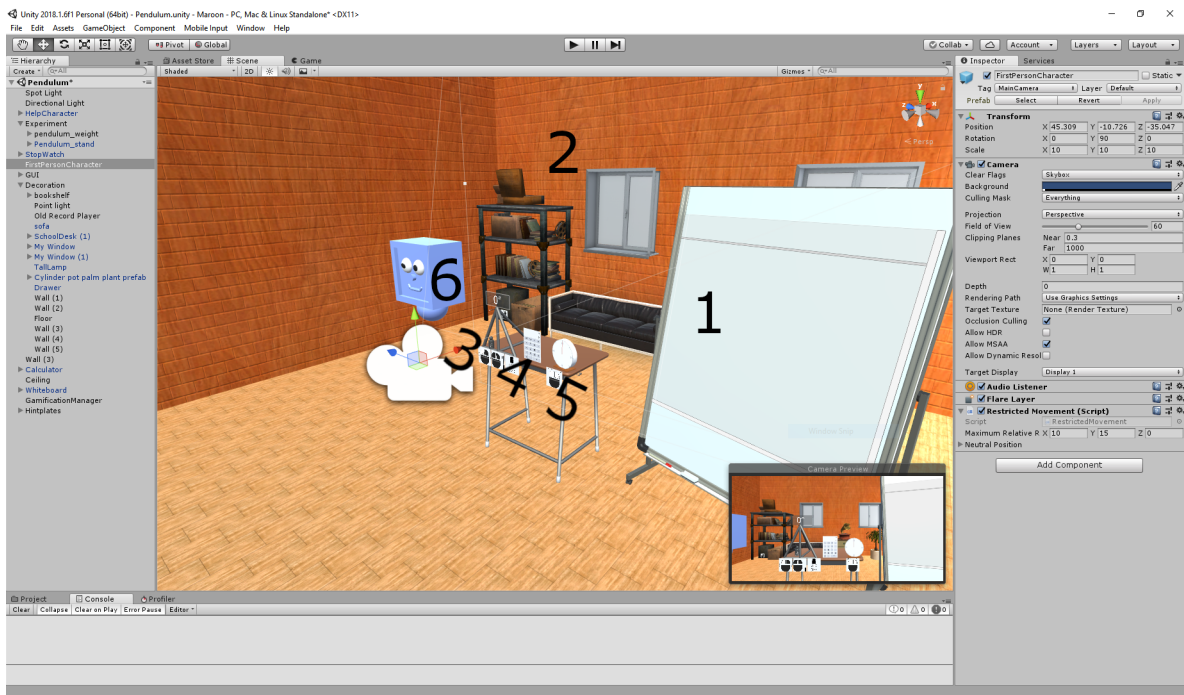


Figure 4.2: The scene during edit in Unity. The camera is the selected object. Enumeration: 1 Assignment Sheet, 2 Background Scenery, 3 Pendulum, 4 Calculator, 5 Stop Watch, and 6 Help Character

## 4 Development

Mouse position	Mouse button	Keyboard Button	Action
Pendulum	left hold	-	deflect pendulum in direction of drag
Calculator	left	NumPad	action according to button pressed
Stopwatch	left	Q	reset and start stopwatch
Stopwatch	left	E	stop stopwatch
Stopwatch	right	-	reset stopwatch
Box	left	-	toggle time laps mode
Anywhere	mouse wheel	-	adjust rope length
Anywhere	-	A	reduce mass (minimum 1 kg)
Anywhere	-	D	increase mass (maximum 2kg)

Table 4.1: Different action on different objects handled by the pendulum manager

### PendulumManager

The *PendulumManager* is the heart of the experiment. On instantiation it registers all event handlers and during the animation it listens on input from the student (either mouse or keyboard). According to where and how the user clicks, different actions are triggered, see Table 4.1 for more details. Furthermore, it is the only class which is directly referencing the *GuiPendulum* class.

This class needs to be created by the experiment expert every time a new experiment is created. As Maroon and SOFIA Evaluation Engine are very agile programs, a no-code-environment seems rather unfeasible at this point. This means, the experiment expert, at least up to this point, nearly always has to have a basic knowledge about C# programming to be able to create new experiments. Nevertheless, this mainly just includes event registration of function, for example, the stop watch start and stop or key presses.

### GuiPendulum

This class handles all representation of feedback in the experiment. Basically, feedback is always sent from the assessment system and is generated most of the time after game event was sent by the *PendulumManager*. However, sometimes Feedback is triggering Feedback. This happens, if the user answers a question and thus receives

## 4 Development

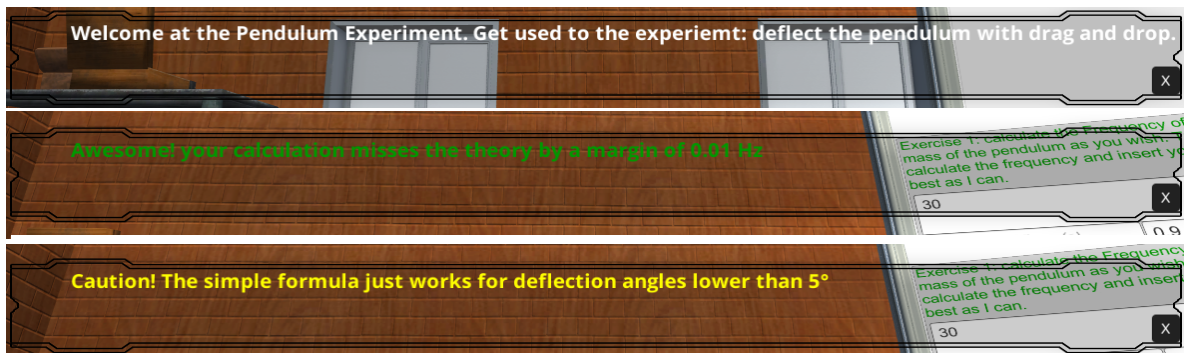


Figure 4.3: The GUI shows off immediate textual feedback. The first message has the color scheme of a hint, the second one of a success and the third one of a mistake.

either a new question or a textual feedback from the system. According to the class of feedback for the user a different representation is chosen:

- **TextFeedback:** Textual feedback without input is represented as GUI text on the screen with a calculated amount of time, more details are described in section 4.1.3
- **QuestionFeedback:** Question feedback is sent to the *TaskEntryManager*, which handles the result as described in section 4.1.3
- **Technical Feedback:** This type was introduced, to be able to communicate with the virtual learning environment, without showing feedback directly to the user, but rather instruct the system what to do. Examples are *EnterSection* or *clear* events.

For handling the questionnaire feedback, the class contains another *TaskEntryManager* class which handles the visualization of the assignment sheet feedback, here called "task entry". Every sent event, so a sent questionnaire too, may create an immediate feedback. The task entry manager sends all results from the assessment system back to the *GuiPendulum*, to decide, how to handle the generated feedback.

### Text Feedback

Basically the GUI is the semi transparent stripe which appears on the screen, whenever text feedback is generated. The GUI provides two key functionalities: The Color scheme and the duration of the text. An example can be seen in Figure 4.3

According to the type of the feedback a color is chosen based on Table 4.2. The reason, why green is not a full green is because of the contrast to the background. A full green is much too bright to be read easily.

## 4 Development

Type	Name of Color	R,G,B
Hint	white	1,1,1
Success	green	0,0.6,0
Mistake	yellow	1,1,0

Table 4.2: The chosen color based on the type of the feedback. This scheme holds for text feedback as well as question feedback.

As feedback text may differ in length, using a constant visualization time is not practicable for the GUI. Because of that, a formula is used to calculate the number of seconds to display the GUI. This formula is presented in Equation 4.1.

$$t_t = \sum_{i=1}^n t_s i \quad (4.1)$$

where

$$n = |W|$$

$W$ ...words in text

$t_s$ ...seconds per word

$t_s$  is adjustable in the *GuiPendulum* class by the experiment expert, but uses 1.0 second as default value. This results in a comfortable yet dynamic time to read the text, especially considering the fact, that the software is used by children in middle school age, who might not be fully experienced readers yet. Furthermore, if the text is considered to be obstructing, a small button is provided to conceal the current message. Nevertheless, this does not effect new messages.

In contrast to the text feedback, the question feedback is handled by a separate class called the *TaskEntryManager* which is part of the GUI. It handles the representation of the question feedback, as well as the user input regarding the questions and thus the Assignment Sheet.

### Assignment sheet

When feedback is parsed through the *GuiPendulum* class, it is decided based on the type of feedback, if it is handled internally (because it is a text feedback) or it is passed on to the *TaskEntryManager*. This class handles the question feedback by dynamically creating the assignment sheet. Figure 4.4 shows the exact hierarchy of



## 4 Development

this prefab. The grayed out elements are a temporary save for the input fields, which are predefined with their base settings.

The inputs of a question feedback are organized in rows. These rows are represented by the Panel in the grayed out picture. This panel carries a Unity provided component, to organize any child elements in its hierarchy in horizontal order. That means, that any defined input object in the feedback, will share the horizontal space with all other input fields equally.

All input fields are quite common knowledge and are not further discussed here, except of the toggle button, as it does provide two different functionalities: Check box and Radio Button. If the input field definition of the feedback defines a group name in multiple toggle boxes, then they are treated like radio buttons, thus only one can be in the selected state at a time. If there is no group name defined, it behaves like a check box, which means, no restrictions on the active state.

Only if a button is provided, it is possible to send the user input. Every button, no matter in which row it was defined, sends all the information of one question. To distinguish, which button was pressed, the triggered action carries the name of the button. There is no button added automatically, if no button was defined.

A question can change its colors too. At first, it is rendered in the color according to Table 4.2 which was defined with the question itself. When a button of the question is pressed and thus feedback is generated for the user, the question changes the color according to the type of feedback returned.

All these described GUI and assignment sheet objects and classes are globally reusable for any experiment. To create the experiment itself the aforementioned three other objects are still needed: The pendulum itself, the stopwatch and the calculator.

### 4.1.4 Pendulum

As discussed in section 3.6.1, the pendulum is completely handled by Unity and its provided Physics engine (Goldstone, 2009). That means, that the weight gizmo only needs to be offset from the center point of its parent and enhanced with a *RigidBody* component. Last step is to add a *HingeJoint* to the parent object to let the weight gizmo rotate just around one axis.

When the rope length is adjusted by the user, technically the gizmo is offset with the exact distance which is written on the weight gizmo itself. The rope is just a rendering which is done in the Update-method of the *PendulumManager* - drawing

## 4 Development

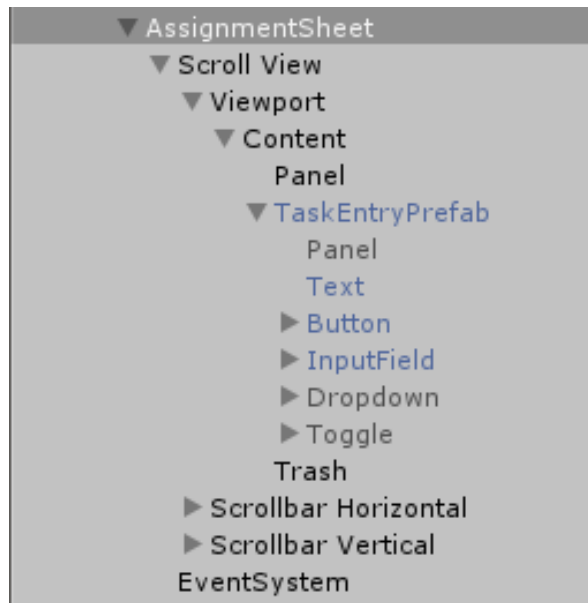


Figure 4.4: The Hierarchy of the Assignment sheet. After inserting anywhere in the scenery, it can be referenced by the TaskEntryManager.

a line from the center of the gizmo to the angle point of the HingeJoint. The weight change happens physically at the RigidBody itself, as well as visually by just altering the scale of the gizmo. Nevertheless, the scale of an object is not in direct correlation to its mass.

Furthermore a component called *AssessmentWatchValue* is added to the pendulum to monitor the variables “Transform.Rotation.X” (the angle of the rotation), “PendulumManager.RopeLength” (cm of offset) and “PendulumManager.Weight” (the current weight). For further details see subsection 4.3.2 and Figure 4.15 and also section 3.6.1.

### 4.1.5 Stopwatch

As it is a vital part of the experiment, a stop watch needs to be provided. Not only for convenience reasons for the student but also for the sake of assessability. The stopwatch is basically a cylinder and a cube with a custom texture. Inside the cylinder is a text field to present the current time in a  $\frac{1}{100}$  second precision. As mentioned in section 3.6.1, the stopwatch is a completely stand alone construct which may be used in any number in any experiment. Thus the implementation of the *StopWatch* class needs to be independent of any other classes. The stopwatch is then encapsulated in a prefab to be easily reused.

## 4 Development

Events will be raised at certain occasions to trigger externally defined event handlers. Furthermore, the stopwatch has no built in functionality to trigger itself – it needs to be triggered externally, which means, it may be mapped to any kind of input. On start, the stop watch class saves the current date time and activates the running variable. With this variable active, the update functions calculates the elapsed time and represents it in a 2 decimal precision on the text box inside the stop watch. Moreover, according to the elapsed time, the correct angle of the seconds hand is calculated and rotated. If a minute hand is provided (which it is not by default), the same happens for that too. In addition, an event called *OnStart* is raised, as there might be multiple listeners triggered, not only the function which is starting the stop watch.

On stop, the running variable is set to false. The Update function now immediately exits. As long as the reset function is not called, the *ElapsedTime* variable of the stop watch will stay at the current value. Yet again, an event called *OnStop* is raised. To reset the stopwatch, the reset function needs to be called. This resets the position of the hands, sets the text to “o.o” and sets the *ElapsedTime* to o.o. This action also raises an event called *OnReset*.

In addition to the raised events, which do provide a full information in the event parameters, this information may also be accessed directly at any time. The downside of this technique is, that the elapsed time can not be guaranteed to be exactly the value as there is no locking at this moment.

The stop watch is one of the two objects, which are created completely independently from the experiment. The second one is the calculator which will be discussed in the next section.

### 4.1.6 Calculator

Like the stop watch, the calculator is independent from the experiment itself. It just communicates via events to seal itself up against other classes.

The basic principle is quite simple. There is a primary number called right number, which is the one, the user currently manipulates by pressing one of the numeric buttons and the period. The period is only accepted once per number. For simplicity, the number is concatenated as a string, and when used in the calculation transformed to a double precision value (short double). If one of the operator buttons is pressed (means add, subtract, multiply, divide and enter), the right number is converted to double and saved as the left number. After entering the second number, which is again saved in the now cleared right number variable, and pressing any operator

button again (except enter), the result is calculated and set as the left number again. The difference is, that the enter button saves the result as the right number, such that it is shown on the big text field of the calculator.

The functions “sin” and “cos” (for sine and cosine) are operating solely on the right number, which means altering it immediately after pressed. Every button press triggers an event with a full featured information class, which button was pressed and which numbers are currently saved in the calculator. Furthermore, it is provided, how the button was triggered, either by keyboard and which button, or by mouse. The calculator is packed inside a prefab too, to make it easily reusable for other experiments.

### 4.1.7 Summary

The core of the visual experiment, the pendulum itself, is fully handled by Unity and its physics engine, which does not only greatly reduce the effort to implement and maintain the experiment but also proves Maroon as the correct choice for the simulated 3D environment. This, and the fact of the collaborative work provided in form of the GUI and language manager to even further reduce the work needed for the experiment expert.

Three of the core objects, the stop watch, the calculator and the assignment sheet, are implemented in a modular way, such that these objects may be reused wherever needed. This is achieved by encapsulating them in Unity prefabs for easy access. All these objects work with events to provide the necessary interface to interact with other object. With the experiment done, the assessment system may be enhanced and the *UnityInterface* – the communication interface – implemented to connect these two programs together.

## 4.2 Assessment System

The assessment system is the vital part of this work. Up to now, it is a solid but yet not entirely feature rich system. In this section, all the improvements are discussed, as well as the tracking module called *UnityInterface*. Furthermore, the model implementation of the pendulum showcase is described. Figure 4.5 shows a basic overview between the components of Sofia Evaluation Engine. The UML was stripped to just the interacting classes, whereas further details can be found in the corresponding subsections.

## 4 Development

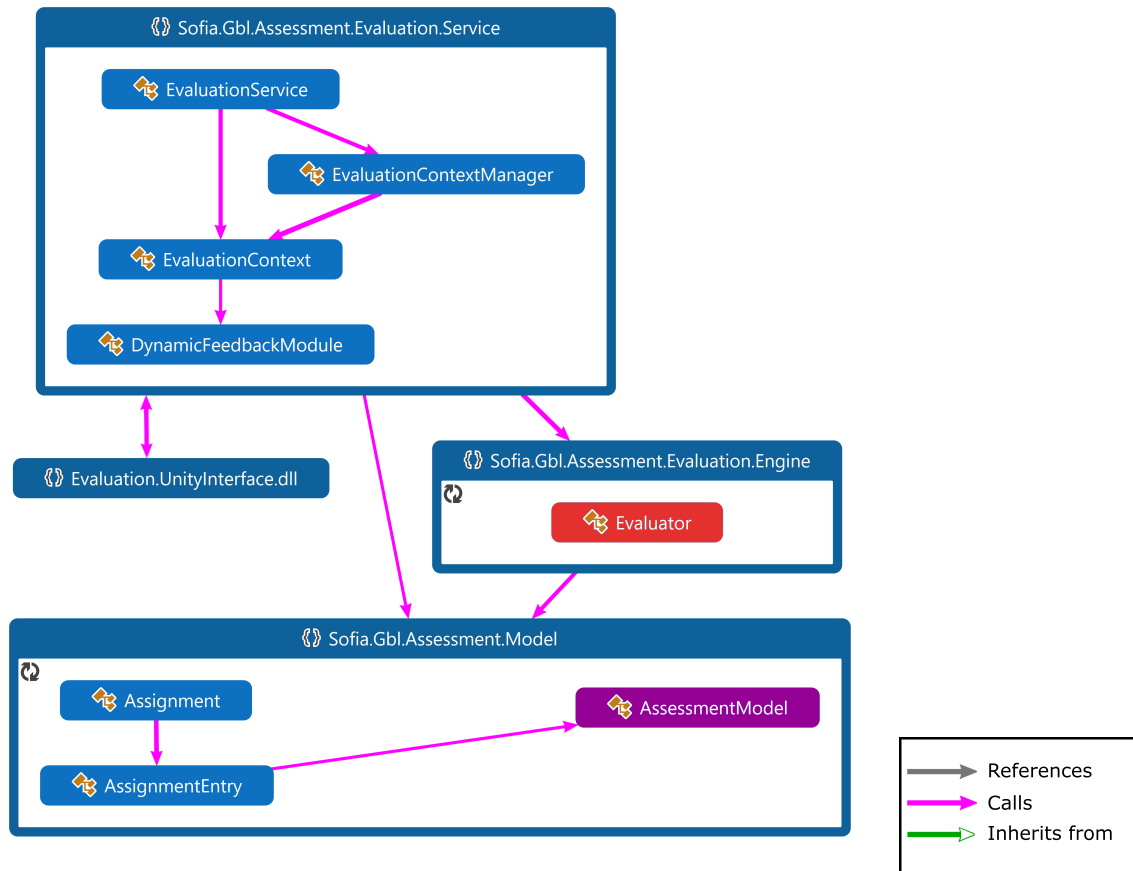


Figure 4.5: Basic UML diagram of the SOFIA Evaluation Engine structure. More detail is shown in Figure 4.6 for the Engine/Evaluator and Figure 4.8 and Figure 4.9 for the Model/AssessmentModel.

### 4.2.1 Evaluator

When having a close look at Figure 4.6, it can be observed, that the graphic already contains the newly planned classes for Assignment, corresponding helper-/subclasses and the classes for the feedback rendering. The starting point is entered through the instantiation of a new *EvaluationContext* and is marked in the graphic with yellow borders. This context stays unique for this exact execution of the assignment and is recognized and handled through its Universal Unique Identifier (UUID). During creation of the context, the *Assignment* is initialized by using the *Evaluator*.

## 4 Development

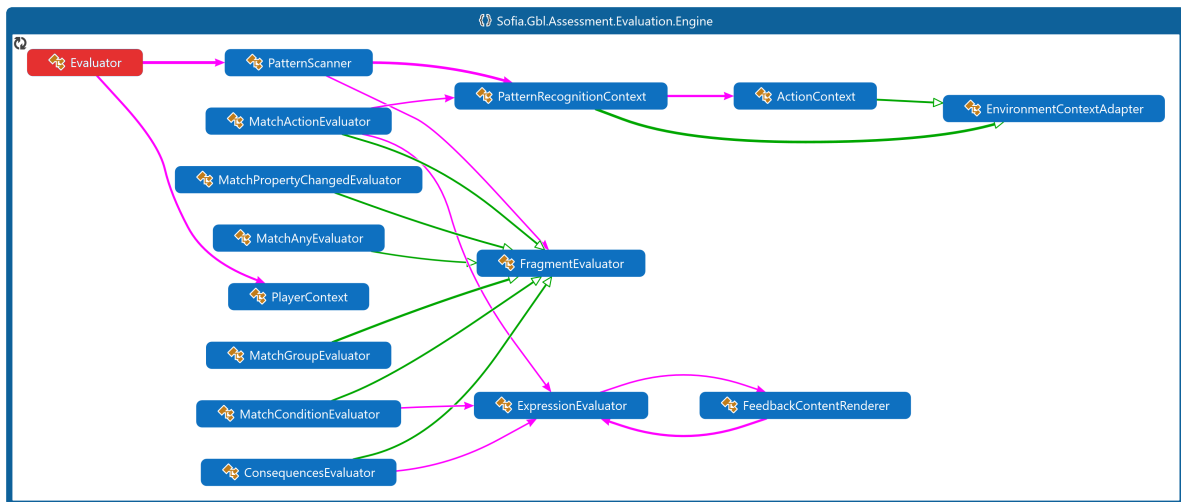


Figure 4.6: UML diagram of the Engine. It loads the *Assignment* class of the model and interprets it by accessing the code of the corresponding *Evaluator*. This is done by the Visitor-Design. The legend can be found in Figure 4.5.

This is done by parsing the corresponding XML file and its referenced *Assessment* XML files.

The primary problem is that the Model as well as the Evaluator needs to be exchangeable - the Model obviously changes with every experiment, the evaluator changes according to the definition of the model (*ExpressionEvaluator*, *MatchActionEvaluator*, *FeedbackContentRenderer*, etc.) This problem is usually overcome by realizing the so called "Visitor Pattern" (Gamma, 1995): The Visitor Pattern was used to provide the necessary flexibility of interchangeable logic and model definition. The XML files are a direct definition of the model. The model itself, according to the Model-View-Controller architecture, does not contain any logic. The visitor on the other hand inherits from the *Evaluator* class and implements all needed **Visit()** methods. This means, that the model as well as the visitor may be exchanged easily.

In this particular case, one of the sub classes of *FragmentEvaluator* are the visitors of the Fragment classes, whereas the *ExpressionEvaluator* is the visitor of all the Expression sub classes. To be fully functioning, a Context needs to be provided to hold all the environment changes and variables. In combination with the model, the *Action* and their *Consequences* may be evaluated to generate the Feedback.

Moreover, at the moment, the evaluator is only able to run one assessment per context. This means that an assignment sheet with multiple steps and different experiments is very hard to model, especially if certain experiments should be reusable.

## 4 Development

- [Clear all Entries](#)
- [Verbose](#)
- [Refresh all Entries](#)

Search:

Task ID	Context ID	Type	Text	Time Stamp	Object
		Verbose	[Pendulum Maroon 3] Adding behaviour AnswerQuestion	11:43:45 AM:502	
		Verbose	[Pendulum Maroon 3] Adding behaviour AnswerQuestion	11:43:45 AM:501	
		Verbose	[Pendulum Maroon 3] Adding behaviour Sofia.Gbl.Assessment.Model.MatchCondition	11:43:45 AM:500	
		Verbose	[Pendulum Maroon 3] Adding behaviour AnswerQuestion	11:43:45 AM:498	
		Verbose	[Pendulum Maroon 3] Adding behaviour AnswerQuestion	11:43:45 AM:497	
		Verbose	[Pendulum Maroon 3] Adding behaviour AnswerQuestion	11:43:45 AM:496	
		Verbose	[Pendulum Maroon 3] Adding behaviour Sofia.Gbl.Assessment.Model.MatchAny	11:43:45 AM:494	
		Verbose	[Pendulum Maroon 3] Adding behaviour Sofia.Gbl.Assessment.Model.MatchAny	11:43:45 AM:492	
		Verbose	[Pendulum Maroon 3] Adding behaviour UseObject	11:43:45 AM:490	
		Verbose	[Pendulum Maroon 3] Adding behaviour UseObject	11:43:45 AM:489	
		Verbose	[Pendulum Maroon 3] Adding behaviour UseObject	11:43:45 AM:488	
		Verbose	[Pendulum Maroon 3] Adding behaviour EnterSection	11:43:45 AM:487	
pendulum_assignment	7461f3bd-936d-45f6-be23-604e712e018c	Info	[activate expression] activating Pendulum Maroon 3	11:43:45 AM:454	
		Verbose	[Pendulum Maroon 3] Adding behaviour EnterSection	11:43:45 AM:414	
pendulum_assignment	7461f3bd-936d-45f6-be23-604e712e018c	Info	[finish expression] finishing Pendulum Maroon 2	11:43:45 AM:406	
		Verbose	[Pendulum Maroon 2] Adding behaviour AnswerQuestion	11:43:41 AM:960	

Figure 4.7: A logger view of the process of finishing “Pendulum Maroon 2” and activating “Pendulum Maroon 3”. The view is sorted from newest to oldest.

### Assignments

One primary goal of the enhancements is to provide the functionality of defining assignments which consist of multiple assessments. Furthermore, these assessments in this particular assignment need to have dependencies, mainly because of performance reasons. Non-activatable assessments are not parsed in any way and as such drastically reduce the number of actively processed behavior-patterns.

Because of that, multiple new classes are introduced in the model. First of all, the *Assignment* class which represents the assignment sheet and consists of multiple *AssignmentEntry* class instances. The assignment entries are distinguished from each other by a GUID, as there might be multiple assignment entries with the exact same parameters. With this GUID, the dependencies can be created. Moreover, all assignment entries carry an “active” and “finished” variable. If all dependencies of an assignment entry are met, only the *EnterSection* actions are added. That is because multiple assignment entries dependencies might be met but only one should be executed next. The active variable is set to true either by the assignment model definition (for the first section for example) or if the *Activate* command is sent via the service. If this variable is true, all *BehaviorPatterns* are loaded.

The finished variable is set to true either by the assignment model definition (al-

though this will not be meaningful in most cases) or if the *Finish* command is sent via the service . Accordingly, this will prevent any *BehaviorPatterns* to be loaded, although all dependencies are met. Furthermore, a dependency only is counted as met, if and only if this variable is true.

Figure 4.7 shows an example of finishing the assessment “Pendulum Maroon 2” and activating “Pendulum Maroon 3”. The logger entries need to be read bottom up, as the events are sorted from newest to oldest. As can be observed, before finishing “Pendulum Maroon 2”, only its behavior-patterns have been added to the actively parsed patterns. After the finish instruction, issued by the user in the Virtual Learning environment and executed by the model, only the *EnterSection* part of “Pendulum Maroon 3” was added. Furthermore, no behavior patterns of “Pendulum Maroon 2” are parsed anymore. Only after the event of executing this enter section pattern and thus executing the activation of this assessment, all its behavior patterns are added.

With that said, it is still possible to reactivate an assessment. However, the dependencies are still an important factor, as no assessment can be (re-)activated with unmet dependencies. Moreover, a reactivated assessment automatically loses its probably set finished state, which may render other assessments into an unactivated state and thus cancels them.

### Expression Evaluator

As already mentioned in section 3.6.2, the evaluators are built up with the visitor-pattern-strategy in mind. That means, the evaluators were constructed with extensibility in mind. Adding more mathematical operators and comparing constructs just needs the addition of more visitor-methods. Nevertheless, the other visitor functions are refactored as well. That means, that the code gets tidied up by using newer language features (as described in section 3.6.2), replacing imperative code snippets with Linq functions or by combining similar functionalities in parametrized functions.

Especially the last two mentioned techniques reduced the lines of code tremendously as for example the simple mathematical operators additions, subtraction, multiplication or division are behaving similar in a way, as they are applying an operator to two numbers. Thus, outsourcing these four functions to one parameterized and applying it by Linq reduced every Expression Evaluation Function to a single line of code.

All in all, the whole Expression Evaluator was refactored the most in the Assessment framework. In some parts, this does not just improve performance, but also and



foremost maintainability. Especially with a such well planned and implemented project, it is crucial to keep up the tidy work, mostly to reduce the work time if further extensions are implemented.

## 4.2.2 Feedback

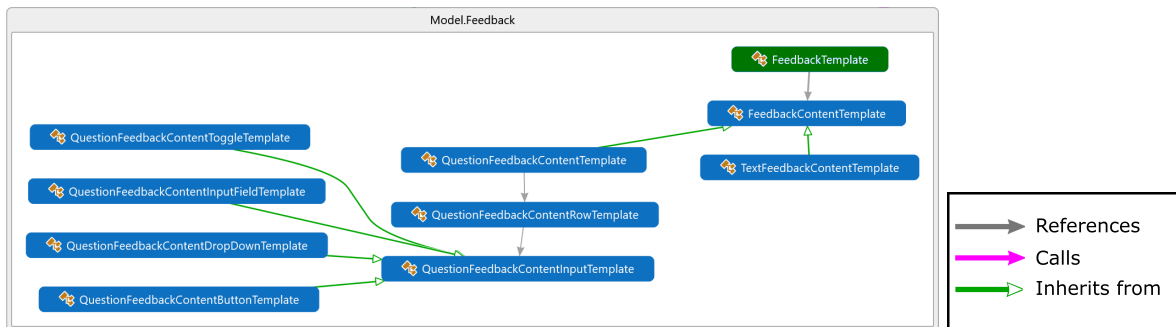


Figure 4.8: UML Diagram of the feedback part of the model. A *QuestionFeedback* holds many *QuestionFeedbackInputRows*, which further holds many *QuestionFeedbackInputTemplates*.

Feedbacks are constructed by defining a template in the XML model. A *FeedbackTemplate* contains a *FeedbackContentTemplate* from which the exact feedback type is inherited. At the moment, there is only one type of feedback, namely *TextFeedbackContentTemplate*. All inheriting classes of feedback carry an additional type flag which can be used by the GUI to adjust the appearance and behavior of the feedback. These types are defined by Dunwell, de Freitas, and Jarvis (2011) and are created from a pedagogical standpoint. To fully support the correspondence of assessment system and virtual environment, a “Technical” type was introduced too. With this type, it is possible to directly influence the behavior of the environment, rather than showing feedback to the user.

To be able to create a working assignment sheet with input for the user, the feedback needs to be extended with another class beside the *TextFeedbackContentTemplate*, which allows for a multi row user input definition. This class is called *QuestionFeedbackTemplate* and consists of an array of *QuestionFeedbackInputRows*. Every row may contain any amount of input fields. The different input fields and an example can be found in Table 4.3. An example of a fully defined question can be seen in Figure 4.10.

## 4 Development

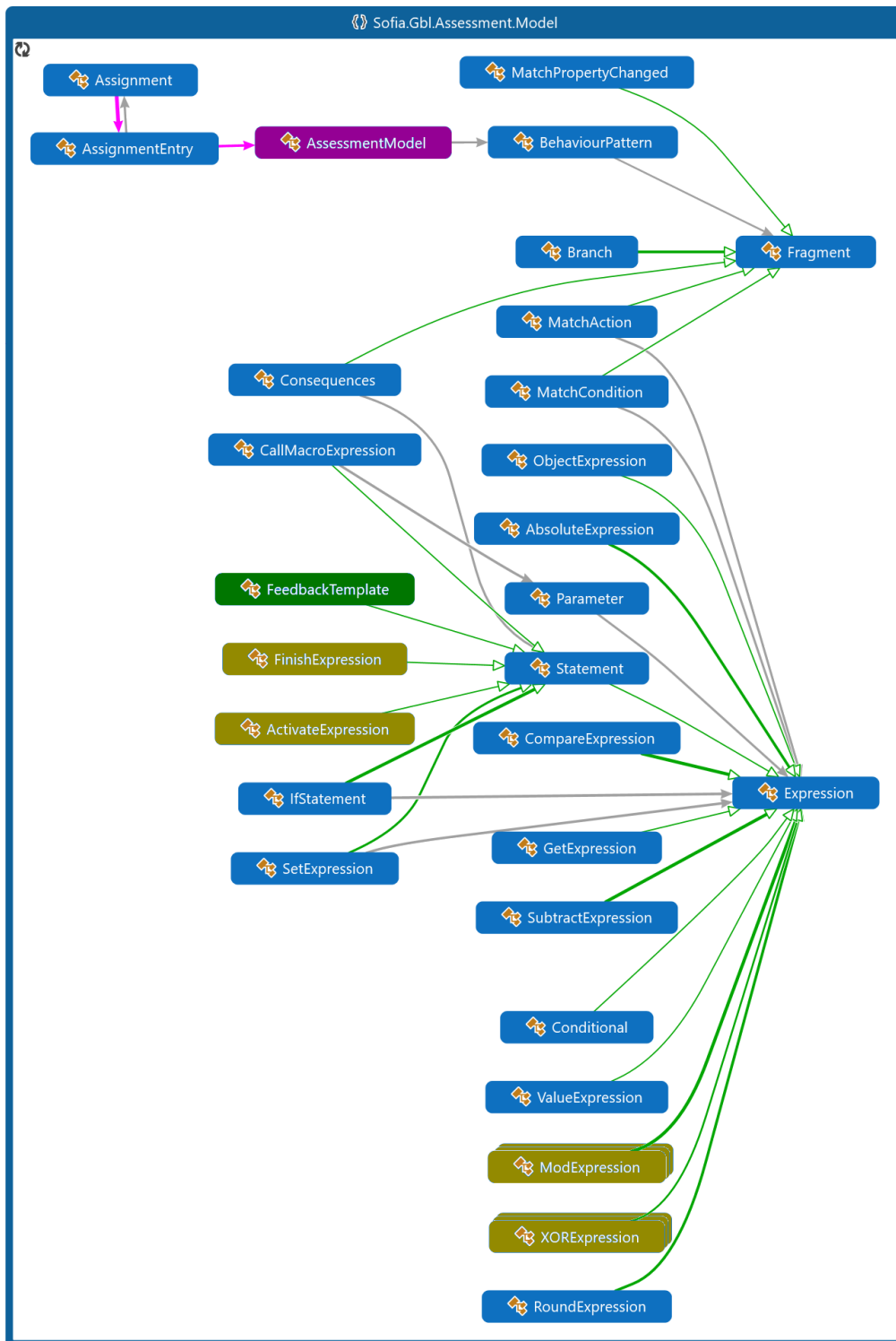


Figure 4.9: UML Diagram of the Model. An *Assignment* contains multiple *AssignmentEntries*. An *AssessmentModel* holds many *BehaviorPatterns*. Yellow boxes specify enhanced/added/altered classes. Multi-layered boxes specify multiple classes of the same type, for example Add, Subtract, etc. More detail of the feedback can be found in Figure 4.8. The legend can be found in Figure 4.8.

## 4 Development

Name	Example	Description
InputField	<input type="text" value="Enter text..."/>	A text box for arbitrary user input. It provides a hint text.
Button	<input type="button" value="Send"/>	A button to send the complete question.
Dropdown	<input type="text" value=""/> ▾	the drop down provides multiple predefined values for selection.
Toggle	<input checked="" type="checkbox"/> Toggle	A toggle box behaves like a radio button or a check box, depending whether a group is provided or not.

Table 4.3: The provided input fields by the question feedback. The design is provided by the Unity engine.

Exercise 1: calculate the Frequency of any pendulum Adjust the length and mass of the pendulum as you wish. Try to measure the period 3 times, calculate the frequency and insert your values here. I'm going to support you as best as I can.

<i>Length of Rope (cm)</i>		<i>Mass (kg)</i>
<i>Measured Time (s)</i>	<i>Calculated Frequency (Hz)</i>	Check
<i>Measured Time (s)</i>	<i>Calculated Frequency (Hz)</i>	Check
<i>Measured Time (s)</i>	<i>Calculated Frequency (Hz)</i>	Check

Figure 4.10: The first question of Assessment 2 presented to the student on the whiteboard.

## 4 Development

		Verbose	Evaluation resulted in suspend	9:57:44 AM 928
pendulum_assignment	9adba529-b2bc-406e-99ba-3ed720f5332d	Verbose	Adding scanned pattern to patterns in progress	9:57:44 AM 927
pendulum_assignment	9adba529-b2bc-406e-99ba-3ed720f5332d	Verbose	Evaluator scanned Step. Was Successful: False	9:57:44 AM 926
		Verbose	Evaluation resulted in suspend	9:57:44 AM 925
pendulum_assignment	9adba529-b2bc-406e-99ba-3ed720f5332d	Verbose	Adding scanned pattern to patterns in progress	9:57:44 AM 924
pendulum_assignment	9adba529-b2bc-406e-99ba-3ed720f5332d	Verbose	Evaluator scanned Step. Was Successful: False	9:57:44 AM 923
		Verbose	Evaluation resulted in suspend	9:57:44 AM 922
pendulum_assignment	9adba529-b2bc-406e-99ba-3ed720f5332d	Verbose	Adding scanned pattern to patterns in progress	9:57:44 AM 921
pendulum_assignment	9adba529-b2bc-406e-99ba-3ed720f5332d	Verbose	Evaluator scanned Step. Was Successful: False	9:57:44 AM 920
		Verbose	Evaluation resulted in suspend	9:57:44 AM 919
pendulum_assignment	9adba529-b2bc-406e-99ba-3ed720f5332d	Verbose	Adding scanned pattern to patterns in progress	9:57:44 AM 918
pendulum_assignment	9adba529-b2bc-406e-99ba-3ed720f5332d	Verbose	Evaluator scanned Step. Was Successful: False	9:57:44 AM 917
		Verbose	Evaluation resulted in suspend	9:57:44 AM 916
pendulum_assignment	9adba529-b2bc-406e-99ba-3ed720f5332d	Info	SetProperty SetValue: PendulumManager.RopeLength = 0.3	9:57:44 AM 915
pendulum_assignment	9adba529-b2bc-406e-99ba-3ed720f5332d	Info	SetProperty SetValue: PendulumManager.Weight = 1	9:57:44 AM 914
pendulum_assignment	9adba529-b2bc-406e-99ba-3ed720f5332d	Info	SetProperty SetValue: Transform.Rotation.X = 0	9:57:44 AM 913
pendulum_assignment	9adba529-b2bc-406e-99ba-3ed720f5332d	Verbose	Processing event	9:57:44 AM 912
pendulum_assignment	9adba529-b2bc-406e-99ba-3ed720f5332d	Info	EvalService received a new Event	9:57:44 AM 911 <a href="#">Show</a>
		Verbose	[Pendulum Maroon 3] Adding behaviour EnterSection	9:57:44 AM 896
		Verbose	[Pendulum Maroon 1] Adding behaviour AnswerQuestion	9:57:44 AM 895
		Verbose	[Pendulum Maroon 1] Adding behaviour Sofia.Gbl.Assessment.Model.MatchCondition	9:57:44 AM 894
		Verbose	[Pendulum Maroon 1] Adding behaviour UseObject	9:57:44 AM 893
		Verbose	[Pendulum Maroon 1] Adding behaviour UseObject	9:57:44 AM 892
		Verbose	[Pendulum Maroon 1] Adding behaviour UseObject	9:57:44 AM 891

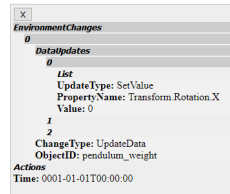


Figure 4.11: The logger view of the WebService with activated tree view of an environment change event. The tree view is activated by clicking on the “show” link beside the event.

### 4.2.3 Logging

While probing the model and the evaluator by altering the existing XML files, as well as during the creation of the new assignment sheet, it became obvious that a thorough feedback for the expert is needed in order to be able to create models efficiently. There are many alternative methods available for solving these problems.

As there is already a web service and thus a running web server as a first entry-point for all client applications, the decision fell on a socket based website to provide real time logging and log observation of the happenings inside the evaluator. Filters are provided and a special tree view for objects to observe all events and feedback sent from and to the evaluator. Figure 4.11 shows an example view of the logger including a tree view of an *EnvironmentChange* object.

### 4.2.4 Model

To provide a working showcase, the model for the experiment needs to be defined as well. As mentioned before, the model now consists of two parts, the assignment and the assessments. As assessments can now access variables of before finished assessments too, the existing models had to be completely redone. To improve the readability of the XML, the hierarchy structure was kept as flat as possible. For the assignment, it now contains of 3 assessments: one to introduce the experiment,

## 4 Development



Figure 4.12: A hint text during assessment 1 of the assignment “Pendulum Experiment”. Small pictographs underneath the objects give a hint on the interaction.

one to check the understanding of measuring the period of any pendulum and the last one deals with the subject of the relation between rope length, weight and the pendulum frequency.

Assessment 1 introduces the experiment by exactly declaring how the controls behave and how to handle the different subjects provided in the scene. To progress successfully in the scene, the student needs to repeat after the GUI text provided, which is deflecting the pendulum, increase the weight, increase the rope length, start the stopwatch, stop the stopwatch. To finish this assessment, a button on the assignment sheet needs to be clicked. An example output of the GUI during this part of the assignment can be seen in Figure 4.12.

Assessment 2 provides help during measuring any chosen pendulum. That means, that the student may choose any rope length and weight and needs to measure the correct period three times. During this process, the angle of deflection is monitored, as well as the correct rope length, weight, measurement of a period and calculation. After successfully measuring three times the period and calculating the related frequency, a final question is asked, where the student needs to calculate the average value of the three frequencies. This assessment should help the student to understand, how a period is measured reliably.

Especially in Assessment 3, the knowledge on how to correctly measure a period will be useful, as this kind of help is not provided anymore. The student needs to choose any rope length and weight at first, then needs to adjust just one of these two factors, and for a third measurement the second one. After successfully doing that, two questions will be presented to check the gained knowledge if and how the different factors are influencing the frequency.

### 4.3 Usage

This section describes how the pendulum experiment is used and which features are provided. Furthermore, the usage of the *UnityInterface*-components are explained and how they behave. Last but not least, the functionality of the logger is explained and some useful tips are provided, how to efficiently filter messages.

Figure 4.13 shows some scenery during the experiment. The intro part guides the user through the controls of the system (Figure b and c), Figures d and f show an example mistake message during part 2, Figure e shows an example hint text during part 2, and Figure g an example Success message during part 2. These messages might happen during part 3 as well.

#### 4.3.1 Pendulum Experiment

The goal was to provide an easy-to-use experiment by making every object clickable wherever possible. All the input possibilities are represented with small pictographs underneath the object, which can be observed in Figure 4.12. The pictures represent mostly the mouse and which button to be used to trigger the according action. The calculator is handled exactly the same as any on-screen calculator. All the buttons are clickable as well as mapped to the numpad of the keyboard.

The GUI provides a small button to discard a single message. This is just for convenience reasons, if the GUI covers input fields on the assignment sheet. The assignment sheet can be interacted most of the time with mouse as well, but text input fields need to be activated via the mouse first to accept input. If an input field is active, it is not possible to leave the experiment (which is done via the space key). To change the language or general settings the Esc key needs to be pressed, to bring up the menu as can be seen in Figure 4.14.

When deflecting the pendulum via drag-and-drop with the mouse during Assessment Part 2, 3, and 4 (see section 5.2 for further details on the description of the



## 4 Development

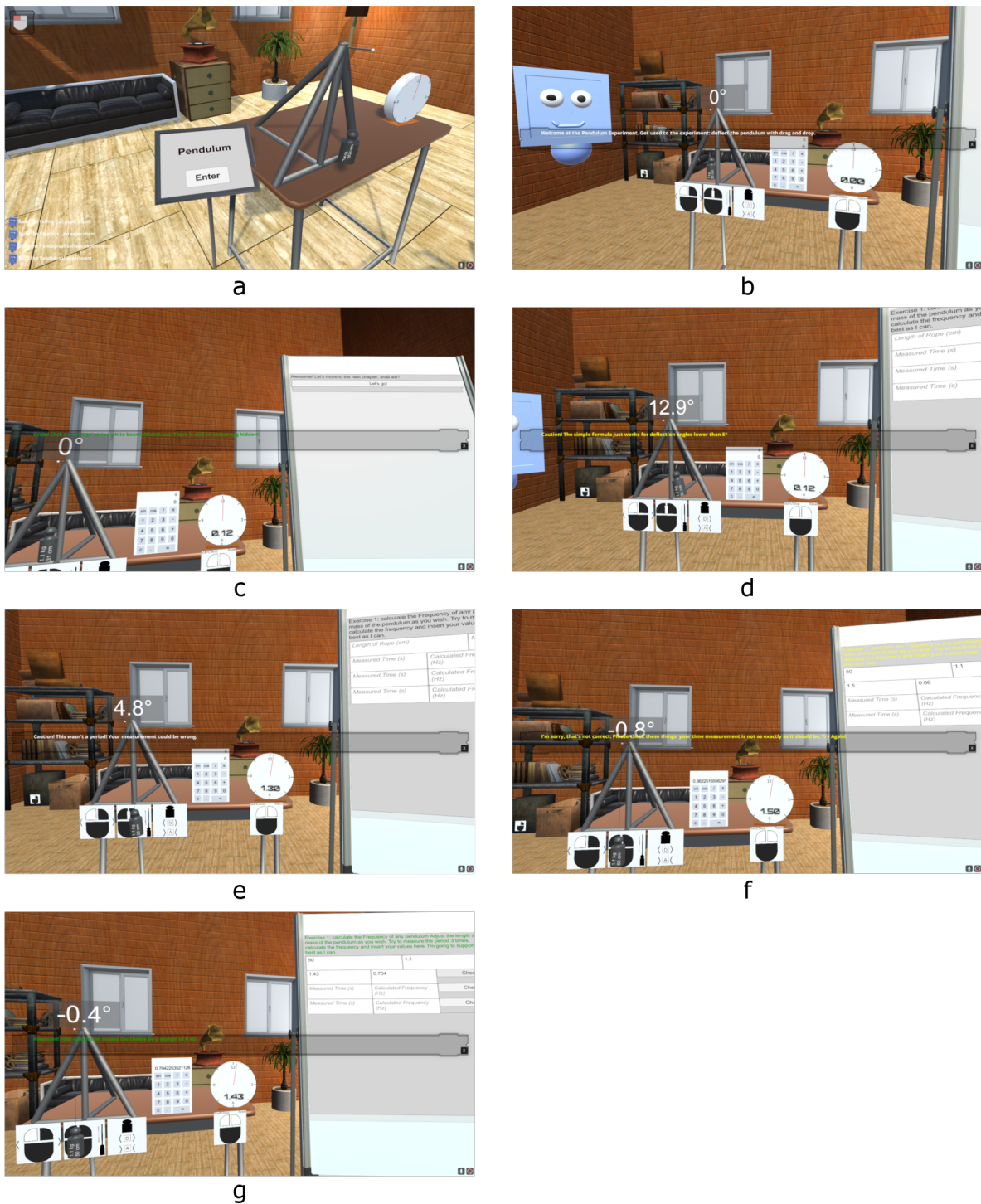


Figure 4.13: Different scenes during the play-through. Figure a: Enter pendulum experiment, Figure b: Beginning of intro part, Figure c: end of intro part, Figure d: mistake message, if the pendulum is deflected too far during part 2 and 3, Figure e: hint message, if measurement was not a full period, Figure f: mistake message on wrong calculation, Figure g: success message on correct calculation

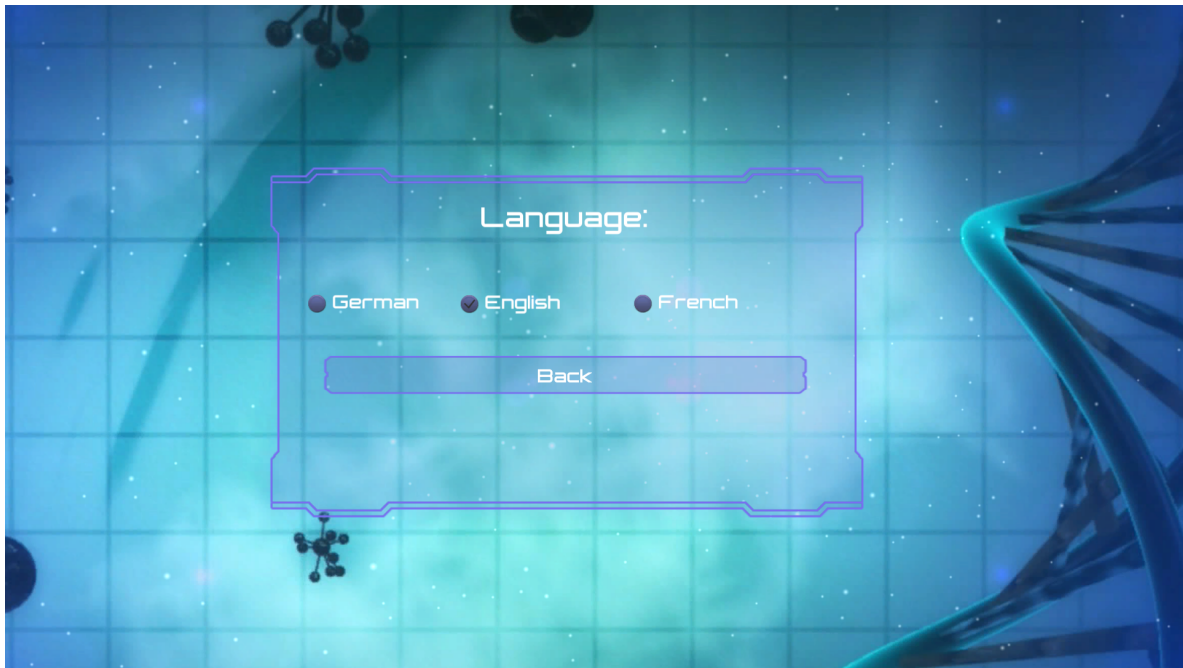


Figure 4.14: The language settings provided by the menu.

tasks), the environment sends a release event to the assessment system (and all monitored environment variables). Because of the model definition, the deflection angle is checked at the assessment system to not exceed five degrees. If so, a feedback message will be sent back to the environment and presented to the user. Immediate feedback like this happens throughout the assessment and can be observed multiple times in Figure 4.13.

### 4.3.2 Unity Interface for Assessment

The Unity Interface consists of 3 parts: a Library for the Web Service inclusion (a DLL-file), the *AssessmentManager* and the *AssessmentWatchValue*. All three of them need to be used in order for the assessment system to work. As described in section 3.6.1, the assessment system communication is realized through a web interface. To avoid crashes of Maroon, the communication is encapsulated inside the interface pattern of the DLL-file. Thus, if the server communication fails, Unity does not cancel the execution of Maroon, but rather the interface catches the exception and disables the assessment communication. This greatly increases stability and usability, as the hassle of an unstable connection and its associated problems are removed from the experiment expert.



## 4 Development

Furthermore, the DLL-file does not only provide the necessary interfaces for the communication with the assessment system, it also provides the *GameEventBuilder* class, which helps with the creation of the game events. As the experiment expert always has to do a little bit of programming for the experiment in Maroon, the decision was made to provide a programming interface for the creation of the game events. This way, the expert is able to send an event whenever needed without restrictions to events. This is especially helpful when programming with event listeners and related triggered events. After creating an Environment, the provided "Send" command is used, to send the event to the assessment system. This command returns any Immediate Feedback which might occur during this process. With that said, the expert does not need to take care of gathering all the environment changes, as this is done by the *AssessmentManager* and *AssessmentWatchValue* classes.

These two classes are used by firstly adding the *AssessmentManager* class to any object in the scene. An example can be seen in Figure 4.15 on the left side. This component provides three fields to define the Assessment server URL, which assignment should be started and the entry point which should be loaded by default. This field may stay empty, although this way it will not initialize the assessment system. To achieve that nonetheless, the expert would need to programmatically activate the entry point by an *EnterSection* command. Furthermore, a "Global Attributes" field is provided, to monitor variables provided by the Unity environment, which are not directly correlated to any object in form of a component, such as the gravity magnitude for example.

Secondly, the *AssessmentWatchValue* component may be used multiple times on any object of the scene to monitor the given values. These values are defined by an object-oriented programming style as can be seen in Figure 4.15 on the right side. When referring to the picture as an example, the object on which this component was attached to, also carries a *Transform* component, which provides a *Rotation* vector, where the X coordinate is monitored. As this component registers itself to the Assessment Manager on load of the scene, every time an action is send to the assessment system by the experiment expert, all these variables will be sent with it as an *EnvironmentalChange*. Figure 4.16 shows a time line, how the resolution of these provided values is done. When the scene is loaded, the values are dereferenced through reflection and stored programmatically, as the resolution of properties with reflection seems to be computational expensive. The resolved parental tree is saved and gets traversed every time the value is requested. The same technique is used to resolve the global values on the Assessment Manager.

## 4 Development

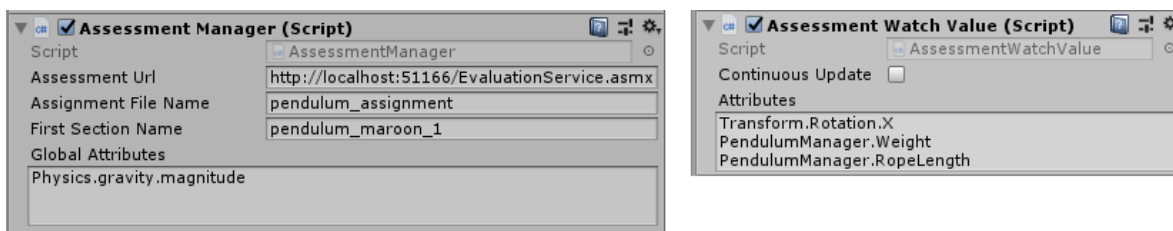


Figure 4.15: Left: The Assessment Manager which holds the information about the server and the first section to be entered, right: The Assessment Watch Value, which may be used multiple times and holds all the variables which should be observed of the object, on which it was added. (see Figure 4.11)

### 4.3.3 Logger

An example view of the logger can be found at Figure 4.11. The logger provides 4 basic inputs: The “Clear All” button, the “Verbose” toggle box and the “Refresh all” button on the left side as well as the Filter search bar on the right hand side. Basically, the logger fetches all stored messages on the server and represents them in a tabular format on the screen. Furthermore, a web socket connection is created to the server, to populate new messages on time. By default, the columns are sorted by the date time of the entry in descending order. If the “Verbose” toggle box is active, all new messages including the Verbose type messages are received. This may lead to performance issues on the client as the amount of messages may exceed the capable amount of the browser very fast. Because of that, it is recommended to deactivate this check box, which automatically leads to a clear and refresh of all except the verbose messages, and registers this setting at the server. If a new verbose logger message is generated on the server, it is only sent to the client, if it is marked for receiving those messages (check box is active). This technique greatly improves performance on the client side.

When the “Clear All” button is pressed, all messages on the server are cleared. That means, all connected clients will lose all messages too. This is needed if the server is running for a longer time. The “refresh all messages” button is used to force a refresh of the current view. Especially with a large amount of messages this might help to get a current working JavaScript routine to exit its loop.

All column headers of the table may be clicked to sort the view in ascending or descending order according to this column. The filter function checks all cells to find the given value. Because of that, Table 4.4 lists some useful filters for particular cases. It is important to mention, that space between words are treated as “OR” not as “AND”, which means the search term “set value” will find any phrases containing (case insensitive) the words “set” or “value”.

## 4 Development

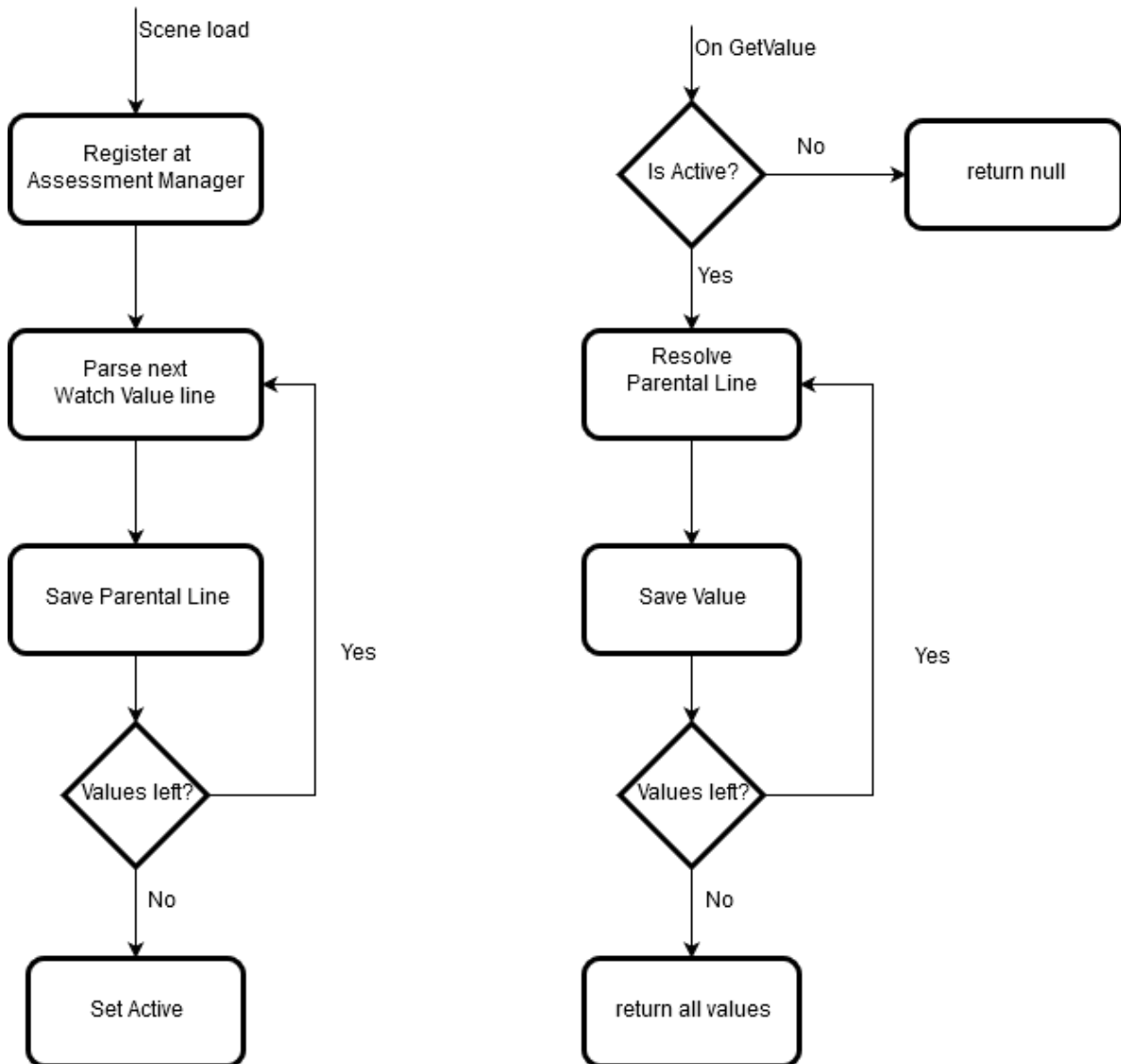


Figure 4.16: Left: On scene load, the Assessment Watch Value gets initialized by registering at the Assessment Manager and resolving the parental lines of the given values, Right: every time *GetValue* is called, the previously saved parental lines are resolved and the values are saved.

Name	Verbose	Description
Show	FALSE	As generated objects are observable by clicking the button "Show", this filter shows all lines, which contain an object. Especially useful to check the received events or the sent feedback.
SetValue	FALSE	Check the set values which might be due to an environment change (received event) or through a calculation according to the active assignment.
Sending	FALSE	Similar to the "Show" filter but only for feedback.
Received	FALSE	Similar to the "Show" filter but only for events.
[	TRUE	Check the general actions performed by the model. Very useful for debugging during creation of a model.

Table 4.4: Suggestions and examples of filter key words for the logger.

## 4.4 Summary

This section went into detail how the implementation of the pendulum experiment inside Maroon was performed. Stopwatch, calculator and assignment sheet were created modularly to fit inside a prefab to be usable in general. The code of the pendulum itself greatly reduced due to the fact, that Unity provides a physics engine. The assignment sheet provides multiple input types such as button, text box, toggle box (which acts as a Radio button when defined with a group name or as a check box if not) and a drop down. All of the feedback, which is the GUI and the assignment sheet, support three different colors (white, green and yellow) to represent the urgency and type of a feedback. Green for success messages, white for hints and yellow for mistakes. Furthermore, technical feedbacks are used to control the GUI via commands from the assessment system.

The assessment system provides an interface for Unity to encapsulate the server communication. This enables failsafe encapsulation of possible exceptions as well as simplified commands for the experiment experts. Furthermore, the assessment manager and the assessment watch values are monitoring specified values on their own without further interaction needed by the expert. For sending events, a very simple to use *GameEventBuilder* is provided to create the given type of event.

## 4 Development

The assessment system got enhanced by the possibility to assess complete assignments, which consists of multiple dependent assessments. Single assessments can be activated and finished through the assessment itself to handle the dependencies. Assessments which dependencies are met are only partially loaded with the behavior patterns, which contain an *EnterSection* command. An assessment needs to be activated to be fully loaded. Furthermore, assessments got enhanced with two main properties: richer selection of mathematical operations and a second type of feedback. The first enhancement provides the possibility to perform more calculations inside the model, whereas the question feedback was one of the main requirements – not only assess the doings of the student, but the gained knowledge too.

Now that the development and implementation of the experiment and the enhancements of the assessment system are done, the evaluation of the usability is the next main focus. To achieve that, development independent students are asked to use the created prototype.

# 5 Evaluation

The aim of this chapter is to show, that the problem definition has been addressed. Thus, the research questions are stated. After that, the target group is introduced, the evaluation results are presented and findings discussed.

## 5.1 Aim and Method

To ensure, meeting the problems described in section 1.2 have been addressed, an evaluation is conducted. The goal was to achieve an easy-to-use system for students aged 10 to 14. The system should provide enough information to support the student during the experiment, but should not give away too much information to ruin the learn effect. Thus the research questions are:

- Are the students staying focused?
- Is the system able to deliver the content and feedback correctly?
- Is the system appealing enough to be used multiple times?

As can be observed, the focus lies specifically on the usability of the system from a student standpoint. It is crucial to know, if the system in its current form provides enough support to engage students while not spoiling the learning effect. Thus the planning of the assignment is a crucial part.

## 5.2 Pendulum Experiment Assignment

The assignment consists of 4 Parts with two different configurations for group A and group B. For simplicity, the parts here are explained first and after that, the specific configuration per group is shown. Part 1 was an introduction on how to use the system. It was a basic how to, given by the feedback system, on how to deflect the pendulum, how to increase the rope length, change the mass and use the stop watch.

## 5 Evaluation

In part 2, the task was to calculate the frequency of a freely chosen pendulum 3 times, with freely chosen meaning the rope length and the mass are chosen by the student. During this task immediate feedback was given for correct angle of deflection, wrong rope length despite the set rope length, wrong measuring in despite of the period (which means there was a hint text if start and stop of the stopwatch have not been the same point, so it was not a full period) and wrong calculation, which includes imprecise time measuring. After successfully measuring 3 times, the average of the frequencies had to be calculated.

Part 3 handled the relation between rope length, mass and frequency. The student had to choose between 3 different pendulum settings, whereas they only may differ in one parameter at a time. That means measurement one and two need to have the same rope length or the same mass in common, but not both. Same goes for measurement two and three. To measure the period and thus calculating the frequency the technique was the same as in part 2. After doing the measurement 3 times correctly, two additional questions popped up to ask for the conclusion. These two questions were "How does the rope length impact the Frequency? The longer the rope, the..." with three answers "higher the frequency", "lower the frequency" and "[no correlation]". For mass it was similar.

Part 4 was added, to check whether the feedback system had an impact at all. For that, part 2, the frequency measuring with any pendulum, was executed but without any feedback at all. The measurements were not written down onto the white board of the experiment but on the questionnaire form for comparing reasons. The answer setup on the questionnaire was the same: one time rope length, one time mass, three times the measured seconds for one period, three times the calculated frequency. The student did not get any feedback at all during this part, neither from the system nor from the experimenter.

To execute the A/B Study, Group A did the parts 1 2 3 and 4 in this exact order, which means, the no-feedback-part was performed last, after the similar part (part 2) with feedback has already been performed with feedback. In contrast, group B executed the parts in the order 1, 4, 2 and 3 which means, the experiment without feedback was performed, before any direct feedback has been given. This should help to provide some insight, if and how the system influences the understanding of the problem. In both groups, every student had 1h30min time to finish the tasks, which just one person could not fulfill.

## 5 Evaluation

Group	Person#	Sex	Age	School Type	Computer Interested
A	1	F	10	NMS	2
A	2	M	13	NMS	4
A	3	M	14	Gym	2
A	4	F	13	Gym	5
B	1	M	12	Gym	5
B	2	F	13	NMS	4
B	3	F	12	Gym	3
B	4	F	11	Gym	3

Table 5.1: The 8 test persons and their demographic data. F = Female, M = Male, NMS = “Neue Mittelschule” (“New Middle School”), Gym = “Gymnasium”. Computer interested is a self-assessed value ranging from 1 = not at all to 5 = absolutely

### 5.3 Target group

The target group was chosen to match the group the program was made for. As Maroon aims to be a learning software for middle school as well as the pendulum experiment is according to the supervisor part of the learning material of physics in middle school grade, children between the age of 10 and 14 were chosen to be accurate experimentee. The age as well as the current school was tracked, but not their names, home addresses, e-mails or other sensitive data to ensure data security.

The survey was conducted at a local path finder group house in Graz over nearly one month, with one test person at a time (supervised conduction). This process was done during multiple scheduled routine courses the path finder do weekly with the attendees. As mentioned before, the chosen target group has been children between 10 and 14 years old, to match the age of middle school in Austria and the associated school curriculum for the lecture “Physics”. To conduct this survey in a city like Graz and not inside a school but rather in an independent institution like the path finders results in a thorough mixture of students from different school types and pre-knowledge, as can be seen in Table 5.1.

### 5.4 Instruments and Setup

To perform a meaningful comparison of a virtual learning environment system with immediate feedback, it should be compared with a system without immediate feedback. Because of that, the choice fell on an A/B-Study. After conducting the



study, the means are taken to compare the results. A/B Studies are usually made to compare the impact of newly created features. The test group is split in half, whereas group A is commonly referred as the “control half”, group B as “treatment half”. There are multivariate setups, also called as A/B/n studies (Kohavi & Longbotham, 2017).

In this particular case, the choice was made to perform the experiment with the assessment system and a part of the experiment without the immediate feedback. Group A had to do the whole experiment with feedback and afterwards the frequency calculation without feedback again, whereas Group B did the intro part with feedback (to get used to the handling of the system), then the frequency calculation without feedback and then carried on with the experiment including feedback.

The student received an info sheet, which provided all information needed in regards of successfully fulfilling the experiment. This sheet was accessible throughout the whole evaluation for consultation. After every part, a Google Forms was used for evaluation and the data was saved to a Google spreadsheet to do the analysis. Furthermore, the student used a laptop provided by the author and a tablet to do the questionnaire, also provided by the author.

The experiment itself was on fullscreen, whereas the laboratory part was skipped by pre-building up the experiment in the laboratory such that the start was just clicking on the experiment instead of getting used to the lobby of Maroon. Also, the server was started locally on the machine to provide full flexibility even in environments without direct Internet access. The assignment was set to be accomplished in maximum one and a half ours, which was achieved by all but one students.

The evaluation of the results is done using the System Usability Scale (SUS) (Kay & Loverock, 2008) and Computer Emotional Scale (CES) (Brooke, 1996). Additionally, system specific questions were asked at the end of the survey. To evaluate the results, the proposed methods are used. The Test questions in German and English can be found in section 7.2 and section 7.2.

### 5.5 Findings

With SUS and CES it is possible to summarize the findings by reducing to the categories of CES as can be seen in Table 5.3 (Kay & Loverock, 2008). However, this is not possible with the five free questions provided, although they follow a similar structure. The results of these questions can be found in Table 5.2.

## 5 Evaluation

Original Question	Translated	A Avg (SD)	B Avg (SD)	Avg (SD)
Die Hilfestellungen des Systems waren sehr hilfreich.	The assistance of the system were very helpful.	3.25 (1.50)	4.25 (0.50)	3.75 (1.16)
Ich empfand die Hilfestellungen störend.	I think, the help was disturbing	2.25 (1.50)	1.00 (0.00)	1.63 (1.19)
Die Hilfestellungen waren ausreichend detailliert.	The assistance was detailed enough.	2.5 (1.73)	3.5 (1.73)	3.0 (1.69)
Meiner Meinung nach wäre Feedback am Ende des Experiments hilfreicher gewesen, anstelle ständiger Hilfestellungen.	In my opinion, feedback at the end of the experiment would have been more helpful, instead of constant help.	2.75 (1.71)	1.00 (0.00)	1.88 (1.46)
Ich bevorzuge ein System ohne ständigem Feedback.	I would prefer a system without immediate feedback.	1.75 (0.96)	1.00 (0.00)	1.38 (0.74)

Table 5.2: Answers to the five free questions. 1 = “strongly disagree”, 5 = “strongly agree”

Emotion Construct	Feelings
Happiness	Satisfied, Curious, Excited
Sadness	Disheartened, Dispirited
Anxiety	Anxious, Insecure, Helpless, Nervous
Anger	Irritable, Frustrated, Angry

Table 5.3: Feelings and their corresponding emotion constructs according to Kay and Loverock (2008)

All in all, the system was overall appreciated, as can be seen in Figure 5.1. With that said, the anger and anxiety still was quite high, which has been observable already during the survey. The reason for that is mixed, as multiple students moaned about the task itself, whereas others struggled with not enough information provided by the system.

The results of the System Usability Scale can be found in Table 5.4. These results underline the overall positive impression of the system. As all participants were German native speakers the feedback questionnaire was just provided in German. The translated version can be found in the appendix whereas the translation is an “as-close-as-possible”. The information sheet can be found there also, but was not translated to English at all. The results of the feedback questionnaire are discussed in the following sections.

## 5 Evaluation

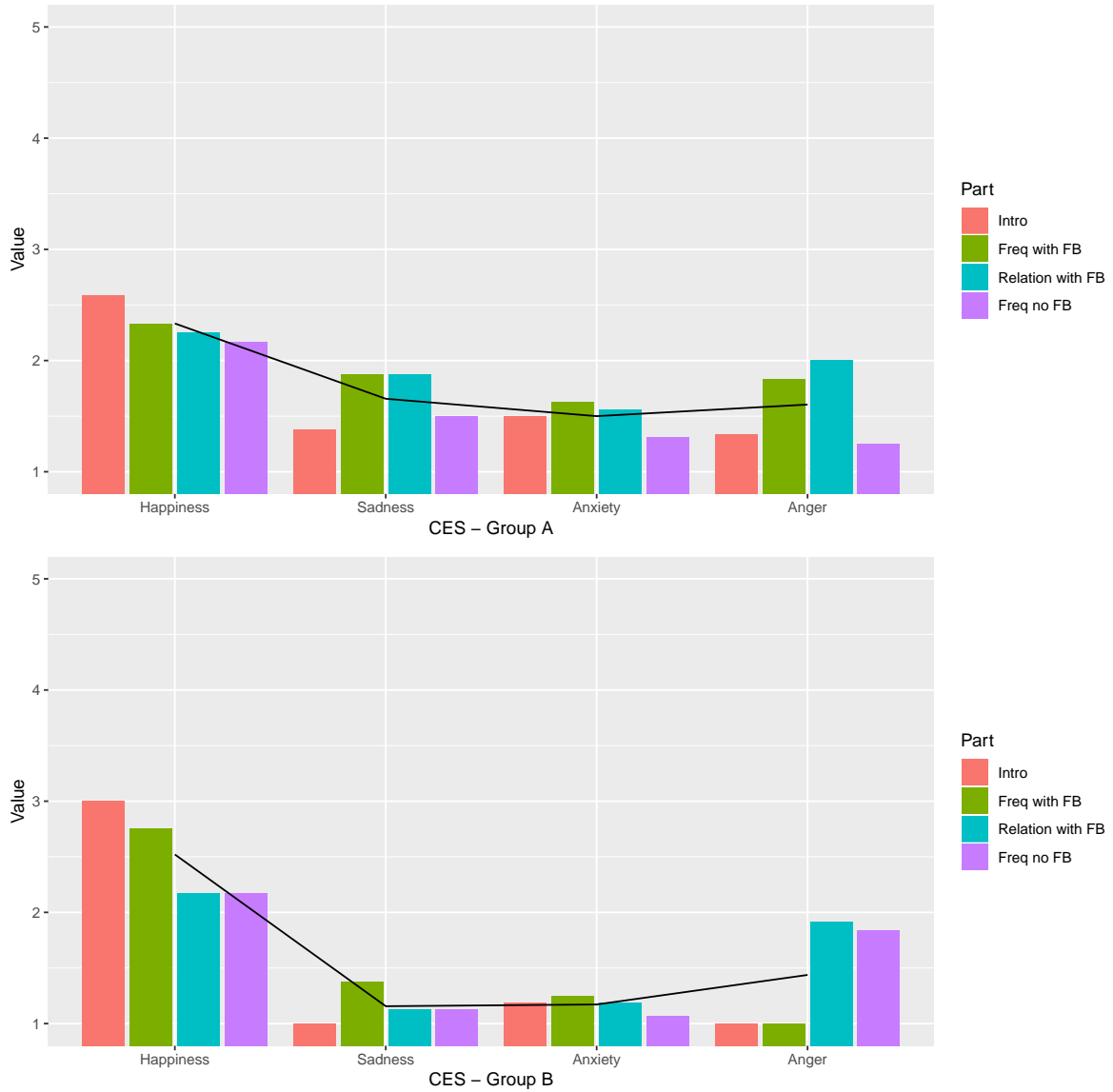


Figure 5.1: Overview of the CES scale by averaging over all parts. The line represents the average of the four parts represented by the bars. Top graphic represents group A, the bottom one group B.

## 5 Evaluation

Group	#	SU Score	SU Rating
A	1	50.00	OK
A	2	81.25	Excellent
A	3	50.00	OK
A	4	96.88	Best Imaginable
Average: 69.53/ <b>Good</b>			
B	1	86.25	Excellent
B	2	86.25	Excellent
B	3	70.00	Good
B	4	90.00	OK
Average: 83.13/ <b>Excellent</b>			

Table 5.4: The summary of the SU scores. It was calculated by using the proposed calculation method as described by Brooke (1996) averaging over all four parts per student.

### 5.5.1 General Response

The pilot test showed multiple problems which had to be addressed before starting the full survey. The biggest issue was, that the instructions on how to use the system were not clear enough. This is the reason why part 1 was introduced as the first assessment for both groups. It seemed to be really helpful, given the fact how people had to fiddle with the usage of the system, even though it was textually described and illustrated with pictographs. Nonetheless, the first part was rated as very useful (median 5 and 4 for group B and A respectively), especially by children which stated to use computers more often for gaming.

The response for the immediate feedback given was very positive. During the tests it was easily observable that students never felt to mess around but rather stick to the task, even though it took about one hour in average. It seems that the motivation was much higher to satisfy the assessment system then the inner force to play and mess around with the system. Furthermore, there was a lot of positive response, as the info sheet, although just one page long with a very big picture of a period (see section 7.2) was obviously not read thoroughly, as nearly every child deflected the pendulum more than  $5^\circ$  at first.

In the provided free text field for positive feedback, nearly all of the participants stated, that the general appearance was appealing. Three stated, that the feedback was really helpful to fulfill the tasks given. More details can be found in Table 5.5. Suggestions for improvements can be found in subsection 5.5.3ff.

## 5 Evaluation

<b>Group</b>	<b>#</b>	<b>Original Text</b>	<b>Translated</b>
A	1	Der Hintergrund. Der Feedbackbogen war sehr klar.	The background. The evaluation form was clearly structured.
A	2	Der Raum ganz am Anfang	the first room [laboratory]
A	3	-	-
A	4	Selbstständiges Arbeiten und Ausprobieren!	Independent working and free roam!
B	1	Die Hinweise während des Experiments waren sehr hilfreich und leiten den Benutzer sofort auf den richtigen Pfad	Hints during the experiment have been very helpful and lead to the right path immediately.
B	2	Animation, Design, Benutzerfreundlich	Animation, design, easy-to-use
B	3	Der Lerneffekt, spielerisches Lernen	learning curve, playful learning
B	4	schöne Umgebung, texte waren lange da	pretty surrounding, feedback text was visible long enough

Table 5.5: Positive free text answers at the end of the survey. Text has been translated from German to English by the author.

### 5.5.2 Comparison A/B Group

The main goal of the A/B case study was to find the impact of the feedback on the gained knowledge. This is achieved by differently ordering the tasks for the groups. Group A measured the frequency without feedback as last task (Task 4). Group B was executing the no-feedback-part first and nearly all of them failed to measure even closely the correct time of a period, or even measure a period at all. All of that, while the confidence rating was very high at the same time. It seemed, that just barely reading about the theory created enough confidence about a practical problem. The measured results can be seen in Table 5.6.

Another difference between the two groups was the deviation between the CES values in all four groups (as can be seen in Figure 5.1). This mainly happened because two students in group A may be considered outliers with respect to a mean/standard deviation evaluation. One female student rated nearly every question opposite then every body else, whereas the other male student agreed with her rating quite often. With a test group of four persons this leads to an overall distorted result, as there are as many outliers as mean data points.

### 5.5.3 Emerged Problems

*"When you want to know how things really work, study them when they're coming apart."*

- William Gibson

All of the conclusions are derived from either the negative feedback as can be seen in Table 5.7 or because of the observations during the experiment. During the tests, some usability problems emerged. The nature of these causes had been most times a too indistinguishable display of messages. One of the most prominent mistakes was during the usage of the pendulum, the same error message could emerge multiple times. The best example for that was the incorrect measurement of one period. The first time, a measurement was made incorrectly, an error message stated, if the measurement has not been a full period. If the measurement was redone shortly after, again not correctly, the reappearance of the message could not be distinguished from the first appearance. This lead to the problem, that students thought, the second measurement was correct, as nothing changed. This problem could be overcome by introducing an animation to the GUI on appearance, to suggest that there was actually a change.

A second very prominent problem was that error messages were just read once. This in particular was a problem during the answering of the assignment sheet and

5 Evaluation

Group A				Group B			
<i>Rope Length (cm)</i>	<i>Calculated (Hz)</i>	<i>Theoretical (Hz)</i>	<i>Delta (Hz)</i>	<i>Rope Length (cm)</i>	<i>Calculated (Hz)</i>	<i>Theoretical (Hz)</i>	<i>Delta (Hz)</i>
50	0.66	0.70	0.04	20	1.08	1.11	0.03
45	0.69	0.74	0.05	30	0.29	0.91	0.62
50	0.66	0.70	0.04	40	1.36	0.79	0.57
40	0.76	0.78	0.02	40	2.81	0.79	2.02

Table 5.6: Results of the measurements of the frequency without feedback performed by 4 students per group. Group A performed this task after completing it with the feedback, group B beforehand.

checking the results. The error message shown was a modular message, containing one or more text phrases what could have gone wrong. After a small amount of occurrences, the text was not read anymore. This led to the problems, that students sometimes just entered the wrong time measurement in the calculator, very probably because of typo, and thus calculated a wrong frequency, although everything else was correct. The resulting error message did state that exact problem. Nevertheless, students just expected a wrong measurement (as it happened multiple times before), did not read the message at all and started to remeasure again. For one girl this was a very common problem which lead to high frustration in her case. This problem could be most probably overcome by sound output which distinguishes especially at the beginning, according to the text shown or by a completely different layout of the text.

Beside these two prominent problems, some minor problems evolved but sometimes not directly according to the system itself, but rather because of misunderstanding by the students and possible lack of support by the system. One child claimed to not have learned about basic mechanical principles in school up to now. In this case, a lot of additional info had to be provided additionally to the information paper and she struggled during the frequency calculation a lot. In general, much more information should be provided by the system after multiple mistakes in a row. Table 5.8 shows multiple instances of suggestions by the students in this regard.

Another problem was that one child forgot the controls during the experiment. This is in particular a very hard problem to overcome, as it highly depends on the user

too, how to handle this problem. As can be seen in Figure 4.2 and Figure 4.12, small plates which shows pictographs of a mouse and buttons were modeled to remind the user on the controls. The girl stated during the test by thinking aloud, that she misinterpret these pictures, whereas she has been the only one doing so, especially after the introduction part. To probably overcome this problem, the introduction part could be made with animations, to directly show the user, how to handle the system. Nevertheless, this problem could be outside othe domain of the system as well, as it reduces to the fundamental problem of using a computer.

As a basic rule of thumb it seems to be safe to state, that reading text is in general a problem, especially for user which are not fully intrinsically motivated to solve the given task. This results in even higher frustration because of wrong doings and thus even less willingness to read the text at all. One child honestly stated, that it was important, that there was feedback as it didn't read the info paper at all. If the feedback would have been with voice output and text, it would probably resolve most of the occurred problems. Matching this fact with an even more sophisticated model which adapts even further to the skill level of the user, this system should provide a very decent and broadly accepted learning platform. The possibilities of how to improve this current systems are discussed in more detail in chapter 7.

### 5.6 Summary

The evaluation of a newly created application is a vital part to gain knowledge about the acceptance of the system. In this section, a discussion about the findings has been made. Two prominent problems arose during the testing. The first was a indistinguishableness between same error messages, the second one was that similar messages were only read once as the style was the same, although the content of the message was different. Both problems may be overcome quite easily in Unity and are not directly a problem of the model.

Other problems were Model related, for example firstly, the introduction part, which was sometimes not thorough enough or there was no possibility to get an even deeper explanation on how the experiment works and should be handled. Secondly, it seemed to frustrate students, if the mistake message stated "this is not precise enough" although the measurement was far off. A third observable problem was the fact, that the model did not adapt enough to the needs of the child as some children were perfectly fine with the amount of feedback, bot others clearly needed more detail information about there no good doings. Chapter 7 discusses about the possibilities of how to possibly tackle the mentioned problems.



## 5 Evaluation

<b>Group</b>	<b>#</b>	<b>Original Text</b>	<b>Translated</b>
A	1	Der Rechner ist zu klein. Der Zeitlupenbutton ist schwer ersichtlich.	Calculator is too small. The time laps is hard to find
A	2	Rechnen	Doing calculations
A	3	Das Programm hilft mir nicht, zu verstehen, was wirklich passiert..	The program doesn't help me to understand, whats really happening
A	4	-	-
B	1	Taschenrechner akzeptiert Textinput wenn Werte in die Lösungsboxen eingetragen werden, Zuruecksetzen des Pendels koennt einfacher gestaltet werden	Calculator accepts text input when values are entered into the solution boxes, resetting the pendulum could be made simpler
B	2	Grüne Farbe des Feedbacks - SCHWER LESBAR	Green color of the feedback is hard to read
B	3	Messung schwer genau zu stoppen - ist frustrierend	It is hard to do an accurate measurement - frustrating
B	4	Das experiment war schwer, ich musste oft probieren	the experiment was hard to do, I had to try very often

Table 5.7: Negative feedback given at the free text answers at the end of the survey. The text has been translated from German to English by the author.

## 5 Evaluation

Group	#	Original Text	Translated
A	1	Detail[lierte Einführung für Menschen, die wenig am PC arbeiten (Mausklick rechts, links). Im Vorhinein Aufklärung über Dinge wie "auf wie viele Stellen muss gerundet werden" und "Punkt oder Komma als Dezimalzahl".	Detailed introduction for people who do not work on the PC very often (mouse click right, left). Awareness of things like "How many places must be rounded" and "Point or comma as decimal" in advance.
A	2	Bei den Feldern auf dem Whiteboard sollten die Beschriftungen zu den Werten stehen bleiben. Dadurch wäre die schnelle Eingabe der Werte einfacher.	For the fields on the whiteboard, the labels should stay despite entered values. This would make it easier to enter the values quickly.
A	3	-	-
A	4	-	-
B	1	Usability der Anwendung könnte verbessert werden, einige UI Elemente wie zum Beispiel der Taschenrechner sind in der Bedienung umständlich	Usability of the application could be improved, some UI elements such as the calculator are awkward to use
B	2	andere Farbe vom Feedbacks, Hinweise zu Beginn länger aufblenden	different colors for feedback, show hints at the beginning longer
B	3	genaue Erklärung ab 3 Fehlversuchen was eine Periode ist mit Videoanleitung oder so, Erklärung warum das nur unter 5 Grad geht	Exact explanation after 3 failed attempts what a period is, including a video tutorial or so, explanation why that only works under 5 degrees
B	4	Mehr hilfe, einmal vorzeigen half mir sehr	more help, showing off once how to do it helped a lot

Table 5.8: Suggestions given at the free text answers at the end of the survey. The text has been translated from German to English by the author.

## 6 Lessons Learned

This chapter summarizes all the knowledge gained by the different phases of the project work.

### 6.1 Research

The amount of papers of virtual learning environments increased tremendously since the 1990s. This suggests that the trend seems to be moving towards online learning courses. With that said, 3D simulated VLEs gain more and more attention too, explicitly since the last years due to affordable powerful computers. Especially the usage of non educational games.

Automatic assessment systems and intelligent tutoring systems are thoroughly researched at the moment. This can be easily observed when counting the amount of papers published in this field of research in the last decade too. It seems to be a similar situation for virtual learning environments. Nevertheless, from the point of view created by the research for this work, external assessment systems and especially assessment systems for 3D virtual environments are rather rare.

Basically, 4 techniques are used to implement assessment systems, rule and example based as internal or external application. Rule based systems are more versatile but harder to maintain in contrast to example based approaches. External systems provide a greater possibility for collaboration assessment then internal systems do. This is because they can run on an external machine and thus supervise all clients.

### 6.2 Development

Unity is a very versatile tool to create physics based applications. It provides all the tools needed to create the pendulum experiment. Basically, no line of code was written to define a pendulum. As the foundation is C#, it was very easy to

import DLL files and thus providing a predefined communication interface to the assessment system.

When implementing the assignment sheet in Unity, it showed, that the UI elements provided by Unity are rather limited and not a very good alternative to HTML pages for example. This especially showed up, when grouping them into rows and columns. The attached components, responsible for the correct layout, were not working and needed manual adjustment by code.

The very well designed structure of code of the SOFIA Evaluation Engine made it particularly easy to extend it with the requirements of the proposed work. This has once more shown that thoroughly planned code can reduce maintenance and extension costs drastically. Following exact design patterns of coding reduces the code tremendously and thus increases readability. Furthermore, creating a class diagram helped a lot as well.

### 6.3 Evaluation

During evaluation it was easily observable that simple text based assessment systems are not the optimal solution. Users tend to not read the text thoroughly enough or even skip it completely. Moreover, students tend to do the same mistake over and over again, which lead to a repeating same error messages and thus increasing frustration. An even more adaptive assessment model might have helped preventing that.

All in all the system was rated as easy-to-use. With that said, during the execution of the assignment the frustration increased because the model seemed to not adapt enough to provide more information. Especially when the same mistake was made over and over again, the system wasn't able to react to this situation enough. Furthermore, for some students, it did not provide enough background information to explain, why some measurements were not a period for example.

Further improvements might be done by providing the information paper inside the environment for example. Every time, a mistake happens during the experiment, the corresponding passage on the paper may be highlighted. Another possibility would be to provide an animated graph, which shows exactly when on the period the stopwatch has been pressed.

# 7 Future Work and Summary

## 7.1 Summary

3D immersive learning environment are a great possibility to support STEM education, especially physics experiments. The possibility to augment the virtual experiment with additional information provides great possibilities to aid students during learning. Moreover, the accessibility to 3D rendering capable computers is higher and easier for students than ever before, which makes the approach of virtualized physics laboratories very interesting.

Nevertheless, to be a really useful tool, it is not enough to just replicate the physics experiment. To provide a full featured lecture enhancement, the student needs to be automatically assessed. The biggest challenge though is to provide an assessment system, which is agile enough to work with any virtual learning environment and also adapts to skill of the student.

This work showed off the state of the art of virtual learning environments with focus on immersive 3D environments, as also of assessment systems. Resulting from this, a conceptual architecture was introduced and a design extended from a pre existing one by Maderer (2013). Accordingly, the proof-of-concept design utilized the existing prototype of Maderer, Gütl, and AL-Smadi (2013) and the new immersive 3D environment Maroon (Pirker, Lesjak, & Guetl, 2017) to show off the flexibility and agility of the prototype. Furthermore, it was extended with the described assignment feature and new small features.

The conducted survey showed promising results and an overall positive feedback. The beautiful background and the feedback itself earned very positive feedback. This fact is underlined by the results of the SUS and CES questionnaires. The main problems emerged have been a too little adaptive model according to the skills and knowledge of the student. Furthermore, some students complaint about too little information during the experiment if something went wrong.

## 7.2 Future Work

As already mentioned multiple times, the great versatility of example based assessment systems comes with the disadvantage of higher time consumption of the experiment expert. This fact can definitely be tackled by introducing semantic classes. With these, a pendulum for example just needs to be defined as a harmonic oscillating object, not as a set of rules. This reduces the amount of rules drastically.

Another possibility would be to provide a graphical editor, instead of writing the XML by hand. With techniques like copy and paste, templates or save and load the amount of work could be reduced as well. Especially copy paste and templates could help a lot. Additionally, adding operations to change variables in bulk would be a great enhancement too.

On the learning environment side, more support for the student could be implemented. Different sounds for different feedback would strongly increase the awareness of alternating feedback text. Furthermore, as already mentioned, the info sheet presents a great opportunity to be represented inside the environment and corresponding text passages are highlighted, if an error occurs. Another thought was to introduce a graph, which draws the period of the pendulum and represents the points, when the student presses the stopwatch.

Thinking even further, a problem could arise when handling clients in the double-digit range. As the amount of events rises, the system may not keep up to provide real time information for every student and thus starting to lag behind. To compensate this problem multiple possibilities could be considered: Use multiple servers, optimize the experiments to throw as little events as possible, execute minor assessment steps at the client and just send results, to name just a few. To find an optimal solution, a thorough study might be needed.

# Bibliography

- Albahari, J., & Albahari, B. (2017). *C# 7.0 in a Nutshell: The Definitive Reference*. Google Books-ID: HrE5DwAAQBAJ. "O'Reilly Media, Inc."
- Aleven, V., McLaren, B. M., & Sewall, J. (2009). Scaling Up Programming by Demonstration for Intelligent Tutoring Systems Development: An Open-Access Web Site for Middle School Mathematics Learning. *IEEE Transactions on Learning Technologies*, 2(2), 64–78. doi:10.1109/TLT.2009.22
- Aleven, V., McLaren, B. M., Sewall, J., & Koedinger, K. R. (2009). A New Paradigm for Intelligent Tutoring Systems: Example-Tracing Tutors. *Int. J. Artif. Intell. Ed.* 19(2), 105–154. Retrieved from <http://dl.acm.org/citation.cfm?id=1734243.1734245>
- Azuma, R. T. (1997). A Survey of Augmented Reality. *Presence: Teleoperators & Virtual Environments*, 6(4), 355–385.
- Baker, S. C., Wentz, R. K., & Woods, M. M. (2009). Using Virtual Worlds in Education: Second Life® as an Educational Tool. *Teaching of Psychology*, 36(1), 59–64. doi:10.1080/00986280802529079
- Berkeley, E. C., McGovern, P. J., Berlin, M. M., & MacDonald, N. D. (Eds.). (1962). *Computers and Automation* (2nd ed.).
- Brooke, J. (1996). SUS - A quick and dirty usability scale, 8.
- Bucanek, J. (2009). Model-view-controller pattern. *Learn Objective-C for Java Developers*, 353–402.
- Collis, B., De Boer, W., & Slotman, K. (2001). Feedback for web-based assignments. *Journal of Computer Assisted Learning*, 17(3), 306–313. doi:10.1046/j.0266-4909.2001.00185.x
- Corter, J. E., Nickerson, J. V., Esche, S. K., & Chassapis, C. (2004). Remote versus hands-on labs: A comparative study. (F1G–17). doi:10.1109/FIE.2004.1408586
- Dede, C., Eisenkraft, A., Frumin, K., & Hartley, A. (2016). *Teacher learning in the digital age: Online professional development in STEM education*. Harvard Education Press.
- Dempsey, P. (2016). The teardown: HTC Vive VR headset. *Engineering & Technology*, 11(7-8), 80–81.
- DiSessa, A. A. (1982). Unlearning Aristotelian Physics: A Study of Knowledge-Based Learning\*. *Cognitive Science*, 6(1), 37–75. doi:10.1207/s15516709cog0601\_2

## Bibliography

- DJSCOE, Vile - Parle (W), Mumbai, Rajesh Desai, P., Nikhil Desai, P., Deepak Ajmera, K., & Mehta, K. (2014). A Review Paper on Oculus Rift-A Virtual Reality Headset. *International Journal of Engineering Trends and Technology*, 13(4), 175–179. doi:10.14445/22315381/IJETT-V13P237
- Dougiamas, M., & Taylor, P. (2003). Moodle: Using learning communities to create an open source course management system. (pp. 171–178). Association for the Advancement of Computing in Education (AACE).
- Dunwell, I., de Freitas, S., & Jarvis, S. (2011). Four-dimensional consideration of feedback in serious games. (pp. 42–62). Continuum Publishing. Retrieved from <https://espace.curtin.edu.au/handle/20.500.11937/7496>
- Felder, R. M., & Brent, R. (2003). Designing and Teaching Courses to Satisfy the ABET Engineering Criteria. *Journal of Engineering Education*, 92(1), 7–25. doi:10.1002/j.2168-9830.2003.tb00734.x
- Gamma, E. (1995). *Design patterns: Elements of reusable object-oriented software*. Pearson Education India.
- Goldstone, W. (2009). *Unity Game Development Essentials*. Google-Books-ID: WfAWzVW9IKoC. Packt Publishing Ltd.
- Greuel, C., Myers, K., Denker, G., & Gervasio, M. (2016). Assessment and content authoring in semantic virtual environments.
- Gustavsson, I., Nilsson, K., & Lagö, T. (2009). On Physical Experiments and Individual Assessment of Laboratory Work in Engineering Education. (84:506–84:509). MEDES '09. doi:10.1145/1643823.1643926
- Haas, J. K. (2014). A History of the Unity Game Engine, 45.
- Hartzler, D. S. (2000). *A meta-analysis of studies conducted on integrated curriculum programs and their effects on student achievement* (Doctoral dissertation).
- Heffernan, N. T., & Heffernan, C. L. (2014). The ASSISTments Ecosystem: Building a Platform that Brings Scientists and Teachers Together for Minimally Invasive Research on Human Learning and Teaching. *International Journal of Artificial Intelligence in Education*, 24(4), 470–497. doi:10.1007/s40593-014-0024-x
- Jaques, P. A., Seffrin, H., Rubi, G., de Moraes, F., Ghilardi, C., Bittencourt, I. I., & Isotani, S. (2013). Rule-based expert systems to support step-by-step guidance in algebraic problem solving: The case of the tutor PAT2math. *Expert Systems with Applications*, 40(14), 5456–5465. doi:10.1016/j.eswa.2013.04.004
- Jing, Y., Behzad, M., Jeff H, P., Sarah Farukhi, A., & S Khizer, K. (2018). The Visual Effects Associated with Head-Mounted Displays. *International Journal of Ophthalmology and Clinical Research*, 5(2). doi:10.23937/2378-346X/1410085
- Johnson-Eilola, J. (2005). Datacloud: Toward a new theory of online work. *New York*, 71–74.



## Bibliography

- Kaplan, J., & Yankelovich, N. [N.]. (2011). Open Wonderland: An Extensible Virtual World Architecture. *IEEE Internet Computing*, 15(5), 38–45. doi:10.1109/MIC.2011.76
- Kaufmann, H., & Meyer, B. (2008). Simulating educational physical experiments in augmented reality. (p. 1). doi:10.1145/1507713.1507717
- Kay, R. H., & Loverock, S. (2008). Assessing emotions related to learning new software: The computer emotion scale. *Computers in Human Behavior*, 24(4), 1605–1623. doi:10.1016/j.chb.2007.06.002
- King, B. (1998). Establishing UniSA-net: The On-line Environment of the University of South Australia. *Information Technology Advisory Committee, University of South Australia*.
- Klemm, W. R., & Snell, J. R. (1996). Enriching computer-mediated group learning by coupling constructivism with collaborative learning. *Journal of Instructional Science and Technology*, 1(2), 1–11.
- Kohavi, R., & Longbotham, R. (2017). Online Controlled Experiments and A/B Testing. In C. Sammut & G. I. Webb (Eds.), (pp. 922–929). doi:10.1007/978-1-4899-7687-1\_891
- Kress, B. C., & Cummings, W. J. (2017). Optical architecture of HoloLens mixed reality headset. *Proc.SPIE*, 10335, 10. doi:10.1117/12.2270017
- Littlejohn, A., & Pegler, C. (2007). *Preparing for blended e-learning*. connecting with e-learning series. Routledge.
- MacIsaac, D. (2015). Google Cardboard: A virtual reality headset for \$10? *The Physics Teacher*, 53(2), 125–125. doi:10.1119/1.4905824
- Maderer, J. (2013). *Flexible Assessment in Immersive Environments*. Graz University of Technology.
- Maderer, J., & Gütl, C. (2013). Flexible Automated Assessment in 3d Learning Environments: Technical Improvements and Expert Feedback. (pp. x–x). .
- Maderer, J., Gütl, C., & AL-Smadi, M. (2013). Formative Assessment in Immersive Environments: A Semantic Approach to Automated Evaluation of User Behavior in Open Wonderland, 13. Retrieved from [https://www.researchgate.net/profile/Mohammad\\_AL-Smadi/publication/254258453-Formative-Assessment\\_in\\_Immersive\\_Environments\\_A\\_Semantic\\_Approach\\_to\\_Automated\\_Evaluation\\_of\\_User\\_Behavior\\_in\\_Open\\_Wonderland/links/oc96051fc5b7c033f0000000.pdf](https://www.researchgate.net/profile/Mohammad_AL-Smadi/publication/254258453-Formative-Assessment_in_Immersive_Environments_A_Semantic_Approach_to_Automated_Evaluation_of_User_Behavior_in_Open_Wonderland/links/oc96051fc5b7c033f0000000.pdf)
- Martin, J., & Banks, R. (2014). *Visual Studio 2013 Cookbook*. Packt Publishing Ltd.
- Meijer, E., Beckman, B., & Bierman, G. (2006). LINQ: Reconciling Object, Relations and XML in the .NET Framework. (pp. 706–706). SIGMOD '06. doi:10.1145/1142473.1142552
- Mitrovic, A. (1997). SQL-Tutor: A preliminary report.

## Bibliography

- Mitrovic, A., & Ohlsson, S. (1999). Evaluation of a Constraint-Based Tutor for a Database Language, 19.
- Moro, C., Stromberga, Z., & Stirling, A. (2017). Virtualisation devices for student learning: Comparison between desktop-based (Oculus Rift) and mobile-based (Gear VR) virtual reality in medical and health science education. *Australasian Journal of Educational Technology*, 33(6).
- Nebel, S., Schneider, S., & Rey, G. D. (2016). Mining learning and crafting scientific experiments: A literature review on the use of minecraft in education and research. *Journal of Educational Technology & Society*, 19(2), 13.
- Ohlsson, S. (1994). Constraint-based student modeling. (pp. 167–189). Springer.
- Pirker, J. (2017). *Immersive and Engaging Forms of Virtual Learning* (Doctoral dissertation).
- Pirker, J., Lesjak, I., & Guetl, C. (2017). Maroon VR: A Room-Scale Physics Laboratory Experience. (pp. 482–484). IEEE.
- Pirker, J., Lesjak, I., & Gütl, C. (2017). An Educational Physics Laboratory in Mobile Versus Room Scale Virtual Reality – A Comparative Study. *International Journal of Online Engineering (iJOE)*, 13(08), 106. doi:10.3991/ijoe.v13i08.7371
- Polya, G. (Ed.). (2003). *Teaching mathematics through problem solving: Prekindergarten-grade 6*. Reston, VA: National Council of Teachers of Mathematics.
- Potkonjak, V., Gardner, M., Callaghan, V., Mattila, P., Guetl, C., Petrović, V. M., & Jovanović, K. (2016). Virtual laboratories for education in science, technology, and engineering: A review. *Computers & Education*, 95, 309–327. doi:10.1016/j.compedu.2016.02.002
- Sanders, M. (2009). STEM, STEM Education, STEMmania, 8.
- Scanlon, E., Morris, E., Paolo, T. d., & Cooper, M. (2002). Contemporary approaches to learning science: Technologically-mediated practical work. *Studies in Science Education*, 38(1), 73–114. doi:10.1080/03057260208560188
- Scerbakov, N. (2015). *TU Graz Teach-Center*. Available.
- Settgast, V., Pirker, J., Lontschar, S., Maggale, S., & Gütl, C. (2016). Evaluating experiences in different virtual reality setups. (pp. 115–125). Springer.
- Short, D. (2012). Teaching scientific concepts using a virtual world - Minecraft. 58(3), 5.
- Sicker, D. C., Lookabaugh, T., Santos, J., & Barnes, F. (2005). Assessing the Effectiveness of Remote Networking Laboratories. (S3F–S3F). doi:10.1109/FIE.2005.1612279
- Al-Smadi, M., Wesiak, G., & Guetl, C. (2012). Assessment in serious games: An enhanced approach for integrated assessment forms and feedback to support guided learning. (pp. 1–6). IEEE.
- Stiles, M. J. (2000). *Effective Learning and the Virtual Learning Environment*.

## Bibliography

- Unity. (n.d.). Unity. Used Version: 2018.1.6f1. 30 3rd Street; San Francisco, , CA 94103; United States. Retrieved September 20, 2018, from <https://unity3d.com>
- Vanlehn, K. (2006). The behavior of tutoring systems. *International journal of artificial intelligence in education, 16*(3), 227–265.
- Warburton, S. (2009). Second Life in higher education: Assessing the potential for and the barriers to deploying virtual worlds in learning and teaching. *British Journal of Educational Technology, 40*(3), 414–426. doi:10.1111/j.1467-8535.2009.00952.x
- Wendt, P. (1956). An evaluation of the effectiveness of instruction and audience reaction to programming an educational television station. *Audiovisual communication review, 4*(1), 79–79. doi:10.1007/BF02718467
- Wieman, C. E., Adams, W. K., & Perkins, K. K. (2008). PhET: Simulations That Enhance Learning. *Science, 322*(5902), 682–683. doi:10.1126/science.1161948
- Wigley, A., Sutton, M., Wheelwright, S., Burbidge, R., & Mcloud, R. (2002). *Microsoft .NET compact framework: Core reference*. Microsoft Press.
- Yamagata-Lynch, L. C. (2014). Blending online asynchronous and synchronous learning. *The International Review of Research in Open and Distributed Learning, 15*(2).
- Yankelovich, N. [Nicole], Walker, W., Roberts, P., Wessler, M., Kaplan, J., & Provino, J. (2004). Meeting central: Making distributed meetings more effective. (p. 419). doi:10.1145/1031607.1031678
- Zhang, D., Zhao, J. L., Zhou, L., & Nunamaker, J. F., Jr. (2004). Can e-Learning Replace Classroom Learning? *Commun. ACM, 47*(5), 75–79. doi:10.1145/986213.986216

## Questionnaire English

While working on the Pendulum Experiment, I felt

	None of the time	Some of the time	Most of the time	All of the time
satisfied	0	0	0	0
disheartened	0	0	0	0
anxious	0	0	0	0
irritable	0	0	0	0
excited	0	0	0	0
dispirited	0	0	0	0
insecure	0	0	0	0
frustrated	0	0	0	0
curious	0	0	0	0
helpless	0	0	0	0
nervous	0	0	0	0
angry	0	0	0	0

## 7 Appendix

	<b>strongly disagree</b>	<b>disagree</b>	<b>undecided</b>	<b>agree</b>	<b>strongly agree</b>
I think that I would like to use this system frequently.	o	o	o	o	o
I found the system unnecessarily complex.	o	o	o	o	o
I thought the system was easy to use.	o	o	o	o	o
I think that I would need the support of a technical person to be able to use this system.	o	o	o	o	o
I found the various functions in this system were well integrated.	o	o	o	o	o
I thought there was too much inconsistency in this system.	o	o	o	o	o
I would imagine that most people would learn to use this system very quickly.	o	o	o	o	o
I found the system very cumbersome to use.	o	o	o	o	o
I felt very confident using the system.	o	o	o	o	o
I needed to learn a lot of things before I could get going with this system.	o	o	o	o	o

## 7 Appendix

	<b>strongly disagree</b>	<b>disagree</b>	<b>undecided</b>	<b>agree</b>	<b>strongly agree</b>
The feedback has been very helpful.	0	0	0	0	0
The feedback was obstructing.	0	0	0	0	0
The Content of the feedback was very detailed.	0	0	0	0	0
I think, a final feedback would be better than an immediate feedback.	0	0	0	0	0

## Questionnaire German

Während ich am Pendelexperiment arbeitete, fühlte ich mich

	<b>nie</b>	<b>manchmal</b>	<b>oft</b>	<b>immerzu</b>
zufrieden	0	0	0	0
entmutigt	0	0	0	0
ängstlich	0	0	0	0
reizbar	0	0	0	0
aufgeregt	0	0	0	0
niedergeschlagen	0	0	0	0
unsicher	0	0	0	0
frustriert	0	0	0	0
neugierig	0	0	0	0
hilfslos	0	0	0	0
nervös	0	0	0	0
verärgert	0	0	0	0

## 7 Appendix

	<b>stimme überhaupt nicht zu</b>	<b>stimme eher nicht zu</b>	<b>weder noch</b>	<b>stimme eher zu</b>	<b>stimme voll zu</b>
Ich denke, ich würde das System regelmäßig verwenden.	0	0	0	0	0
Ich finde, das System war unnötig komplex.	0	0	0	0	0
Ich denke, das System war einfach zu verwenden.	0	0	0	0	0
Ich glaube, ich bräuchte hilfe von einer technisch versierten Person, um dieses System verwenden zu können.	0	0	0	0	0
Die einzelnen Funktionen des Systems sind sehr gut integriert.	0	0	0	0	0
Meine Meinung nach war das System in sich widersprüchlich.	0	0	0	0	0
Ich kann mir vorstellen, dass die meisten Leute sehr schnell lernen werden, dieses System zu verwenden.	0	0	0	0	0
Ich finde, das System war sehr unhandlich.	0	0	0	0	0
Ich fühlte mich sehr sicher beim Umgang mit dem System.	0	0	0	0	0
Ich musste sehr viel lernen, bevor ich das System verwenden konnte.	0	0	0	0	0



## 7 Appendix

	stimme überhaupt nicht zu	stimme eher nicht zu	weder noch	stimme eher zu	stimme voll zu
Die Hilfestellungen des Systems waren sehr hilfreich.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ich empfand die Hilfestellungen störend.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Die Hilfestellungen waren ausreichend detailliert.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Meiner Meinung nach wäre Feedback am Ende des Experiments hilfreicher gewesen, anstelle ständiger Hilfestellungen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

## 7 Appendix

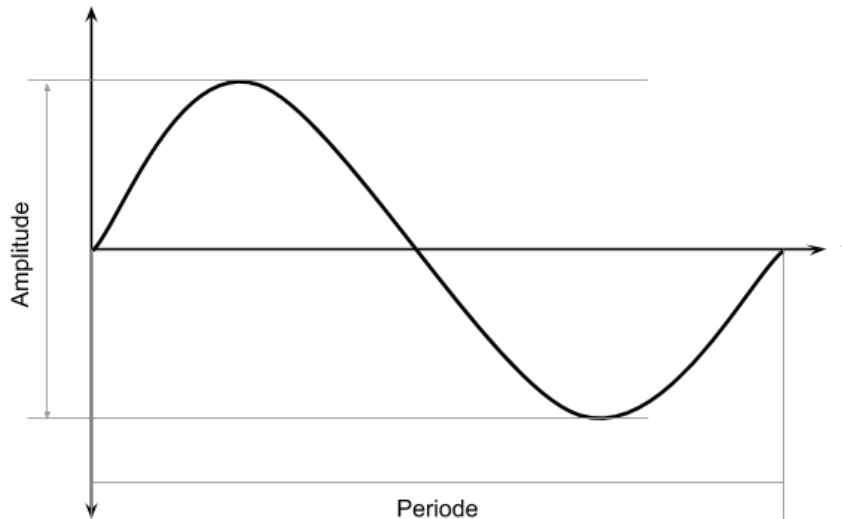
# Infosheet

## Pendel

Ein Pendel ist ein schwingender Körper, dessen einziger Einfluss bezüglich Kräfteinwirkung die Gravitation ist. Des weiteren wird es an der freien Fallbewegung durch ein Seil oder eine Stange gehindert, weshalb sich diese links-rechts-Bewegung ergibt.

## Periode

Ein vollständiger Durchgang eines Pendels wird **Periode** genannt. Diese Grafik zeigt eine schematische Darstellung einer Periode:



Diese Grafik kann beispielsweise durch einen Seismographen (Gerät um Erdbeben aufzuzeichnen) erstellt werden: Das Papier wird von links nach rechts (Pfeilrichtung t, t für Zeit) unter einem Bleistift entlang gezogen. Wenn der Stift hin und her schwingt, zeichnet dieser die obige Grafik.

Die **Amplitude** gibt an, wie stark das Pendel maximal ausgelenkt wird. Je stärker die maximale Auslenkung, desto höher die Amplitude.

## Frequenz

Die **Frequenz** eines Pendels steht im direkten Zusammenhang mit der Periode. Die Formel um die Frequenz zu berechnen, lautet

$$f = \frac{1}{t}$$

wobei  $f$  die Frequenz ist und  $t$  die **Zeit einer Periode** (= wie viele Perioden werden pro Sekunde durchlaufen).. Die Form, die in der Abbildung zu sehen ist, hat aber im Fall eines Pendels nur Gültigkeit, solange das Pendel nicht stärker als  $5^\circ$  ausgelenkt wird.