communicated by:
Lehr- und Forschungsgebiet Informatik 9

Prof Dr. Ulrik Schroeder

The present work was submitted to Learning Technologies Research Group
Diese Arbeit wurde vorgelegt am Lehr- und Forschungsgebiet Informatik 9

Learning Analytics in Serious Games with xAPI
Learning Analytics in Serious Games mit xAPI

Master-Thesis
Masterarbeit

presented by / von

Schulz, Jan-Patrick
354467

Prof. Dr.-Ing. Ulrik Schroeder

Prof. Dr. Jan Borchers

Aachen, May 28, 2021

Contents

Abstract

One way to enhance the learning experience is to use Serious Games. In Serious Games it is possible to collect learning data of the players. The learning data can be analyzed to evaluate the achievement of learning goals of the players and to enhance the Serious Game. To show how Learning Analytics can be added to a Serious Game, the Collaborative Multitouch Serious Game *transaction MTLG* is upgraded in this master thesis. Learning goals are defined to evaluate the performance of the players. On basis of these learning goals a dashboard is created.
transaction MTLG is enhanced with online multiplayer to allow collecting learning data with xAPI during the COVID-19 pandemic and bugs of the game are fixed. The used xAPI definitions are taken from the vocabulary of the Learning Technologies Research Group. In a user study learning data are collected and analyzed with the graphics of the dashboard. This analysis shows that it is possible to gain meaningful insights on the player performance in the Serious Game *transaction MTLG*. These insights can be used to enhance the learning experience of the players. The achievement of the learning goals is also evaluated with the dashboard.
A dashboard usability study shows that the created dashboard has a high usability and can be used to analyze and display the data of the learners. Thus the goal of adding Learning Analytics to *transaction MTLG* is reached.

Part I

Introduction

Chapter 1 Motivation

Day to day learning

In the day to day life one keeps learning every day, a new path through town, a new word to use, or using a new application on a smartphone. This learning is happening on the fly and is not perceived as learning in a classical sense, as it is not in school. But nonetheless, it results in new strategies to master new challenges. Without learning new things life would become repetitive and dull, but still for a lot of learners it seems boring to learn in a classroom setting. One way to make the learning tasks more appealing is the usage of Serious Games.

Learning with Serious Games

With Serious Games the learning material is embedded in a gaming concept and the motivational aspects from games can be used to make the learning material more interesting. Thus it becomes easier for the students to learn with the help of a Serious Game.

Goal-oriented learning

Whether one learns in day to day life, a classroom setting, or with a Serious Game, one tries to achieve some goal. Either find a faster path to get to our favorite place in town, a way to articulate more sophisticated, or increase our smartphones functionality with a new application. In a classroom setting a course is attended to gain knowledge about the subject. As well as a Serious Game, which promises to teach about a topic. Whether the learning in these examples was successful can be measured by verifying the achievement of the preset goals. When learning in day to day life there is no explicit test. It is important for our satisfaction whether the goal was reached or not, because there is no need to compare the performance between different persons. In a traditional way the learning success is evaluated via tests, which allows a direct comparison between all, who attended the course. But it is hardly possible to monitor the performance of all students in the course throughout the whole learning process. When using a digital Serious Game all actions of the user in the Serious Game can be stored and then analyzed. Thus it becomes possible to monitor the learning progress throughout the whole learning experience. This can be used for example to give the students more in-depth feedback or to enhance the learning game further.

Need for analytics

The information acquired during a digital Serious Game needs to be analyzed. Just looking at the raw data will not yield much knowledge gain in acceptable time. Furthermore, the amount of collected data can be overwhelming if for example each mouse movement in the game is tracked. Thus it is important to have an analytics engine to perform analysis of the data that is collected in the game. Furthermore, Learning Analytics are interesting for different stakeholder like students or teachers and specialized Learning Analytic types exist for these stakeholders.

xAPI as format

A lot of different data formats exist to store data. For storing learning data this master thesis uses xAPI. xAPI is used commonly for storing learning data and it gives flexibility while still allowing interoperability. In addition, data stored with xAPI is human readable, which allows a better communication of the data between different stakeholders.

Presentation of analysis

The result of the analysis needs to be displayed in an understandable way that does not result in wrong deductions. If the result is not displayed clearly enough and the viewer does not understand what is shown then the analysis can result in actions that are even harmful to the learning progress of the student.

Plan of this thesis

This master thesis will construct a dashboard that will display the results of Learning Analytics of a Serious Game. This dashboard will then be evaluated with a pilot dashboard usability study. The Serious Game used to collect the learning data is the Collaborative Multitouch Serious Game *transaction MTLG* by Schablowsky [1], which was programmed during his master thesis. The learning data of the players will be collected via xAPI.

Task of this thesis

The goal of this work is to enable Learning Analytics for the Collaborative Multitouch Serious Game *transaction MTLG*. For this Learning data are collected from the game with xAPI. This data are used to evaluate the learning goals of *transaction MTLG*. To evaluate the learning goals the data must be collected from players and cannot be generated synthetically. Since *transaction MTLG* is a collaborative game that is played on a single device and since the current COVID-19 situation does not allow for meetings in person, *transaction MTLG* needs to be upgraded to allow online multiplayer. Then a study is conducted to collect the learning data and get a self-assessment of the players. Out of the learning data a dashboard presenting visualizations to evaluate the learning goals is created. As last step this dashboard is evaluated with a pilot usability study.

Overview of content

In the next chapter the related work of the basics of this thesis will be presented. These basics are Serious Games, Learning Analytics, the usage of xAPI, and Learning Locker. Then the previous implementations of the game are explained.

Learning Analytics with xAPI for transaction MTLG

The second part of this master thesis is Learning Analytics with xAPI for *transaction MTLG*. The first chapter of this part is the conceptual design of the Learning Analytics with xAPI. This chapter starts with the learning goals followed by the xAPI usage and the last section is the data flow. The next chapter is the conceptual design of the dashboard. At first paradigms of dashboard creation that are used in this master thesis are described. Then all used graphic types are shown, followed by a description of the dashboard and its graphics and how these graphics can be used to answer whether the players reached the learning goals. The next chapter will show the implementation. At first the upgrade of *transaction MTLG* to allow the collection of learning data is described. This includes enabling the game to be played via different browsers to comply with the current contact restrictions, as the game is designed to be a collaborative multiplayer game. It will also be explained how the configuration of the game was fine-tuned. Then the main elements of the implementation of the dashboard will be shown.

Evaluation

The third part of the master thesis is the evaluation of the collected data. At first the user study to collect learning data from the game is described. To verify whether the learning goals were reached by the participants a questionnaire is done. The following section deals with the analysis of the learning data via the dashboard. The next chapter describes the dashboard usability study.

Result

The fourth part of the master thesis is the result. The conclusion is drawn and shown that the goals of the master thesis are achieved. At last, recommendations on future work are made.

Chapter 2 Related Work

Chapter overview

In this chapter at first an overview of Serious Games is given and how Serious Games can be categorized. Next, Learning Analytics is described with different types of Learning Analytics. At last xAPI is described and how the xAPI statements are created and what the elements of the statements mean.

2.1. Serious Games

Definition of Serious Games

In the literature no generally accepted definition of Serious Games exists. The definitions of Serious Games differ in the details. In this chapter two definitions of Serious Games are presented and discussed. The following definition of Serious Games is by Loh et al. [2]:

> "Serious Games are digital games and simulation tools that are created for non-entertainment use, but with the primary purpose to improve skills and performance of play-learners through training and instruction." [2]

Serious Games are digital games or simulation tools. Additionally, they need to be created without a focus on entertainment but with a focus on the improvement of skills and performance of the players. The term play-learners refers to persons, who learn by playing. A Serious Game uses training and instruction as a mean to teach.

Categorization of games

Loh et al. [2] show in figure 2.1. that entertainment games have a high enjoyment value, while their human performance improvement remains lower as the enjoyment value. Serious Games are the exact opposite with higher human performance improvement as enjoyment value. Whether a video game is classified by Loh et al. [2] as Serious Game or entertainment game depends on which of the two values enjoyment or human performance improvement is higher. The circle in the bottom left containing games with both low enjoyment and human performance improvement contains message broadcasters and edutainment games. Message broadcasters are in the bottom left, because the main focus of a message broadcaster is to transmit a specific message instead of focusing on the actual human performance improvement.

Categorization of Serious Games

Alvarez et al. [3] present a way to categorize Serious Games by gameplay, purpose and sector. The gameplay criteria can either be a video game or a video toy. A video game has a clear goal, which the player needs to reach in order to win the game. For example solve a specific task to advance. A video toy does not give the player any explicit goals to reach. For example exploring a world. The criteria purpose can either be broadcasting a message, provide training, or the sharing of data. The third criteria sector orders the game to a market. Possible markets are military, education, business, or advertisement, but it is not limited to these. The criteria sector also describes the target audience either as public, professionals, or students. A game that is a video toy is characterized by Alvarez

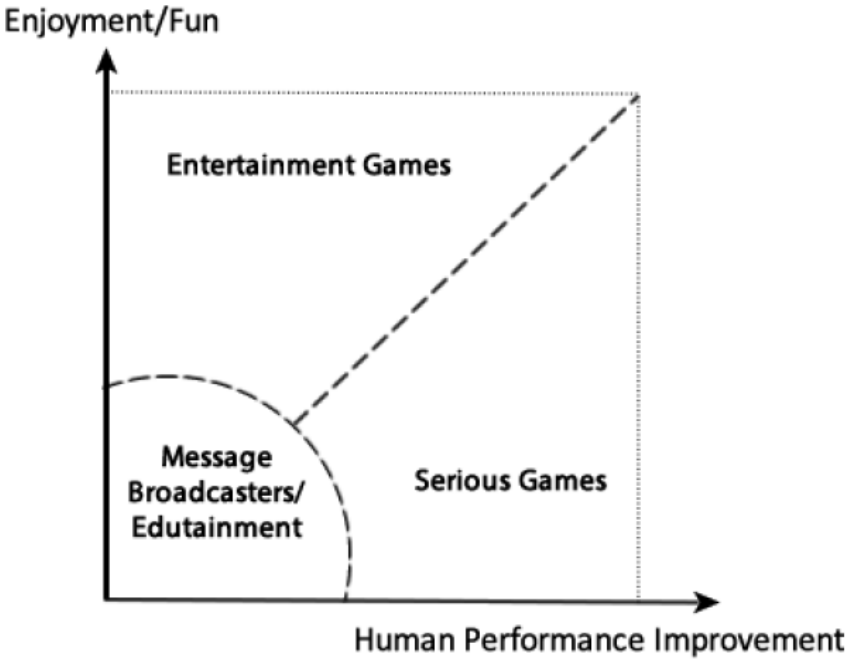


Figure 2.1.: Gametypes by Loh et al. [2]

et al. [3] as serious play instead of Serious Game. The definition of a Serious Game by Alvarez et al. [3] is that a Serious Game combines a video game with one or more utility functions, which are described by the purpose criteria, and it targets not only the entertainment market, but also another market from the market criteria.

Alternative definition

The definition by Alvarez et al. [3] has in difference to the first definition by Loh et al. [2] that not only games that have primary purpose to improve skill and performance can be seen as a Serious Game but a Serious Game also needs to have a clear goal for the player to achieve. The other definition by Loh et al. [2] says that a Serious Games primary purpose is to improve skills and performance. The first definition [2] differs from the second definition [3] of a Serious Game by removing message broadcasters from being a Serious Game and values the enjoyment factor of Serious Games higher. Furthermore it simplifies the purpose of the game to human performance improvement.

2.2. Learning Analytics

Section overview

This section will at first give a definition of Learning Analytics. Next, different approaches of Learning Analytics will be presented. Following that, objectives that Learning Analytics try to achieve will be defined. Then different dimensions of Learning Analytics are shown. After that a model on how to structure Learning Analytics is described. At last, a possible implementation of Learning Analytics in a Serious Game is shown.

Definition of Learning Analytics

Learning Analytics is trying to gain knowledge out of data that is generated during learning. The most commonly used definition for Learning Analytics was made by the 1st International Conference on Learning Analytics and Knowledge:

> "Learning Analytics is the measurement, collection, analysis and reporting of data about learners and their contexts, for purposes of understanding and optimizing learning and the environments in which it occurs" (as cited in Siemens et al. [4], 2011, Learning Analytics section, para. 2).

So in short: Learning Analytics is using data generated while learning to improve learning. Using Learning Analytics will result in an improvement cycle. Collecting data, improving the learning, collecting data, and so on. Through this the learning quality will improve increasingly over time.

Need for analysis

With the use of digital media for learning a lot of learning data is generated by the students. This data alone does not help to increase the quality of the teaching without proper analysis. The amount of data generated by Learning Management Systems (LMS) makes it unfeasible to try to analyze the data without the help of analysis tools.

Approaches to Learning Analytics

Due to the use of LMS a lot of learning data was captured, which helped the advancement of Learning Analytics [5]. Ifenthaler [5] lists different approaches of Learning Analytics:

- **Social Learning Analytics** deals with the analysis of data that is generated when learners interact with each other.
- **Teaching Analytics** gives the teachers the possibility to look into the learning processes and thus the ability to understand the learning process and intervene if it is necessary.
- **School Analytics** analyze all data from one school to support school wide decisions.
- **Curriculum Analytics and Learning Analytics Design** is used to evaluate the quality of different courses and study programs.
- **Measurement or Assessment Analytics** look at the learn process by the student and evaluate it with feedback to the student.
- **Multimodal Learning Analytics** use learning data from different media, such as wearables, GPS tracker, and voice recognition to get deeper insight in the learning process.

Learning Analytics in this thesis

In this master thesis Learning Analytics with the focus on Teaching Analytics will be used to help the teacher understand the learning process and intervene if it is necessary. As well as a focus on Measurement Analytics to display the performance of students and help the student to understand their performance.

Objectives of Learning Analytics

After describing different approaches of Learning Analytics the next section deals with the objectives of Learning Analytics. Verbert et al. [6] have researched the following objectives of Learning Analytics:

- Predicting learner performance and modeling learners
- Suggesting relevant learning resources
- Increasing reflection and awareness
- Enhancing social learning environments
- Detecting undesirable learner behavior
- Detecting affects of learners

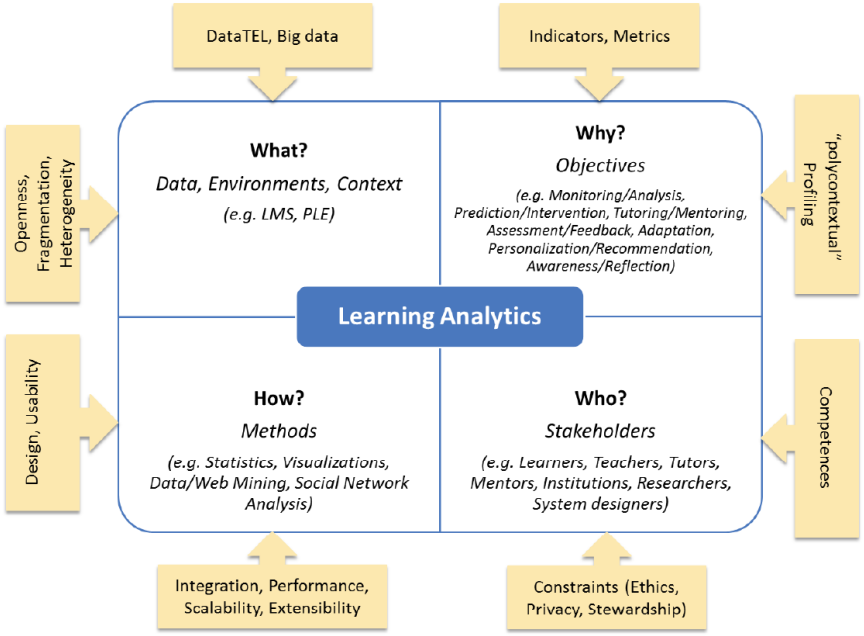


Figure 2.2.: Learning Analytics [7]

With these objectives Learning Analytics can positively affect the learning process. The Learning Analytics used in this master thesis will focus on enhancing social learning environments, because *transaction MTLG* is a collaborative multiplayer game. Furthermore, reflection and awareness can be increased through a presentation of the analysis.

Four dimensions of Learning Analytics

The paper by Chatti et al. [7] gives a broad view on Learning Analytics and describes its four dimensions. In figure 2.2 the dimensions What, Why, How, and Who of Learning Analytics are shown. The What dimension includes on which resource Learning Analytics is conducted. This could be data from a Learning Management System, a Personalized Learning Environment, or a Massively Open Online Course. This data is then analyzed to answer the objectives that are formulated in the Why dimension. Chatti et al. [7] list the following possible questions:

- Monitoring and analysis
- Prediction and intervention
- Tutoring and mentoring
- Assessment and feedback
- Adaption
- Personalization and recommendation
- Awareness and reflection

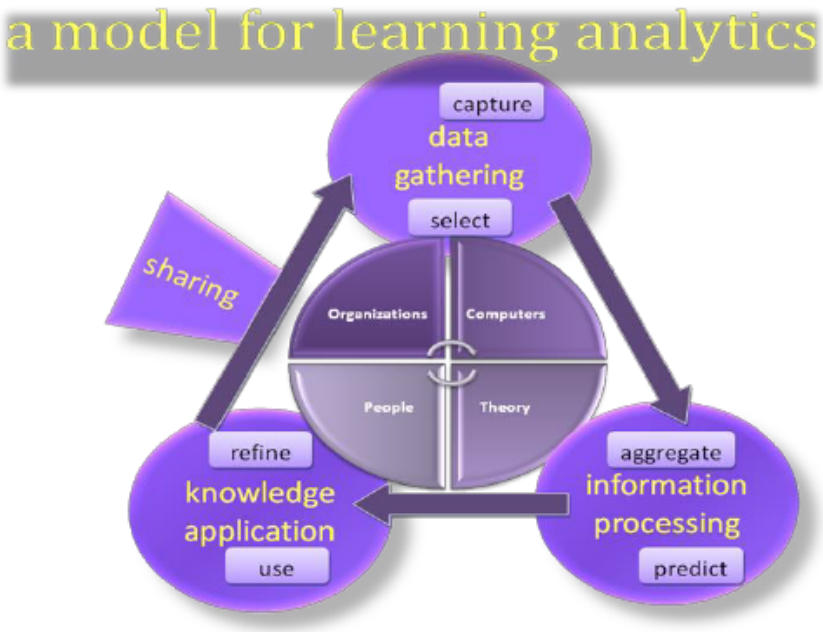


Figure 2.3.: Learning Analytics model by Elias [8]

The How section deals with the tools and methods that are used to get results for the objectives. The Who section deals with the stakeholders of implementations of Learning Analytics. Chatti et al. [7] say in regard to the stakeholders that it is important to involve both teachers and students in Learning Analytics to increase the user acceptance. On the border of the sections are additional keywords that influence the corresponding section.

Learning Analytics model

After describing the basics of Learning Analytics the next section shows a concept how Learning Analytics can be set up. Elias [8] presents a model for this purpose. The graphical representation of the model can be seen in figure 2.3.

Description of the model

In the step data gathering is determined which data to capture from the learners and which data to use for the analysis. In the information processing step data is aggregated and used to predict possible values. The third step knowledge application is using the findings from information processing to refine the learning environment for the learners. The results can be shared to help other projects with their Learning Analytics. The circle in the middle of figure 2.3. contains the four elements computers, theory, people, and organizations. These four are a substantial part for Learning Analytics. Computers are needed to be able to collect enough data. Computers also means other digital devices in this context. The theory is needed to deduct value from the ground laying data. People are needed to interpret the results and use them to refine the learning process. Organizations are needed to support and implement Learning Analytics as well as to share their findings to improve the overall quality of Learning Analytics. Furthermore, Elias [8] says that it is important that the time between data collection and refinement of the learning process should be reduced as far as possible so the results can help to improve the learning of the people the data was collected from. This would be an improvement to current cases where mostly changes to improve the learning are made after a course has concluded with no effect on the current course.

Practical approach to Learning Analytics

The next section gives a more practical view on how to implement Learning Analytics for Serious Games. Freire et al. [9] combine Learning Analytics and Game Analytics to

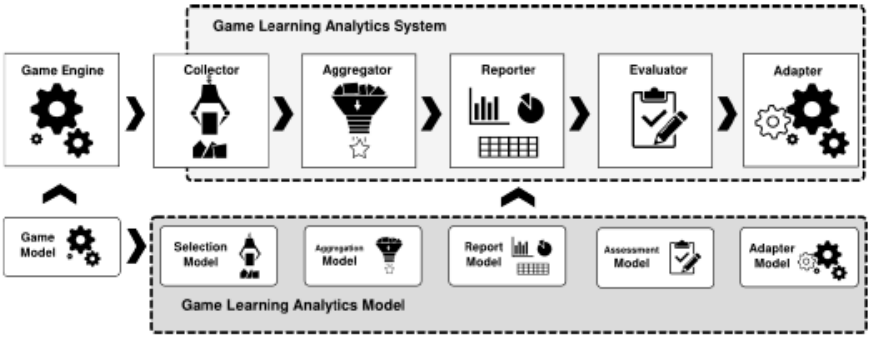


Figure 2.4.: Learning Analytics conceptual architecture by Freire et al. [9]

define the term Game Learning Analytics. Game Analytics are used by game developers to monitor what is happening inside their games. Since in a Serious Game the gameplay elements are essential to the learning progress of the student, Freire et al. [9] propose the merge of Game Analytics and Learning Analytics to Game Learning Analytics. A possible implementation of a system that uses Game Learning Analytics proposed by Freire et al. [9] is shown in a graphical representation in Figure 2.4.

Description of the approach

Figure 2.4. is divided into the system in the top row and the model in the row below. The data flow is from the left to the right. The game engine contains the basic game functionality. The collector takes the game events from players from the game engine and stores them in a format that the aggregator can access them. The aggregator aggregates the data. The reporter generates reports that are shown in the dashboard. Then an evaluator can check whether the reports are as intended or if changes are needed to adjust the game. If changes are needed they are done via the adapter to impact the game. The model below the system contains the model of each of the elements of the system. The system elements are configured according to their specification in the model. Furthermore, Freire et al. [9] emphasize that real-time analysis can be used to intervene in a game if it is striding of the intended path to get the game back on track.

Benefits of the approach

With this implementation Game Learning Analytics can also be used to predict student performance, improve the Serious Games, and personalize the game for each student [9]. The student performance can be predicted by finding students that have completed the same tasks in the same pattern and then looking at their test results, thus it is possible to categorize the students and predict their performance on the other tasks. A Serious Game can be improved by using the analysis to find parts of the game in which the students had problems with the task. This could be caused by a too great difficulty or an unclear definition of the current task, and can be found by low success rates and long answering times. When such a task is identified the Serious Game can be enhanced to make this task easier understandable and more solvable. If another task is too easy it can also be found with analytics. Then the difficulty can be adjusted as needed. Thus the user experience of the Serious Game can be enhanced. The game can be personalized for each student by evaluating, which types of tasks the student has a good performance on and which topics the student has a higher likelihood of answering wrong. Thus tasks can be chosen on the topic the student needs to improve on.

2.3. xAPI

"One of the most widely used information standards in the educational field is xAPI" by Alonso-Fernandez et al. [10]. This quote shows that xAPI has a high relevance in the educational field.

xAPI overview

Experience Application Programming Interface (xAPI) is a standard for learning data. The learning data can be collected with xAPI in different digital learning environments, such as Learning Management Systems, Serious Games, and more. Since xAPI is a standard the data in conformance with it can be accessed and analyzed by different systems. xAPI defines statements that can be stored in a Learning Record Store (LRS). A basic xAPI statement is constructed out of an actor, verb, and object as well as optional data like a timestamp. These statements are validated for their conformance, when they are sent to a LRS to keep a high data quality. [11]

Interpretation of specification

xAPI has specifications, but these are general and leave room for interpretation. This makes it possible to adapt the usage of xAPI for a specific context, but problematic is that there is not a single correct way to use it. Thus different institutions can have their own xAPI specifications for similar contexts. [12]

xAPI specification description

In the next section different parts of the xAPI specification [13] will be described that can be used for collecting the data of learning activities:

- **Actor** describes the initiator of a statement, which could be an agent or a group.
- **Verb** is used to define the interaction between the actor and the object. No verbs are defined by the xAPI specification except the verb void. Void is used to delete other statements if they are not correct. A void statement cannot be deleted. Each verbs needs an IRI to link it to its definition. Verbs should be used in the past tense.
- **Object** describes what the actor has interacted with. This can be either an agent, activity, statement reference, or sub-statement, but only one nesting level is allowed for sub-statements. No interaction activities will be used as these are tailored for traditional e-learning tasks and they are mostly used for tasks that have a correct answer, which is written in this special type of activity.
- **Result** is used to return the result of the statement. This can be one of the predefined properties like score or an extension to define a custom result.
- **Context** is used to give the statement additional information to categorize it. The team property will be used to assign the statements to the session.
- **Timestamp** is used to record the time when the statement was executed. The timestamp is in the format ISO 8601.
- **Extensions** are used to extend either activities, result or context to gain additional possibilities to collect data.

T-Mon

Each of these elements has properties that are required, recommended, or optional for the xAPI specification.
In the paper by Alonso-Fernandez et al. [10] the platform T-Mon to analyze xAPI statements that are collected in Serious Games is presented. This platform enables an analysis that is not dependent on the design of the game, it just needs the collected xAPI statements. On these statements Learning Analytics is done in a general not game specific way. For example, the progress of the players through the Serious Game is shown, the scores the players have reached, and the time the players took for each part of the game. These analysis is displayed in a dashboard that uses different kinds of diagrams. For the analysis Jupyter Notebooks are used in T-Mon and the platform is freely accessible. The idea of T-Mon to have a generalized approach to Learning Analytics is helpful to get a fast understanding of some basic analysis on the collected xAPI statements. If a more in-depth analysis is needed, the Learning Analytics provided by T-Mon are not suitable anymore, as they only included general visualizations and do not take the game design into context. The visualizations used by T-Mon are informative and can be used to help designing the dashboard as well as some general analysis that can be useful for analyzing the collected xAPI statements from the game *transaction MTLG*.

2.4. Learning Locker

Description of Learning Locker

Learning Locker is a Learning Record Store (LRS) [14]. It is used to store Learning Data in xAPI format. It is possible to explore the data that are stored in Learning Locker with dashboards that the user can create. It is also possible to run simple queries on the collected xAPI statements. These queries are like SELECT clauses in SQL. The results of these queries can also be sent to another system, for example for an analysis. A lot of additional configurations can be made in Learning Locker, but these are not relevant for this master thesis.

Chapter 3 Previous Implementation

Chapter overview

In this chapter the two different implementations of the game this master thesis is based on will be presented. The first implementation is *transaction*. This Serious Game is used in a lecture. The second implementation is *transaction MTLG*, which is based on *transaction*, but simplified and implemented as a Collaborative Multitouch Serious Game.

3.1. transaction

Overview of transaction

The game *transaction* is a business game of the controlling chair of the RWTH Aachen University [15] [16], which is used as mandatory learning material for lectures. In *transaction* the player has to manage an automotive company. The player needs to buy parts to produce intermediates to then produce cars. For this the player has to hire personal and buy machines. The game is played over multiple periods in which the demand for cars changes.

Functionalities of transaction

Some additional functions are added that the player has to consider. Buying parts can be done from different deliverers with different prices, minimal buy amounts, and delivery costs. There are five different machine types that the player needs to buy for the production, while not overusing the factory space. There are four different types of personal. Three different types of loans, with interest rate, loan run time, and special repayment. For each of the different products to produce the player needs different machines and personal. For some of the products even multiple machines and personal are needed, while the machines and the personal have different production capacities. Parts, intermediates, and cars can be stored for the next period, with an individual storage price for each. It is also possible to sell items from the storage for a fixed price. Machines are depreciated each period and can be sold. The personal can also be fired for severance pay. Cars can be sold up to a defined demand each period. The player can see a balance sheet and profit and loss statement for additional information about the company. In addition the player gets an overview on the profit and loss as well as on the demand for cars, produced cars, and on the sold cars each period. The demand for cars is known for the next two periods. The sold and produced cars can also be compared to the average of all other players as well as on the capital. Each player can see its own ranking and the ranking of the best and worst players for every already completed period.

Additional notes

It is important to note that the player will not have enough money to buy all machines and to produce cars so a loan must be taken to produce enough cars. Since the demand changes each period, the player has to find a strategy to deal with uncertainty. In order to progress into the next period the players need to answer questions of the lecture and can watch information videos for this.

3.2. transaction MTLG

Overview of transaction MTLG

In the master thesis by Schablowsky [1] the game *transaction* was reprogrammed with the MTLG framework [17] to implement the game as a Collaborative Multitouch Serious Game. The game is played by multiple players together on a big touchscreen, where each player stands on one side of the touchscreen. Instead of one player managing the whole company each player has to choose which modules of the game he wants to control. The modules are:

- procurement
- storage
- personal and machines
- production
- final assembly and sale
- loan
- finance
- game information

Adaption of functionalities

transaction MTLG supports the same base functionalities as *transaction*. In order to make the game playable in one session and with multiple players *transaction* is simplified. The different options to buy parts are simplified to the bulk discount. The storage usage has the same storage cost for each element group. It is only possible to select one type of loan. There are only 2 types of personal as well as only one machine type. The production of cars is similar, only the amount and type of machines and personal is different. The option to sell cars is roughly the same even though the price for cars cannot change in *transaction MTLG*. Learning to use the balance sheet and profit and loss statement is not possible in *transaction MTLG*, as the finance module is a mix of both calculations. The depreciation of machines is also not supported in *transaction MTLG*.

Part II

Learning Analytics with xAPI for transaction MTLG

Chapter 4 Conceptual Design – Learning Analytics with xAPI

Chapter overview

This chapter describes the foundations that are needed to be able to apply Learning Analytics. At first the learning goals are defined based on *transaction* and *transaction MTLG*. These are needed to be able to define the different graphs of the Learning Analytics dashboard. Then the conceptual design of the xAPI usage is described. It is necessary to define the collection of data with xAPI to be able to interpret the data correctly in the analysis. At last the Data Flow through the systems is shown to be able to get the data to the Learning Analytics.

4.1. Learning Goals

Definition process

In order to define the learning goals of the game *transaction MTLG* it is helpful to look at the game *transaction MTLG* was based on. This is the Serious Game *transaction*, that is described in chapter 3.1.
It is necessary to look at the features of the game and deduct, which learning goals could be hidden behind them, because the learning goals of the inventor of the game are not accessible. For each of the following learning goals it is discussed in what extend they are applicable for *transaction MTLG*. The learning goals cannot be transferred directly as *transaction MTLG* is a simplification of *transaction*.

4.1.1. Deepen the Lecture Material

Lecture material

A learning goal of *transaction* is deepening and repeating the lecture material in another context. This is the case, because questions need to be answered to progress through the game. Since *transaction MTLG* is not played accompanying a lecture this learning goal is not relevant for *transaction MTLG* and this feature is not implemented.

4.1.2. Understand the Functionality

Functionality of transaction

Transaction has different functionalities for managing a business. One learning goal is to experiment with these functionalities and find out how they work. These functionalities are:

- different options to buy parts
- storage usage

- loan usage
- personal management
- machine management
- production
- sale of cars
- balance sheet
- profit and Loss statement
- depreciation

Functionality of transaction MTLG

In *transaction MTLG* these functionalities are also supported in a simplified way, thus the learning goals of the base game can be applied to *transaction MTLG*. The different options to buy parts are simplified to the bulk discount. The storage usage has the same storage cost for each element group. It is only possible to select one type of loan. There are only 2 types of personal as well as only one machine type. The production of cars is similar, only the amount and type of machines and personal is different. The option to sell cars is roughly the same even though the price for cars cannot change in *transaction MTLG*. Learning to use the balance sheet and profit and loss statement is not possible in *transaction MTLG*, as the finance module is a mix of both calculations. The depreciation of machines is also not supported in *transaction MTLG*. This leaves the following functionalities as learning goals:

- usage of bulk discount
- storage usage
- loan usage
- personal and machine management
- production
- sale of cars

4.1.3. Develop a Strategy under Uncertainty

Strategy

Another learning goal of *transaction* is creating a strategy to deal with the uncertainty of the demand for cars. This learning goal can be applied to *transaction MTLG*, because the uncertainty is the same in this game. The strategy the players need to develop is not as complex, because almost all functions are simplified from *transaction*. Nevertheless the basic idea on how to deal with the uncertainty stays the same.

4.1.4. Understand the Interplay of Processes

Interplay of Processes

This learning goal is that the player understands the processes of the company during each period of the game. This learning goal is also valid in *transaction MTLG* as the structure of the company is the same.

4.1.5. Apply Collaboration

Collaboration

The game *transaction* is a single player game thus no collaboration between the players is possible. The focus of *transaction MTLG* is the collaboration between the players. Thus it is important to try to measure the level of collaboration between the players.

4.2. Experience API – xAPI


```
{
  "stored": "2021-05-11T15:22:18.245Z",
  "active": true,
  "completedForwardingQueue": [],
  "failedForwardingLog": [],
  "client": "6042461789a583bc63816c5",
  "lrs_id": "6042536e789a583bc63816c7",
  "completedQueues": [],
  "activities": [
    "https://gitlab.com/learntech-rwth/xapi/-/blob/master/definitions/multitouch/activities/button.json#1_4o2"
  ],
  "hash": "9031410bbc64a77a466cb8c05243d768ddcb9fa8",
  "agents": [
    "mailto:player5533@mail.temp"
  ],
  "statement": {
    "authority": {
      "objectType": "Agent",
      "name": "transactionMTLG",
      "mbox": "mailto:transactionMTLG@mail.temp"
    },
    "stored": "2021-05-11T15:22:18.245Z",
    "context": {
      "team": {
        "objectType": "Group",
        "name": "Session5305",
        "mbox": "mailto:team5305@mail.temp"
      }
    },
    "actor": {
      "objectType": "Agent",
      "name": "Player 5533",
      "mbox": "mailto:player5533@mail.temp"
    },
    "timestamp": "2021-05-11T15:22:17.650Z",
    "version": "1.0.0",
    "id": "16a81ab1-af9f-4a68-8b55-a5b11a423d29",
    "result": {
      "score": {
        "raw": 600
      }
    },
    "verb": {
      "display": {
        "en": "pressed"
      },
      "id": "https://gitlab.com/learntech-rwth/xapi/-/blob/master/definitions/multitouch/verbs/pressed.json"
    },
    "object": {
      "definition": {
        "name": {
          "en": "Seats_Button"
        },
        "extensions": {
          "https://gitlab.com/learntech-rwth/xapi/-/blob/master/definitions/seriousgames/extensions/activity/buttontype.json": "decrementor"
        },
        "type": "https://gitlab.com/learntech-rwth/xapi/-/blob/master/definitions/multitouch/activities/button.json"
      },
      "id": "https://gitlab.com/learntech-rwth/xapi/-/blob/master/definitions/multitouch/activities/button.json#1_4o2",
      "objectType": "Activity"
    }
  },
  "hasGeneratedId": true,
  "deadForwardingQueue": [],
  "voided": false,
  "verbs": [
    "https://gitlab.com/learntech-rwth/xapi/-/blob/master/definitions/multitouch/verbs/pressed.json"
  ],
  "processingQueues": [],
  "person": null,
  "timestamp": "2021-05-11T15:22:17.650Z",
  "relatedActivities": [
    "https://gitlab.com/learntech-rwth/xapi/-/blob/master/definitions/multitouch/activities/button.json#1_4o2"
  ],
  "relatedAgents": [
    "mailto:player5533@mail.temp",
    "mailto:team5305@mail.temp",
    "mailto:transactionMTLG@mail.temp"
  ],
  "organisation": "5fe3195b2acc473c58ea29b4",
  "_id": "609aa12a9e9b9874508e0110",
  "registrations": [],
  "pendingForwardingQueue": []
}
```


Figure 4.1.: Example xAPI statement saved by Learning Locker

xAPI standard

xAPI is chosen as the standard to collect and store the learning data as it was presented in the chapter related work. In this section the usage of xAPI in the master thesis is explained.

4.2.1. xAPI Statement Design

xAPI specification

The design of xAPI statements is defined in the xAPI specification [13]. This section shows which possible objects, verbs, and activities are used in this master thesis to store the learning data of the learners. The used definitions can all be found in the git repository [18].

xAPI metadata

A sample xAPI statement that is stored in Learning Locker can be seen in figure 4.1. The xAPI statement is enriched with metadata by Learning Locker. The metadata is added to the statement. The core statement is in the sub JSON object "statement". This metadata contains additional information as well as redundant information from the statement to make it easier to process the statements without the need to parse the whole statement object. In line 2 Learning Locker added the stored attribute, which shows the time when the xAPI statement was stored by Learning Locker. This can be used to identify the lag between Learning Locker and the client that sends the statement. Furthermore, Learning Locker stores the activities, agents, verbs, timestamp, related activities, and related agents redundant. The other elements of the metadata are not used in this master thesis.

xAPI structure

The structure of the xAPI statement that is stored in the object "statement" is in accordance with the xAPI specification. The statement consists of:

- **Authority**, the authority that sends the xAPI statement
- **Stored**, the time the statement is stored in Learning Locker
- **Context**, the context of the statement. This is used to save the session the statement originated from in form of the team. The session id is stored in the name of the group object of team. The mail is added to keep xAPI conformance.
- **Actor**, the information about the user that made the action in the game that generated the xAPI statement. The objectType is agent and the name as well as the mbox contains the user ID from the game.
- **Timestamp**, the time the statement was generated in the game
- **Version**, the version of the xAPI statement
- **Id**, the unique identifier of the statement
- **Result**, the result if applicable of the statement, which is stored in score.raw
- **Verb**, which action the actor did. The verb contains the English display name, as well as the id, which is a reference to the definition in the git repository
- **Object**, contains the English name of the object. The type is a reference to the git repository, as well as the id. The id is a link. The link of the id has the unique number of the object in the game coded at the end behind the #. The objectType is activity.

Verb values

All e-mail addresses that are used in the xAPI statements are not useable addresses. They are still needed to have xAPI statements that are in conformance with the specification. The verbs and activities that are needed to describe all interactions of the players with the game could all be found in the vocabulary and no new definitions need to be added. The possible values for verbs that are used in this master thesis can be seen in the table 4.1. In both of the following tables "*" corresponds with "https://gitlab.com/learntech-rwth/xapi/-/blob/master/definitions".

Name	Id	Usage
Started	*/seriousgames/verbs/started.json	game or period is started
Finished	*/seriousgames/verbs/finished.json	game or period is finished
Synchronized	*/generic/verbs/synchronized.json	stage is synchronized
Asked	*/generic/verbs/asked.json	question for next period
Answered	*/generic/verbs/answered.json	answer to question
Selected	*/generic/verbs/selected.json	module chosen by player
Stopped	*/lms/verbs/stopped.json	stopped working on module
Changed	*/generic/verbs/changed.json	value of a slider is changed
Pressed	*/multitouch/verbs/pressed.json	button was pressed

Table 4.1.: xAPI Verbs

Object Values

The possible values for objects that are used in the master thesis can be seen in the table 4.2.

Name	Type	Usage
Game	*/seriousgames/activity/game.json	game that is played
Level	*/seriousgames/activity/level.json	period of the game
Stage	*/seriousgames/activity/stage.json	module of the game
Question	*/generic/activities/question.json	question for next period
Answer	*/generic/activities/answer.json	answer for next period
Draggable	*/multitouch/activities/draggable.json	sliders in modules
Button	*/multitouch/activities/button.json	buttons in modules

Table 4.2.: xAPI Objects

xAPI extension

The xAPI object Button has the extension "https://gitlab.com/learntech-rwth/xapi/-/blob/master/definitions/seriousgames/extensions/activity/buttontype.json", which is the buttontype of the button, to distinguish between buttons that increment a value and buttons that decrement a value.

4.2.2. xAPI Statement Collection

xAPI statement creation

For all actions by the user that should be logged, a call of the function createXAPIstatement() is added in *transaction MTLG*. This function gets the parameters playerID, elementID, val, and sessionID. These parameters are then evaluated by the function to include the current values of the parameters in the xAPI statement. PlayerID is used for the actor, elementID is used for verb and object, val is the value of the action if applicable, and sessionId is used for the team. The element and verb can both be calculated from the element ID, because each elementID is unique and the player can only interact with each element in a single way. After the values for the xAPI statement are calculated, the

function sendXAPIstatementToServer sends the xAPI statement to Learning Locker. The parameters for the connection are set in this function.

xAPI statement source

The xAPI statements are generated by actions of the players. Thus most of the statements will be generated by using the sliders and buttons to change the values in the game. Other xAPI statements will be generated at the start and end of the game and of a period. Also stage synchronization xAPI statements are generated at the end of each period. At the start of the game when the user selects a module a xAPI statement documents the selection. In addition to these xAPI statements, a statement is sent to the server when the user did not interact with a module for 3 seconds. Other xAPI statements are generated when the question for the next period is asked by one Player, as well as xAPI statements for the answer to this question.

Usage of xAPI statements

The collected xAPI statements are used to create charts for the dashboard to help the user to analyze whether the learning goals that were defined in the previous chapter are achieved. The xAPI statements of the sessions to be analyzed are requested from Learning Locker. Afterwards, the statements are processed to generate the data format the graphics need. This will be further discussed in chapter 6.2.

4.3. Data Flow

Section overview

This section shows the data flow between *transaction MTLG*, Learning Locker, and the webserver. The data flow can be seen in figure 4.2. On the left is *transaction MTLG*. This is the Serious Game the learning data are collected from. When a player interacts with the game a xAPI statement is sent from *transaction MTLG* to Learning Locker to store the statement. Learning Locker evaluates that the statement is in conformance with the xAPI specification. The webserver has a subroutine to request the learning data of the sessions. For this a start time, end time, and file name need to be specified. All statements between the start and the end time are requested from Learning Locker and then saved in a .txt file with the specified name. It is also possible to request statements from multiple sessions at once, but for clarity one session is saved in one file in the implementation. On start-up the webserver reads the data of the specified files and transforms the data. When a client connects to the dashboard the server creates the requested graphics out of the transformed data. Then the graphics are sent to the dashboard for display.

Data flow description

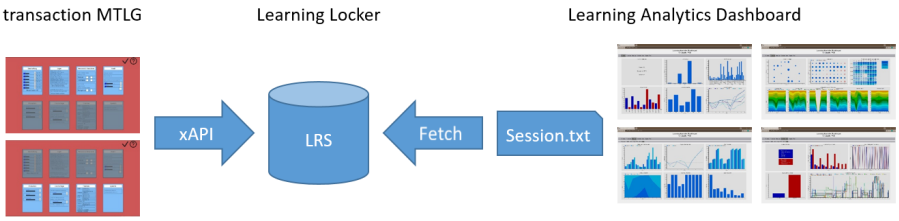


Figure 4.2.: Data Flow

Chapter 5 Conceptual Design - Dashboard

Chapter overview

This chapter describes the conceptual design of the dashboard. At first the paradigms that were used to create the dashboard are presented. Next, each of the graphic types that is used is described. At last, each graph of the dashboard is described and noted what information the graph gives in regards to the learning goals.

5.1. Paradigms and Design of the Dashboard

Visual Information Seeking Mantra

Shneiderman [19] presents an order how a visualization should support actions via the Visual Information Seeking Mantra: "overview first, zoom and filter, then details on demand". In the creation of the dashboard this quote is built-in, with the general tab as a landing page that gives the user an overview. The zoom and filter of the data is realized with the navigation of the menu by selecting (zooming on) the learning goal and filtering on the selected session. The user can on demand look at the specific values in the graphic by hovering.

Paradigms

In order to create a consistent dashboard the dashboard will be created by adhering to the following paradigms as much as possible:

1. slim and simple design
2. full page usage and no scrolling
3. no bright distracting colors
4. player 1 is blue, player 2 is red
5. one tab per learning goal

Slim design

The first paradigm "slim and simple design" means that no fancy functionalities are added to the dashboard, that the dashboard structure can be understood fast, and that it does not contain a lot of nesting. The dashboard of this thesis is designed following this paradigm. The dashboard has no functionalities that the user needs to learn, the structure can be comprehended fast, as the navigation is at the top and each tab has space for 6 panels, and the structure is shallow without unnecessary nesting. In order to reduce the nesting no element was added to capture the two rows of graphics.

Full page usage

The goal of the paradigm "full page usage and no scrolling" is that all information of one tab can be seen at the same time and that the user does not need to scroll. Furthermore, with this paradigm no space that could be used is left empty. In the dashboard this paradigm is applied as scrolling is not possible and all panels of the page are used. Only

the graphic test tab has a single open panel, because this tab does not belong to the core dashboard and is only used for an additional part of the dashboard usability study.

No bright distracting colors

The third paradigm "no bright distracting colors" is used to not scare the users off the dashboard, because it is too bright and flashy. In order to implement this paradigm a color palette is created that mostly uses blue hues. For the graphs that need a lot of different hues the color palette is extend over green to yellow to red. These colors are all still chosen with a low intensity so that they do not appear bright.

Player 1 blue, player 2 red

The fourth paradigm is that player 1 always gets the color blue and player 2 always gets the color red. This paradigm is used to emphasize the graphics that display data that belongs to one of the players instead of belonging to the session. This paradigm is implemented in the dashboard. The affected graphs are in the general and collaboration tab.

One tab per learning goal

The fifth paradigm is that each tab is dedicated to one learning goal. This increases the clarity of the structure of the dashboards and allows the user to be able to find answers to questions faster. In the final dashboard this paradigm could not be adhered to completely. The first problem of the realization of this paradigm is that the different learning goals do not have exactly 6 graphics each. Thus adhering to this paradigm would cause empty panels or pages that the user needs to scroll on. The second problem of this paradigm is that some graphics can be used to get insight into multiple learning goals. So these graphics either are only in the tab of one learning goal or duplicated for both. In the dashboard the graphics are ordered by the learning goal they belong to, but still some cross-referencing between the pages is necessary. Thus this paradigm could not be fulfilled completely.

5.2. Graphic Types

Section overview

This section describes the different graphic types that are used in the dashboard. The code base structure of the graphics is taken from Observable @d3 Gallery post, which is created by Bostock [20], except for the base of the bubble scatter plot which is taken from [21] by Holtz.

5.2.1. Bar Chart

Bar chart

A bar chart is used to compare the values of the different items on the x-Axis with each other. For this each value of the x-axis has a bar with a fixed width and a height corresponding to the value on the y-axis. This graph type gives a fast overview of the difference of these values. When a bar is hovered the value of the bar is shown. The basic graph is by Bostock [22].

5.2.2. Grouped Bar Chart

Grouped bar chart

A grouped bar chart is similar to the bar chart, but for each value on the x-axis the grouped bar chart has a group of bars. These bars are in this dashboard all evaluated with the same y-axis. The positioning of the bars shows which of the bars belong to each value of the x-axis. In order to know what each bar is representing, each bar has a different color and a legend is added. A hover function for the values of the bar is added. The basic graph is by Bostock [23].

5.2.3. Multiline Graph

Multiline graph

A multiline graph displays multiple lines to compare the values of the lines to each other. Each line has a different color and a legend shows what the line is displaying. If the y-axis of the graph contains negative values a grey dashed line is added at the value zero to increase the readability of the graph. The basic graph is by Bostock [24].

5.2.4. Bubble Scatter Plot

Bubble scatter plot

A bubble scatter plot is used to show the relationship between three variables. The first variable is on the x-axis, the second on the y-axis, and the third in the size of the bubbles at the intersection of the x and y-axis. The volume of the circle is used to display this third variable. This causes a circle with twice the radius to represent a value four times as big. It is also possible that multiple bubbles are in the same place with different colors to add another dimension. Then a legend is added to show what each color represents. Hovering a bubble shows the value of the bubble. This graph types allows a quick comparison of multiple variables with each other. The basic graph is by Holtz [21].

5.2.5. Normalized Stacked Area Chart

Normalized stacked area chart

A normalized stacked area chart shows how different values are in relation to each other. For this the values are represented by areas that are stacked on top of each other to fill the whole area. Each area has its own color and a legend as well as hovering shows what category the area shows. This graph shows the development of the values over the progression of the x-axis. The basic graph is by Bostock [25].

5.2.6. Heatmap

Heatmap

A heatmap is used to compare the variables like the bubble scatter plot. Here the different values are not represented by the size of the bubble but by the color. In the dashboard a table with 3 values per table item is enhanced by using color density as an indicator for the value. The higher the value the greater the density. This graph is developed from scratch.

5.3. Dashboard for transaction MTLG

Section overview

In this section each graphic that is used in the dashboard is described and it is explained how the graphic can be used to analyze how the learning goals of the game were reached by the learners. This chapter will describe the graphics in the dashboard tabs from left to right starting with the general tab.

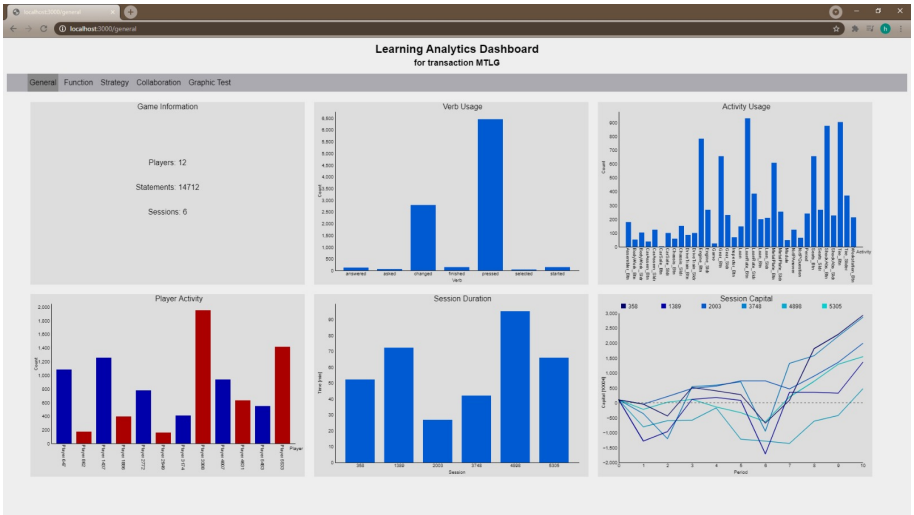


Figure 5.1.: Dashboard General

5.3.1. General

The general dashboard tab (figure 5.1.) is used to give an overview over the sessions and the xAPI statements that were collected. This dashboard is used to give an overview of the collected data. This is close to the idea of T-Mon by Alonso-Fernandez et al. [10]. The panels are ordered from left to right and top to bottom.

Top row

The first panel shows how many unique players and sessions are in the collected learning data as well as the number of collected xAPI statements. The second panel is a bar chart that shows how often the different actions were performed by all players. This gives an overview how the game was played. The third panel shows with which objects the players interacted in the game and how often. This shows which elements had a higher usage due to more accurate setting of the value.

Bottom row

The fourth panel shows the activity of each player in the game. The first player of each session is blue and the second player is red. The color coding is used in all graphs were data is shown that belongs to a single payer. This graphic shows how the workload of game interaction is split between the players. The fifth panel shows a bar graph of the duration of the different sessions. The last panel shows how the capital of all sessions changed over the periods. This can be used to compare the performance of all sessions to each other as the goal of the game is to maximize the companies capital.

5.3.2. Function

The function dashboard tab (figure 5.2.) shows how the different functions of *transaction MTLG* are used by the players of all sessions.

Top row

The first panel shows a bubble scatter plot showing how many cars each session stored per period. The number of stored cars is shown by the volume of the circle. This graphic shows how the function to store cars is used by the players. The second panel also contains a bubble scatter plot, but here three bubbles are shown on top of each other. The bubbles represent the loan, installment, and cash values of the session per period. The color of cash has an opacity to make it possible to see all three circles, since the installment circle cannot have a greater volume than the loan circle. If no loan circle is displayed the installment was equal to the loan, which means that the loan was paid back completely. This panel can be used to evaluate how the loan function was used by the players. The

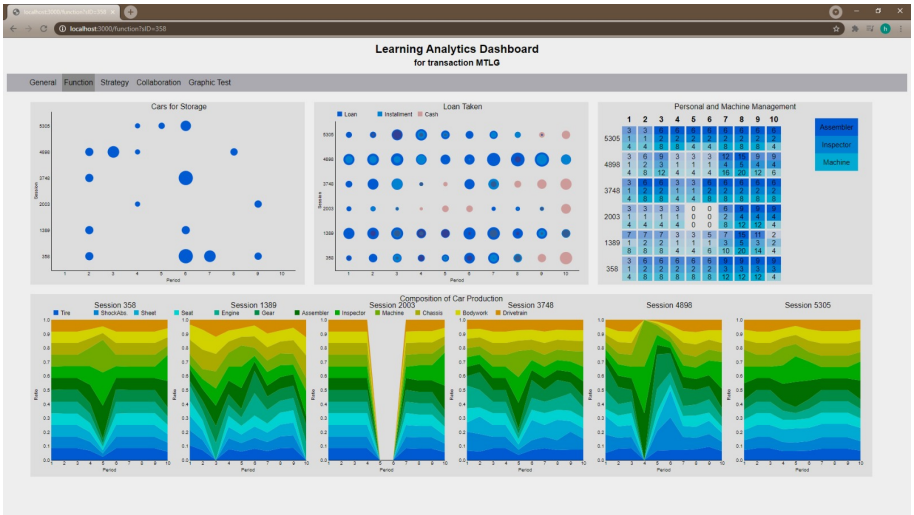


Figure 5.2.: Dashboard Function

third panel displays the number of assemblers, inspectors, and machines throughout the game for all sessions in form of a table. Each table item has a color coded entry for assemblers, inspectors, and machines. The intensity of the color gives additional support for the reader to see the relation of the value to the maximum of each row. This panel can be used to analyze the use of the personal and machine function.

Bottom row

The panel in the bottom shows how the production components can be used to produce cars in relation to each other for each period. Every session has its own graphic. If all areas have the same height the production materials are all equally available. If the areas have different heights the production components are not equally available. The color differentiate the different production materials and a legend as well as hovering shows the production component the area belongs to. This panel can be used to find out how the management of material is done by the sessions.

5.3.3. Strategy

The strategy dashboard tab (figure 5.3.) shows graphics that help to evaluate the strategy of the different sessions. Thus every session has ist own strategy dashboard page, which can be selected using the filter below the site navigation.

Top row

The first panel shows a grouped bar chart. This chart shows the produced cars, sold cars, and demand for cars in every period. These three categories are also in the legend of the graph. If a bar is hovered the exact value of the bar is displayed. This graphic can be used to analyze how the production changed in relation to changes in the demand. In the second panel a multiline graph shows the capital development of the current session as well as the mean capital of all sessions per period. Here the strategy of the session can be compared with the average of all sessions. The third panel shows the bulk discount usage over all periods in a grouped bar chart. The min price bar shows the price for buying all parts with bulk discount even if a bulk discount is not available for the players. So the bar displays the theoretical minimal cost. The max price bar displays the cost for buying all parts without bulk discount. The actual price displays the price that was paid in the period to buy the parts. This graph displays the savings of using the bulk discount in relation to min and max price.

Bottom Row

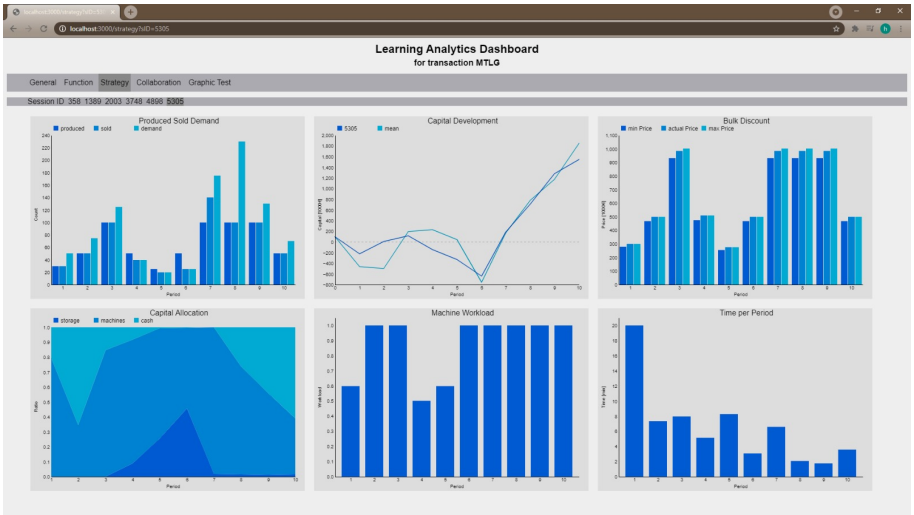


Figure 5.3.: Dashboard Strategy

Figure 5.4.: Dashboard Collaboration

The fourth panel displays for what the sessions use their cash. The normalized stacked area graph consist of cash, machines, and storage and displays their relative relation to each other. Hovering and the legend show which area represents which value. The fifth panel shows the machine workload in percent for each period in a bar chart. This graph displays how the strategy of the players used their production potential. The sixth panel shows the time in minutes per period in a bar graph. This graphic can show where the strategy needed to be adapted when the period time is longer. The graph also shows how much the players learned to play the game efficiently with their strategy.

5.3.4. Collaboration

The collaboration dashboard tab (figure 5.4.) shows the graphics to evaluate the collaboration of the players during each session. Like the strategy tab this tab is also filtered by session. In this tab the data concerning the players is displayed in blue or red in accordance to the player color like it was in the general tab.

Top row

The first panel displays in a bar chart which player asked whether to progress into the next period. This graph can be used to get an indicator whether one player is leading how the game should be played. The second panel is a multiline graph that shows whether a player was active in a 30 seconds window. If a player interacted with the game the line jumps to active if no action by the player happened the line stays at inactive. This graph can be used to find times where both players were discussing how to proceed with the game, as well as to be able to see whether one player had more activity than the other. The third panel displays the module selection of both players. This can be used to set the activity times and amount of the players in contrast to the modules they choose.

Bottom row

The fourth panel is a grouped bar chart that displays the number of activities by each player in each period. This graph can be used to see how many actions were needed to adjust all values for the current period. The last panel displays how the values of all inputs changed over the course of the game in a multiline chart. This chart can be used to see in which order the values were adjusted and changed in each period. Furthermore, if values were corrected during a session it is possible to see from which module the change was initiated.

Chapter 6 Implementation

Chapter overview

This chapter will at first present the upgrades to *transaction MTLG* that are needed to be able to conduct the study to collect the learning data. Then the dashboard implementation will be described.

6.1. Upgrade of transaction MTLG

Section overview

This section will describes the changes to *transaction MTLG*. The first part deals with the upgrade to allow to connect to a single game instance from multiple clients and to play the Serious Game collaboratively without the need for an in person meeting. The second part describes the modes selection to evaluate the collaboration for these modes. The next part describes the usability improvements of *transaction MTLG*. Then the bugs are listed that were fixed in order to collect the data without having runs that need to be discarded, because a bug appeared. And at last the fine-tuning of the game configuration is explained.

6.1.1. Online Multiplayer

Reason for online multiplayer

During the current COVID-19 situation *transaction MTLG* needs to be extended in such a way that it is possible to play the game collaboratively from multiple clients, as it is not possible to have multiple persons stand around the large touch display to play the learning game together. The basic idea is to play the game with a browser and to communicate over a call. This way it is possible to play the game collaboratively without the risk of an infection.

Architecture difference

In figure 6.1. the difference in the architecture between the basic version and the expanded version can be seen. On the left side is the original architecture and on the right

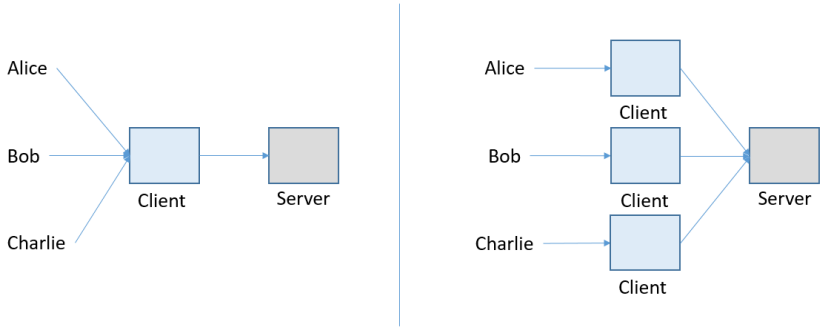


Figure 6.1.: Communication comparison

side the new architecture. In the old version all players were playing on the same device, which is the only client that communicated with the server. It is possible to have multiple clients connected to the server, but it is not possible for two different clients to join the same game. On the right side the new architecture is shown were each player has its own client which connects to the server. Each client can join a game session started by the server and thus multiple clients can play the game together.

Value synchronization

Each time a value is changed in the game with a slider or a button, the change is send to the server. The server than sends a notice to all other clients that are connected with the game session to notify them of the value change. This allows seeing the values update when another player changes them as well as data synchronization between the clients. When the game is started the server waits until enough players are connected to a session before the server starts the game for the clients. Finishing a period also needs all players to agree, so the server checks whether all players want to progress to the next period.

6.1.2. Mode Selection

Different modes

In the basis game every player can see the modules that the other players have chosen. These modules do not have the correct orientation for the player to read them properly, but each player can still see the information that is shown on them. The first version of the upgraded game did not allow the user to see the modules that were selected by other players. For example in this version, the player that has chosen the module "finances" needs to tell the player, who buys the parts how much money is available. In order to reduce this communication overhead another mode is added where the modules other players choose are displayed greyed-out. Then everybody has all information, but can only change the values in the chosen modules. In the settings it is possible to toggle whether the modules that other player chose are displayed greyed-out or not displayed. This is helpful to research in which case the collaboration between the players is better. This research is not subject of this master thesis.

Mode implementation

To add the greyed-out modules, in addition to removing a module from the list of available modules, an additional greyed-out module is created for all other users in the same game session. The server sends all updates on the parameters of the game to each client, so the communication between the clients and the server does not need to be extended. The update functionality to changes sent by the server is included when a module is created whether or not it is greyed-out. So greyed-out modules are updated like other modules. Only the methods for the input of the module are not created and the grey box above the module is added.

6.1.3. Usability Improvements

Usability Improvements

During the test runs of the upgraded game some difficulties with the game interaction were discovered. The values that display the bulk discount were too small. To make it easier to read the bulk discount the field size was increased as well as the font size. It was also not possible to see with which period the game ends before progressing to the last period. Thus when the demand for the last period is displayed in the game information module a text note is shown to notify the player of the last period. As it was not easy for some players to drag the sliders to the correct position in the modules production and final assembly and sale, buttons were added next to the sliders like it was done for the procurement.

6.1.4. Bug Fixes

Known bugs of transaction

In the game *transaction MTLG* are two known bugs. These two known bugs are not fixed in this master thesis, because it was possible to avoid them. These bugs are an error on loading the game, as well as when changing the position of the modules.

Module moving bug

The module moving bug is happening when multiple players try to move modules at the same time. Then the modules that were not moved stay greyed-out for one of the players. Another aspect of the module moving bug is that the values from the player that moves modules do not update on changes by other players. This is a scenario that is likely to happen as one player might not take part in changing modules and thus change a value during this time. With this bug it becomes possible to produce and sell cars without the need to actually buy the parts the car needs. It is even possible to sell cars that were not assembled.

Game loading bug

The game loading bug that does not display all values correctly happens when a session is started after another session was played. In this case some values of the old session are taken into the new session, which causes incorrect calculations. These two bugs were fixed by making it impossible for them to appear in the game execution of the study. The first bug was avoided by disabling the module movement. During the playthroughs the players did not miss the functionality to move the modules. The second bug was avoided by letting the players play the game in one continuous session without interruptions.

While testing *transaction MTLG* the following new bugs were discovered:

- Intermediate products are not stored
- Loan calculation
- Finance module value names
- Game set-up

Intermediates not stored bug

The bug that does not store intermediate products occurs when the players do not use all intermediate products to construct cars out of them. All intermediate products that should have been produced, but were not stored are then reverted back to parts. So it was not possible to incorporate intermediate production and storing into the strategy and thus a module to produce the intermediate products was obsolete as only the car assembly was relevant in regards to part consumption.

Loan calculation bug

The loan calculation bug occurs when the players take any loan. The bug has two different aspects. The first is that the loan does not impact the balance of the company negatively. This means that the players can take an arbitrary loan to increase their company's value. The second aspect is that this would not cause a bug if the loan needs to be paid back over the duration of the game. Since the player can select the installment each period anew, the loan does not need to be repaid by the players. This bugs makes it impossible to compare the company results against each other. To fix this bug the loan is added to the finance module to be included in the end result of the company.

Finance module value names bug

This bug means that the names of the values in the finance module were not chosen correctly. The incorrect names are cash (in the game: "Mittel") and balance sheet (in the game: "Bilanz"). The value cash shows the value of the previous period's balance sheet value. This value does not change over the period and is only a reminder of the cash that is available. The name balance sheet for the value it displays is wrong. A balance sheet is used to display the financial condition of a company. For this the balance sheet contains multiple values and two different sides: assets and liabilities plus equity. A balance sheet is not a single value, but a calculation of different values. In order to clarify this the two

used value names were removed and the values for loan, cash and capital were added. The loan contains the total loan of the company. The cash contains the previous periods' cash minus the expenditure plus the revenue and the taken loan. The capital is the cash minus the loan plus the sell value of the machines. So the cash is the money the players have left and the capital makes it possible to compare different sessions. It is important that the calculation in this module is not used this way in business administration, it only gives an overview of the company's financial situation.

Game set-up bug

The files that were delivered with the game were not enough to start the game. It was possible to get the game running so that the main menu of the game was reachable, but it was not possible to start a session. In the main menu the players cannot register a moderator to login. And without a moderator that can create a session the game could not be played. In addition to that no players were defined too. In the database no entries for the players and moderators were made. The players can be added by hand to the database, but this is not possible for the moderators as their password is saved as a hash value in the database. Thus the code had to be changed to write a moderator to the database. To remove this bug a database image will be included in the submissions, this way players and a moderator are available to use in the game.

6.1.5. Fine-tuning of Game Configuration

Configuration base idea

The configuration of the game was changed to challenge the players with decisions that can show how they use the functionality, as all functionalities are relevant at some point in the game with the game configuration. So comprehensive calculations were done to tune the demand, sell value of cars, all values of the machines (buy, sold, and maintenance), and the bulk discount as well as the starting funds. All other values were not changed, especially the amount of base products to create the intermediate products was not changed.

Configuration changes

The starting funds were changed to 100,000€ so that the players do not have enough money for the first period and need to take a loan if they want to produce something. The price for the machines was increased to 200,000€, the sell value was increased to 95,000€, and the maintenance cost to 40,000€ per period. The amount per sold car was increased to 19,000€. The storage cost of intermediate products was set to 120€ so that it is cheaper to store bodywork than the parts of bodywork, the same cost to store the chassis compared to the parts, and more expensive to store the drivetrain than the base parts. This allows players to make a strategy were it can be sensible to store intermediate products.

Period	1	2	3	4	5	6	7	8	9	10
Demand	50	75	125	40	20	25	175	230	130	70

Table 6.1.: Demand per period

Impact of demand changes period 1-5

The demand change over the periods can be seen in table 6.1. The demand is shaped in such a way that the players are confronted with different situations. It is important to keep in mind that the players can always see the demand of the current and next period in the game information. The first period is kind of a tutorial as the demand is 50, which is exactly one production increment, since the machines and employees can each produce 50 parts or cars. The second period has a demand of 75, which in addition with the first period does not have a need to pre-produce cars for. When the players are in the second period they see the demand of 125 in the next period. Now they can either increase their

production limit to pre-produce cars or keep their production of 50 to play it safe. As it will be explained later, if the players calculate it out it is better to increase the production. When the players progressed to period 3 they can continue with the strategy from period 2, as the demand does not rise and the machines are used to fulfill the demand of the current period. Here it is not sensible to buy additional machines. When the players are in period 4 they have a difficult decision if they increased their production limit to 100, because the demand for the current and next period is 40 and 20. If a machine does not produce cars for 3 periods it would be better to sell the machine, but the players can only see 2 periods. So it is a gamble whether to keep the unused machines. When the players progress in the next period they can still only see a low demand. If they did not reduce their production to 50 the same question as in the previous question is asked again. But to sell the machines that are not used now is not good even though the players cannot see it, so selling machines in period 5 cannot be considered an error by the players, just as being unlucky.

Impact of demand changes period 6-10

In period 6 the players can either chose to pre-produce cars for the next period or not. Since the combined demand of both periods is 200 it seems to be a good way to increase the production to 100 and store 75 cars for the next period to fulfill the demand completely in both periods. In period 7 the players can see that the demand rises again to 230 in the next period. If they already have the capacity to fulfill the demand for this period they probably should wait to see how the demand changes in the next period but one. If they cannot fulfill the demand completely it seems appropriate to increase the production this period. In period 8 the players see that the demand decreases by 100 to the next period. If they intended to buy machines to fulfill the demand of the current period better they should now rethink that as the machine workload of machines above a production limit of 100 would not be at 100%. When the players progress to the 9th period they see that the game ends after the next period and that the demand of 70 is even lower. Thus now they need to find a good way to use their production capacity to end the game in the best way as the information the players now have is perfect, and to keep in mind that the storage does not contribute to the end result. Thus the storage should be as empty as possible at the end. When they progressed to the last period they just need to stick with their strategy from period 9. It is important to note that it is now sensible to sell all unused machines as they still cost maintenance, but not to fire employees as their severance pay is higher than their salary.

Demand impact conclusion

Overall the first 3 periods are straightforward, while in period 4 and 5 the players need to find a way to deal with the uncertainty of the low demand and whether it rises again and whether or not to sell their machines. In period 6 to 7 the players need to deal with a strongly increasing demand and in period 8 again with a strong decrease. In the last two periods the players have to deal with a decreasing demand as well as finishing the game.

Bulk discount changes

The bulk discount was changed that it is easier for the players to recognize that it is cheaper to buy the parts in bulk with discount than buying without the discount, if the parts that are stored are used in the next period. Therefore, the discounted price is 15€ and more below the full price. Only tires and seats have a discount of 10€ per item. But it is still cheaper to buy the material in bulk if at least one third of the bought parts are used in the current period. So for seats only 68 seats need to be used and for tires only 100 need to be used. This equates to 17 and 25 produced cars respectively. Thus the bulk discount on these two parts can be used almost every period. The bulk discount for engines and gears is the opposite way around. Since the discount is 100€ per item for engines and 300€ per item for gears the parts can be stored a lot of periods before it is more expensive than only buying the parts on demand, even though it might be necessary to take a loan to

Category	Name	Base	Factor	Value
Parts	Tire	125	4	500
	Shockabs.	120	4	480
	Sheet	110	5	550
	Seat	220	4	880
	Engine	4300	1	4300
	Gear	3300	1	3300
	Total			10010
Production	Machine	40000	0,08	3200
	Inspector	2800	0,02	56
	Assembler	2800	0,06	168
	Total			3424
Sale	Car	19000	1	19000
	Total			19000
			Total Profit	5566

Figure 6.2.: Calculation of profit per car

buy these parts. The flat discount allows storing the gears for 20 periods. If a loan needs to be taken for the full amount and the gears are stored for all 10 periods the interest for 1 period is less than the price to buy the parts instantly. Thus since not all parts need to be stored and the loan can be repaid or is not necessary at all it is almost in all periods sensible to buy gears with the bulk discount. For engines it is a little harder to estimate as a discount of 100€ only allows storage for 4 periods, but since the amount of engines that needs to be bought is 120 and not 180 as for gears it is also sensible to buy engines for storage in almost all periods. The only exceptions are periods 4 to 5 as the demand in this period and the outlook is not high enough to buy such a high stock with no risk. Thus the bulk discount can be used a lot over the course of the game with a small increase in profit.

Profit per car

With the presented values each car that is produced if the machines are fully occupied generates 5566€ profit if it is sold in the period it is produced. The calculation can be seen in figure 6.2. Given this profit per car it is sensible to buy an additional production capacity of 50 in period 2 as the cost of 105000€ per machine is earned by selling 75 cars in the period they were produced and 25 cars after being stored for one period over period 2 and 3. Then the players made 117850€ over not fulfilling the demand even without usage of the bulk discount. In the calculation the price for the personal and interest rate is not included, but even if they are included the players will still make a profit with increasing their production to 100 in period 2.

Configuration enables learning goals

Another important consideration behind setting the values as described was that each learning goal is included in a run to evaluate the players based on the achievement of the learning goal. The usage of the functionalities is given as in the game each functionality can be used at a specific period to increase the profit of the company. The strategy under uncertainty is also given simply by design, as only the demand of the current and next period is shown. In addition, the strong changes in the demand as well as the low

demand in period 4-6 leads the player to make assumptions on how the demand will change and how to adopt to these changes. The interplay of processes and collaboration is supported by these values through a demanding game as a lot needs to be calculated and all different aspects need to be remembered.

6.2. Dashboard

Section overview

This section first describes the architecture of the dashboard to give an overview on the architecture. The MVC pattern is used and the data handling is described. Then the implementation of the dashboard is described with the data transformation and the creation of the graphics.

6.2.1. Webserver Architecture

MVC pattern

The web server for the dashboard is written in JavaScript as a Node.js express web server. The server follows the MVC design pattern. The base structure of the server is taken from Maboud's GitHub project [26]. The server is started with the server.js file, which uses the expressManager.js to create the router to listen to the URLs the server should serve. The URLs are in the src/router/index.js file. Then using the MVC design pattern each web page has a controller that manages the user interaction and site display a model to get the required data and view to display the site to the user.

Learning data fetching

The data can be collected from Learning Locker with executing the function getXAPIStatementsFromLRS(). In this function call the timestamps of start and end of the session need to be set. The received data is stored in a text file on the server to reduce the traffic to the Learning Locker. Queries of Learning Locker to forward specific data are not used to keep full control over the data flow and ensure reproducibility over time even when the Learning Locker instance is not accessible anymore. Thus the data stored in Learning Locker is also saved in files in the project to keep the complete data accessible.

Dashboard data transformation

The dashboard data is processed by the class transformData. This function takes all statements that were collected from the LRS and processes them into the format the graphics need.

Graphic construction

The graphics are built on the server side for the following 3 reasons:

- to reduce the amount of data that needs to be sent to the web browser
- to speed up the site load speed as the graphic is not built by the browser
- to not allow access to the raw data by the clients for reasons of data protection

The drawback of the server side render is that interaction of the graphics is not as easy to realize. Thus during the dashboard usability study a focus point is whether interaction with the graphics could increase the usability. The view of the server uses includes for the menu and filter to allow reusability of these website elements. As template engine pug is used.

6.2.2. Implementation

Additional data transformation

The architecture above is implemented in the web server. In addition, the data retrieved from the LRS needs to be transformed to the format the graphics need. Then the graphics need to be created from these data as svg and then sent to the browser that requested them.

Data transformation

The data transformation iterates through all statements and aggregates them to get the needed information to build the graphics. Each graphic that is displayed has its own dictionary with an entry for every session. After the initial processing on the server startup the data only need to be processed to generate the graphics. This is fast enough that the graphic generation does not need to be precomputed like the data transformation. The data transformation is written with the objective that it is more important that the code is readable than to optimize the execution time. Thus it takes some time to transform the data on startup, but since this is a onetime calculation the time loss is acceptable with the benefit of increased readability of the code.

Graphic creation

The graphics are generated by the server when a page is requested by a browser. For this the controller requests the graphic from the model with the correct data. Each graphic type has its own class that is an extension of the chart class. This enables reusability of code and individual differences between the graphs. With this concept it becomes easy to add additional graph types. The new graph type just needs to extend chart and then have its own function to display the data. Then if the data is passed to the graphic in the right format the graph can be created by the model and delivered to the view through the controller.

Part III

Evaluation

Chapter 7 Learning Data Collection and Analysis

In this chapter the collection of the learning data with a study will be described as well as the analysis of the collected data with the dashboard. The goal of the study is to get learning data and a self-assessment of the players to be able to compare the analysis with the player's estimation of their performance. The Learning Analytics performed with the dashboard are of the types Teaching Analytics and Measurement Analytics as defined by Ifenthaler [5]. As the Learning Analytics focusses on measuring the performance of the players as well as possible enhancements of the game to improve the teaching of the game.

7.1. Study

Section overview

This section describes the study that was conducted to collect data to be able to evaluate the dashboard with data generated by players. As previously mentioned the study needs to be conducted without the players meeting each other due to the current covid-19 pandemic. Thus the game was changed to be played over the browser. It was decided that the game is played by two players together to reduce the amount of participants. The study is started with declaration of consent, followed by a quick tutorial, then playing the game, and finished with a questionnaire.

Study participants

The study has 12 participants that play 6 games with two players each. Of the participants 7 are male and 5 female. 9 of the participants are in the age group 18-26, 1 in the age group 27-35 and 2 in the age group 54-62. 8 participants are attending university, 2 finished their studies, 1 finished training and 1 finished school. 2 of the participants were from business administration, 2 from social sciences, 2 from engineering, 2 from medical, 2 from natural science, and 2 did not disclose their background. 3 of the participants already completed a business game and 6 participants played in a test run of *transaction MTLG*.

The study started with the users reading and signing the declaration of consent to stay in conformance with the DSGVO. This is relevant for the personal data that is collected in the questionnaire at the end of the study. The declaration of consent can be seen in appendix B.1.

Game tutorial - set up

Following the declaration of consent the tutorial is shown, which gives an explanation on how to start the game and an overview on the functionality of the modules. The tutorial is in appendix B.2. The tutorial is in German since all participants are native German speakers. The tutorial consists of screenshots of the game with additional information in speaking bubbles. The tutorial is presented to the participants. The first two pages of the tutorial show how to start the game. The third page shows the start of the game. In this

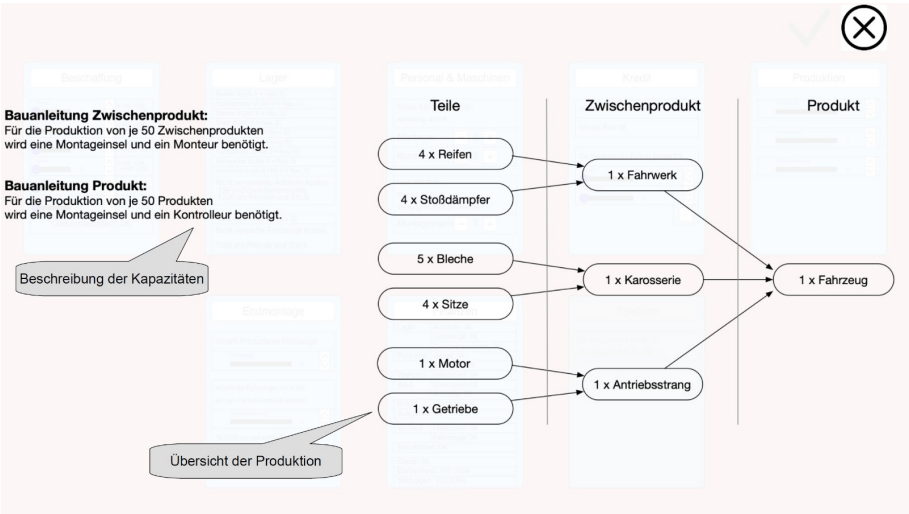


Figure 7.1.: Instruction by Schablowsky [1]

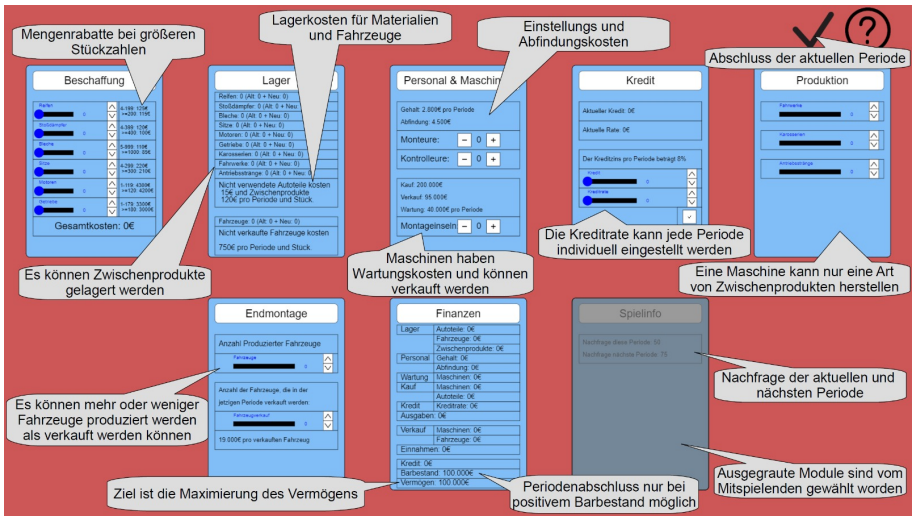


Figure 7.2.: Game Overview

game state the players decide how they want to distribute the different modules. There are no constraints on the module selection. It is only said that the participants should distribute the modules rational. It would be possible that one player chooses all modules and the other player only watches. No constraints were added, because this can possibly allow insights in the module selection.

Game tutorial - instructions

The fourth page, which is shown in figure 7.1., shows the information graphic to describe how cars are produced. This information graphic is taken from Schablowsky [1]. On the right the composition of the car is shown. A car needs the three intermediate products chassis, bodywork, and drivetrain. These can be produced out of parts. 1 chassis needs 4 tires and 4 shock absorbers. 1 bodywork needs 5 sheets and 4 seats. 1 drive train needs 1 engine and 1 gear. The text on the left shows that 1 machine and 1 assembler can produce 50 intermediates. And 1 machine and 1 inspector can produce 50 cars. It is important to note that a machine can only be used to produce one type of product, a machine already producing a chassis cannot produce drivetrains, bodyworks, or cars even if the machine is not fully occupied.

Game tutorial - game

The fifth page shows a possible game state and is used to describe the different modules. This page can be seen in figure 7.2. The speaking bubbles on this page focus on giving additional insight that the players might not notice otherwise. For the first module production (ger: "Beschaffung") the bulk discount is in the focus. The second module storage (ger: "Lager") has the information that all parts, intermediates, and cars can be stored with storage cost. The third module in the top row is personal and machines (ger: "Personal und Maschinen"). The speaking bubbles emphasize the wage and severance pay as well as the possibility to sell machines. The loan (ger: "Kredit") module has the information that the installment can be individually adjusted each period. The last module in the top row is production (ger: "Produktion"). The information that each machine can only produce either intermediates or cars is repeated here. In the bottom row the module final assembly (ger: "Endmontage") has the note that a different amount of cars can be produced than the parts that were produced in the previous modules, if the materials were stored or are to be stored. The finance module has the hint that the cash (ger: "Barbestand") needs to be positive to advance to the next period and that the goal is maximizing the capital (ger: "Vermögen"). The last module is the demand preview (ger: "Spielinfo") here the note is added that greyed out modules are selected by teammates.

Study finalization

After the tutorial the game is played by the participants over 10 periods. All playthroughs have the same predefined values. After the game a questionnaire is given to get feedback about the game and a self-assessment to the learning goals in order to set them in relation to the analysis of the data via the dashboard.

7.2. Analysis via Dashboard

Section overview

In this section the learning data collected from the game will be analyzed in regard to the learning goals of *transaction MTLG* that are presented in chapter 4.1. At first Game Analytics of the game are presented with the general tab. After this each learning goal is analyzed with the dashboard. In the order function, strategy, interplay of processes, and last collaboration. The graphic test will be analyzed with the dashboard usability study in chapter 8.2. All pages of the dashboard can be seen in appendix C.

7.2.1. General

Subsection overview

In this subsection the graphics of the general tab will be analyzed. This tab is used to get an overview of the sessions. Most analyses of this tab are of the category Game Analytics. To explain the behavior of some of these graphics, results of graphics of other tabs will be included. The main result of this analysis is that it was difficult to enter the desired values with the buttons and sliders. This section analyzes the 6 sessions with 2 players each. In total 14,712 statements where collected from these sessions and analyzed.

Verb Usage graph

The graphic "Verb Usage" shows the verbs of the xAPI statements. The verbs that were mostly collected are 6,453 times pressed and 2,784 times changed. The other verbs are collected far less. This is the case, because these two verbs are used to adjust the values of the modules during each period. The other verbs answered, asked, finished, selected, and started are from statements generated by the game control structure, which starts and finishes the periods and game as well as the initial module selection. It is interesting to note that the number of statements with the verb pressed is approximately twice as big as the number of changed verbs. This indicates that in the current game implementation it is tricky to correctly enter the desired values with the slider.

Activity Usage graph

The "Activity Usage" graph in the top right shows how often each activity is the object of the xAPI statements. The seven activities that were used more than 500 times are

Engine_Btn, Gear_Btn, LoanRate_Btn, MetalPlate_Btn, Seats_Btn, ShockAbs_Btn, and Tire_Btn, which are all buttons to change the values of the procurement as well as the button to change the installment. The corresponding sliders to these button also have high activity, as they are all in the 8th to 15th highest bars. The other bar in this range is the Period bar, which correspond to starting and finishing a period, which is players(12)*(start+finish)(2)*periods(10)=240 appearances. It is interesting to note that the LoanRate_Btn and LoanRate_Sldr have the most activities, which can signal that the players either changed the installment often, or that it was especially hard to set this slider as wished by the players. It could be easier for players to enter the values and thus make the game more enjoyable if the way to input these values is changed. Changing the loan value with the slider was easier for the players, because the Loan_Btn was pressed less often than the Loan_Sldr, because the players did not take the loan in such a way that their cash was lowest, or that the change of 5,000€ on one button press helped to reach the desired value faster.

The Assembler_Btn, Inspector_Btn, and Workstation_Btn were pressed an expected amount of times, which shows that not many uncertainties were happening with adjusting these values.

The sliders and buttons for the production of the intermediates and car production were used less than the procurement buttons and slider. Since the number of presses of the button is far less than the usage of the slider the value could be adjusted well enough with the slider and button combination, since the range on the slider is smaller in all periods than the range on the slider for the procurement. Most times these sliders were just set to max by the players.

It is interesting to note that the CarSale_Btn to change the value of cars that the players want to sell was only used 4 times. This means that it was easily possible to set the desired value with the slider. This is the case, because in most periods all cars that were produced were sold as well and thus the slider just needed to be set to max.

The last four activities Game, Module, NxtPQuestion, and NxtPAnswer are used for the game control and have an almost fixed amount of occurrences during a session.

In summary it is not easy to enter the desired values with the sliders thus the buttons needed to be used often. The installment value was changed the most times, but sliders in general were hard to set on the right value. The CarSale_Sldr was most of the time just set to max.

Player Activity graph

The "Player Activity" graph in the bottom left shows that one player generated a lot more statements than the other player. This is the case, because one player had the procurement module where six values need to be adjusted with the slider-button combination. In all other modules combined only 7 slider-button combinations are present. Only the players 4007 and 4631 have activity numbers that are closer together.

Session Duration graph

The "Session Duration" graph is in the bottom middle. The session duration changed greatly between the sessions. The fastest session is 27 minutes and the slowest is 95 minutes long. One session took 42 minutes and the other 3 sessions took between 45 and 75 minutes.

Session Capital graph

The "Session Capital" graph in the bottom right shows how the capital of all sessions changed over the periods. All sessions increased their starting capital over the course of the game. The end capital can be grouped in three clusters. Session 4898 had the lowest capital, 5305, 1389 and 2003 are in the middle, and 358 and 3748 are at the highest capital. Furthermore, all but one session had a sharp reduction of the capital in period 6 or 5. This happened, because in these rounds machines were bought to fill the demand. The only session to that did not drop in the capital was 2003, which did not pre produce for

period 7. The difference in the capital drop in round 1 results from different amounts of products bought for storage.

7.2.2. Function

Subsection overview

The next part of the analysis will explore how and whether the learning goals that were defined in chapter 4.1. can be evaluated using the graphics of the function tab of the dashboard. This part will evaluate the use of the functionality by the players. The main result from this analysis is that most of the functionalities were used by the sessions, only the bulk discount was not used by some sessions. Some of the functionalities are analyzed in later chapters.

Bulk Discount graph

The "Bulk Discount" graph can be seen in the graphic in the strategy tab on the top right. Here it can be seen whether the sessions used the bulk discount and to which extent. Session 358 did not use any bulk discount when buying parts except the bulk discount thresholds that were hit by the production. Due to this the actual price bar is closer to max price bar than the min price bar. Session 2003 and 5305 also did not use the bulk discount. Session 1389 used the bulk discount in the later periods. Session 3748 tried to use as much bulk discount as possible each period. In some periods they bought all parts using the bulk discount. The same is true for session 4898 even though they used a little less bulk discount. When buying parts with the bulk discount the players will need to store these excess parts for the next period. This can be seen if in the rate of car production graphic the distribution of area size is not equal between all areas. This is the case for Session 358, 1389, and 5305.

Cars for Storage graph

The "Cars for Storage" graph can be seen in the top left panel in the function tab. It can be seen that all sessions used this functionality. For every period excluding the first and last at least one session stored cars. Storing cars in the last period would reduce the capital of the session, as the storage does not count towards the capital. Storing cars in the first period is risky, as the demand is 50, which is one full production cycle, and the demand in the following period is only 75. Thus using another full production cycle with producing another 50 cars in period 1 would result in 75 cars that cannot be sold during the first 2 periods with full production capacity. If a session stored cars in the second period the amount was always 25. This is the case, because the demand rises from 75 to 125 from period 2 to 3. Thus producing 100 cars in period 2 and 3 enables the players to completely fulfill the demand. In period 6 most cars were stored by some sessions in order to fulfill the high demand of period 7. Some sessions stored 75 cars, but others none. All in all the functionality to store cars was used by all sessions.

Loan Taken graph

The "Loan Taken" graph can be seen in the middle panel of the top row in the function tab. The game configuration is in such a way that the players need to take a loan in period 1 to be able to buy machines and produce cars. The size of the loan differs in relation to the amount of parts bought with bulk discount. All sessions but one were able to pay back the loan fully at the end of the game. Only two sessions were able to pay back the loan in the middle periods and then take a loan for the last periods again. All sessions used the loan and the individual installments. In the setting of the game the players could choose how much of the loan they want to pay back each session. Since the interest rate is added to the loan each period it is best to use all cash to pay back the loan at the end of the period. If not enough cash is available in the next round a loan can be taken again. Only session 2003 and 3748 did this consistently through the game. The other sessions had at least one period where the installment was not maximal, which can be seen by the red circle on top of a visual of the dark blue circle. Thus even though the functionality was used by all sessions only two noticed, that all cash should be used to repay the loan.

Personal and Machine Management

The "Personal and Machine Management" table can be seen in the top right panel. The easiest part to check whether the players understand this panel is to check the change from period 9 to 10. Since for firing the employees a severance pay must be paid that is greater as the loan it is not rational to fire employees in period 10. Opposite to that machines that are not used in period 10 should be sold, because then the maintenance does not need to be paid. Session 1389 fired personal and session 3748 did not sell the machines. The other sessions noticed this implication. Another part of the personal and machine management is that for each machine one personal is needed. Thus assembler plus inspector should equal the number of machines. The ratio of employees of 3 assembler to 1 inspector is recommended as they are needed to produce a car from the parts within one period, but not necessary as intermediates can be stored. Thus it is only analyzed whether each machine had a worker. This is the case for all sessions, so everybody used this functionality correct in this regard. Having more employees than machines is not as expensive and might be part of a team's strategy. This was the case for period 2 and 3 in session 1389, in period 8 and 9 for session 2003, in period 9 in session 4898, in period 5 and 6 for 5305. Thus only session 358 has no inconsistency in the personal and machine management. It is interesting to note that session 2003 did not have any employees or machines in period 5 and 6.

Composition of Car Production graphs

The "Composition of Car Production" graphs can be seen in the bottom panel. Here it is shown how the different elements needed to produce cars are in relation to each other. If the areas of the different elements have the same size than the production of cars is balanced. This is the case for session 358 and 2003. All other sessions have fluctuations in the ratio of the elements to each other that do not result from changes in the demand. This can be explained by the usage of the bulk discount from these sessions, because in order to reach a threshold for the discount more parts are bought, which also increases the amount of cars that can be produced with this material, thus increasing this area in the graph. In the graphs for the other sessions that actively used the bulk discount there are no extreme shifts in area sizes that show a mismatch of resources. The difference in the ratio in period 5 with an over representation of the personal and machines is caused by the low demand of 20 in this period. Session 2003 has no area in period 5 and 6, because their storage was empty and they did not have any personal or machines. All in all the production function was used by all sessions without obvious errors.

Interesting finding

Two interesting occurrences are period 3 of session 1389 and periods 4 and 5 of session 4898. In period 3 of 1389 almost no parts were available, only gears and engines. This is the case, because they pre-produced chassis and bodywork in period 2. Session 4898 did not have any production in period 4 and since they did not buy parts and had an empty storage the graph only shows personal and machines for this period. In period 5 they restocked their storage with usage of bulk discount thus the gear takes the greatest area in the graph.

Achievement of learning goal

The learning goal to understand the functionality is mostly achieved by all sessions of the game. It is only mostly reached, because errors are made in the personal and machine management and the bulk discount is not used by all sessions. The performance on this learning goals differs for the sessions and functionalities, but all functionalities except the previously two are understood satisfactory by all sessions.

7.2.3. Strategy

Subsection overview

The analysis of the strategy will focus on one of the session as an exemplary analysis. Findings of the other sessions will also be added. The main analysis will be done on

session 5305. The players of session 5305 did two mistakes at the start of the game, but were able to learn from their mistakes in later situations.

Produced Sold Demand graph

The top left graphic "Produced Sold Demand" in the strategy tab displays the amount of produced and sold cars and the demand for cars each period by the session. This session was the only one that did not produce and sell 50 cars in the first round. The basic idea behind this was that the loan they need to take was not as big, because less parts need to be bought this way. Since the most expensive part of the car production is the machine this idea was not correct. In the later periods they did not continue with producing only a part of the cars of the demand even if the machines can produce more. In period 2 they produced and sold 50 cars and did not repeat the mistake from period 1. For period 3 they increased the production to 100 even though it was already possible to see that the demand will drop to 40 in the next period now going high risk with the production increase. In period 4 they could see that the demand decreases further to 20. So they produced only 50 cars. But they did not sell their machines even though they would not be used in the current and next period. This can be seen by a machine workload of only 50%. Selling or holding the machines in this position is a gamble as it is cheaper to hold unused machines over 2 periods in comparison to selling and buying back new machines. In period 5 when the demand is at 20 and the demand of the next period increases to 25. They sold the machines they kept on hold to reduce the loss of buying a machine and then selling it again, because they now expected that the demand stays low, but unluckily the demand raises again two periods in the future. But they overproduced cars again like in the previous period, because the lost margin for stored cars is not as big and they expected a demand raise in the future. In period 6 they could see that the demand rises to 175 cars. So they produced 50 cars again and with the stockpile of cars they produced over the low rounds were able to sell 140 cars with only increasing their production capacity to 100. In period 8 they had no stockpile of cars left, but could see that the demand of cars drops to 130 in the next period. So they did not increase their production capability like they did in period 3 and adapted their strategy by learning out of the previous mistake. So they only produced and sold 100 cars of the demand in period 8 and 9. For the last period they reduced their production capabilities to 50 to not have unused production capabilities again.

Conclusion of Produced Sold Demand

All in all they did two mistakes at the beginning, not producing fully in round 1 and buying machines in period 3. They later learned from their mistakes and tried to produce as much as possible to not waste possible car sales. In the periods with low demand they kept producing more cars to build a stockpile. In period 8 they did not repeat the mistake from period 3 and did not buy machines to fulfill the complete demand.

Capital Development graph

The "Capital Development" graph is in the top middle panel. The capital development over the periods in comparison to the mean of all sessions is that they were above the mean until period 3. Since they did not buy machines in period 2 their capital increased while the curve of the mean decreases. In period 4 the capital decreases strongly, because they did not sell their machines. In period 6 they have a higher capital as the mean, because in the other session machines were bought to pre-produce cars for round 7, which this session did not need to do due to the stockpiling of cars over the low demand periods. The rest of the periods the capital increases almost the same as the mean. The difference in the last two periods is from not buying the parts for the last period in period 9 to profit from the bulk discount.

Capital Allocation graph

The "Capital Allocation" graph in the bottom left shows that session 5305 only used the storage for the cars and did not use it for parts or intermediates. As the storage value only increases in the periods were they started to stockpile cars.

Machine Workload graph

The "Machine Workload" graph is in the bottom middle panel. The machine workload is at 100% for all periods except the first and the low demand periods which is to be expected. The low machine workload in the first period is discussed in the analysis of the "Produced Sold Demand" graph.

Time per Period graph

The "Time per Period" graph shows that they needed a long time to get used to the game in period 1, but after taking their time to understand the game in the first 3 periods, they were able to progress fast through the periods only needing longer in the periods where they needed to adjust their strategy. These were period 5 were they sold their machines and period 7 were they decided to not increase their production. All in all the players from this session learned from their mistakes and adapted their strategy accordingly, but they did not use the bulk discount.

Achievement of learning goal

In the above analysis the strategy of session 5305 is analyzed. Even though the strategy was not well defined in the first periods the players are able to adapt the strategy and handle similar situation with their changed strategy. Thus they were able to develop a strategy under uncertainty and achieved this learning goal even though they made mistakes.

Interesting findings

Interesting points in other sessions are that session 2003 had always 100% machine workload only in session 5 and 6 they had no machines and no production. Session 1389 and 4898 always fulfilled the demand. Another interesting part is that the sessions that had a focus on the usage of the bulk discount have a lower capital at the end of the game as the session that did not use the bulk discount. This could either be coincidence or the bulk discount usage distracted from developing a better strategy.

7.2.4. Interplay of Processes

Indicator

An indicator for interplay of processes is the time the players needed per session. This can be seen in the bottom right graphic in the strategy tab. In three cases the time taken per period is the highest in the first period. Session 1389 took the most time in period 2. Session 3748 took most time in period 4. Session 2003 took most time in period 5 and 8. In all sessions it can be seen that in later periods the time taken to work through the periods was less than in the earlier periods. The only exception is session 2003. This session had the lowest session duration by approximately half of the next fastest session. So the players of this session were able to get a fast grasp of the game and only took longer in the periods where they had to think about their strategy.

Achievement of learning goal

All in all it can be said that all sessions were faster with progression of the game and thus learned how to manage the company and what actions need to be done per period. Session 2003 knew from the start what they needed to do and were fast in all periods except the previously mentioned. The interplay of processes is understood by all sessions over the course of the game and the learning goal of understanding the interplay of the processes is achieved by all sessions.

7.2.5. Collaboration

Indicator

Another learning goal is the collaboration of the players. Since the collaboration did not happen through means of the game that can be logged like a chat, no direct way to measure this learning goal is in the game. Nevertheless, indicators for the collaboration are shown in the collaboration tab. This analysis is also done for session 5305 as a proxy for the other sessions. As expected it is difficult to measure the collaboration of the players, but from the analysis the players of all sessions have worked collaboratively.

User Activity per Period graph

The "User Activity per Period" graph is in the top middle panel. The user activity is higher in all periods except the second one for the red player, who has the procurement module. It is astonishing that the activity of this player is less in period two, but this is caused by buying the parts for the first two periods in the first period with usage of the bulk discount. Thus in period two almost no actions were required in the procurement module.

User Activity per 30 seconds graph

The "User Activity per 30 seconds" graph shows when each player was active. It can be seen that both players were active throughout the session with some time frames were both players did not have an activity. In these time frames the players were most likely discussing the strategy for the current period.

Next Period Question graph

In session 5305 the question for the next period was asked 11 times by player 5533 in red and 2 times by player 5483 in blue, which can be seen in the "Next Period Question" graph in the bottom left. The number of questions for the next period does not equal the number of periods, because the questions for the next period can also be rejected. Even though player 5533 asked the question more often the collaboration cannot be seen from this graph.

Changes of Values graph

The "Changes of Values" graph in the bottom right shows when the values were changed throughout the game. The y-axis shows the percentage of the max value during the game the value has. The different categories in which values can be added in the game are also distinguishable by color. The blue hues are for the parts. The blue-greenish hues for the personal and machines, the green yellow hues for the loan, the yellow to orange hues for the production, and the red hues for final assembly and sale. This makes it easy to see in which order the modules were worked on. The end of period can be seen by an overlapping of many lines that go back to 0. In the change of values graph for session 5305 this happens the first time at minute 17 when the first period is finished.

Order of module editing

At first the order in which session 5305 worked on the modules will be analyzed. In general the order the values were entered is procurement, then production, followed by sale and lastly loan. In the first two periods they had huge problems to set the installment to the desired value. In period 1 they decided to not use an installment at all after trying to enter the value for about 2 minutes. In period 2 it took them about 2 and a half minutes to enter the desired installment. In later periods they only needed 10 to 30 seconds to enter this value. And in period 6 to 8 they set the installment to almost the perfect value as they had almost no cash remaining at the end of these periods. In periods were they stored cars the first value they changed was the amount of cars they want to sell. At about minute 39 and 46.5 and 49.5. Interestingly though at minute 39 they reduced the value again to 0. The personal and machine values were changed after buying the parts and before the production in the periods where they increased these values and in periods they decreased these values after the sale and before adjusting the loan.

Corrections of values

This paragraph describes at which times they corrected values at a later point. In the graph the shaking of the lines a few seconds before they settle on the desired value is caused by the way the values are entered with a slider and buttons. Since it is not easy to set the slider to the desired value the value is approached with the slider and then danced around the desired value until finally the slider is close enough to the value so that the buttons can be used to correct the remaining offset.

In the first period they first set all values of the procurement to a value. A little later they decided to reduce this value by about half before a 5 minute pause of interaction with the game. The pause of 5 minutes in the first period was used to discuss how many cars they want to produce, whether to stay with the entered values or to use values they corrected to. As a reminder session 5305 only had 60% workload on their machines in period 1.

And in this pause they confirmed this decision. This can be deducted because after the pause they did not change the amount of parts that they bought. Even though adjusting the production and final assembly took some time afterwards. In period 3 at minute 26 they first set all values and after 2 minutes into this period they corrected these values by one production cycle up. In period 5 at minute 43 they also changed their values as well as in period 7 at minute 52. The increase in period 7 was taken back by the players after about 30 seconds. This leads to the assumption that they wanted to test what the finance looked like without calculating it by themselves. All in all the values were changed in period 1, 3, 5, and 7 by the players and problems with setting the installment slider to the correct value became evident. At the start of each new period they made a pause of about 30 seconds to look at the new demand and discuss it.

Achievement of learning goal

As it is seen above the collaboration of the players cannot be evaluated directly with the collected data. The data that are collected leads to the evaluation that the players of session 5305 did play together. While playing the game they had pauses that are likely used to discuss how they want to produce the cars. Furthermore, they corrected their values which leads to them noticing or testing another configuration. Since they split the modules in a way that not one player can do everything alone they needed to do these adaptions together. The user activity and next period questions also do not indicate missing collaboration. Thus the learning goal of using collaboration to play the game is achieved by session 5305.

Interesting findings

It is interesting to note that the question for the next period was asked mostly by one player through all sessions. It seems that the basic idea, that if one player directs how the game should be played also is the one who initiates the next period, is not correct as in all sessions was collaboration between the players. It is a routine that the same player asked the question to the next period each time and thus this is not well suited as an indicator for whether somebody leads. The module selection between the players was balanced in all sessions. In the two sessions where one player choose 5 modules and the other only 3 the player with the three modules had the procurement module which has the most input values. Session 1389 had a 10 minute interval at the start of period 2 in which they did not interact with the game. This is the case, because during this time they created an excel sheet to help them with the calculation.

7.2.6. Graphic Test

Graphic Test

The graphic test tab is an evaluation of the graphical representation of the "Changes of Values" graph. This tab is not intended to be analyzed as the information is redundant to the bottom right graphic of the collaboration tab. This tab is only for the dashboard user study to make it possible for the participants to compare the two different ways to present the data. The 3 graphics in the top row show the same data as the graphic in the bottom row. The difference is that the y-axis displays the absolute values and that the different lines are split based on their value range into high, med, and low. Furthermore, these 3 graphics are only half as wide as the previous graphic to fit into one row.

7.3. Comparison to Self-Assessment

Section overview

This section compares the result of the analysis of the achievement of the learning goals in chapter 7.2. with the self-assessment of the players. The questionnaire that is used is in appendix B.3. The answers of the participants can be seen in figure 7.3. As in the previous section the function usage will be analyzed for all sessions, but the strategy and collaboration only for session 5305. A positive self-assessment response is if the value is

	Session 358		Session 1389		Session 2003		Session 3748		Session 4898		Session 5305	
	P. 647	P. 862	P. 1437	P. 1886	P. 2772	P. 2949	P. 3174	P. 3388	P. 4007	P. 4631	P. 5483	P. 5533
1	4	3	3	3	4	2	4	4	4	4	3	4
2	**4**	**4**	**3**	**3**	**2**	**3**	**3**	**4**	**3**	**3**	**3**	**4**
3	**3**	**4**	**4**	**4**	**4**	**4**	**4**	**4**	**2**	**3**	**3**	**3**
4	**3**	**4**	**2**	**4**	**4**	**4**	**3**	**3**	**3**	**4**	**4**	**2**
5	**4**	**4**	**3**	**3**	**1**	**2**	**3**	**4**	**4**	**3**	**4**	**3**
6	**4**	**4**	**4**	**3**	**3**	**4**	**3**	**4**	**4**	**3**	**4**	**3**
7	3	4	3	3	4	2	2	3	4	3	2	3
8	2	4	1	2	2	3	3	3	4	3	2	3
9	3	3	3	2	4	4	3	4	4	3	4	4
10	3	3	2	3	3	3	3	3	3	3	3	1
11	3	4	3	3	4	4	4	4	4	4	3	3
12	4	4	4	3	4	4	4	4	4	4	3	4
13	2	2	4	3	2	1	2	2	4	4	4	4
14	3	3	4	4	4	2	3	3	3	2	3	3
15	3	2	1	2	3	3	2	2	2	3	3	4

Figure 7.3.: Question Overview

either a 4 or 3. A negative self-assessment response is a 2 or 1. In addition to this in cases where the answer to the same question of participants that played a session together is 2 or more apart from each other the answers will be highlighted in red in figure 7.3.

Functionality

In the analysis of the dashboard the result is that all sessions were able to use all of the functionalities. In the questionnaire questions 2-6 ask the player how they used the functionality. The values of these questions are bold. Most teams answered with a positive response to this question. In total 6 answers to these questions were negative. These 6 answers are marked with a grey background in figure 7.3. Three of them are on the loan management. Two from the same session for production for storage and one for procurement. The procurement for session 2003 is evaluated negatively by one player, because they did not use the bulk discount to store the parts. This is probably also the case for the two negative responses to question 5 by the same session, as they also did not store cars in a lot of sessions. But since they used both the car storage functionality and the bulk discount, they choose this evaluation, because they felt they could have used the functionality better. As previously mentioned the loan management was not done perfectly by any session. Thus it is expected that some players felt that their loan usage was not good enough for a positive response. This is the case for a player from session 1389, 4898, and 5305. For session 1389 and 5305 the installment was evaluated negatively. This is in accordance to the data from the dashboard, which says that in these sessions in at least one period the installment was not maximal. For session 4898 the taking of the loan is negatively evaluated. In the dashboard they were not able to repay the cash over the course of the game, thus the negative evaluation can also be founded on the dashboard. All in all it is surprising that not more players gave a negative response to the questions concerning the loan. This is probably the case, because they did not notice that the installment rate can be set maximal.

All in all the self-assessment of the functionalities can be considered like the results that were deducted in the analysis from the dashboard, as 3 of the 6 negative answers were on the loan management, which was not used perfectly by any session.

Strategy

Now the self-assessment of the strategy is analyzed for session 5305 and compared to the dashboard. The adaption of the production to the demand as well as the capital usage is evaluated as positive by one player and negative by the other player. In the dashboard it can be seen that the production could have been better in the last 3 periods and the capital usage is also good over the course of the game. Only the loan could have been repaid better. The machine workload could have been better, since the machines were not fully occupied in period one. Even though both players gave the value 4 as answer

to machine workload. The capital development is evaluated with a value of 3 and 1 for this session. Since the capital the session had at the end was about the same as the mean the evaluation with 1 seems a bit too criticizing, but it is not possible for the players to compare their result with other results.

All in all the self-assessment of the strategy is reflected mostly in the dashboard analysis, except the workload which is not perfect by the dashboard analysis. Here the players could be explained why there is a difference in the assessment to gain a learning effect.

Collaboration

The collaboration by the players was evaluated with a 3 by one player for questions 11 and 12 and with 4 by the other player. In the analysis of the dashboard it is not possible to see that the collaboration between the players was weak. Thus this self-assessment seems to be in accordance to the dashboard analysis.

Diverging answers

In the results of the questionnaire were 7 instances where the players answered the questions divergent, which is the case if their answer has a difference of 2 or more. This is the case 3 once for session 358 and 1389, twice for session 5305 and thrice for session 2003. In session 358 the players disagreed with the capital usage. Session 1389 disagreed with the loan repayment the player that had the loan module answered negatively, while the other player said they repaid the loan perfectly. It seems that the player without the module did not notice how the loan was repaid. In session 2003 question 1, 7 and 14 were answered with a difference. It seems that the player that answered the questions more negative, because it was not easy to understand the game. This can be seen by the negative answer to question 1, which asks how well the production cycle was understood. The other differences in the answers are probably caused by this. Session 5305 had a difference in the question about the taken loan. One player answered this question with 4, while the player without the module answered this question with 2. In the dashboard no negative usage of the loan for this session can be seen. The other difference is the capital development where one player answered 1 and the other 3. Since the players had no way to compare to other sessions this discrepancy can appear. There was no instance were one player answered with 1 while the other player answered the same question with 4.

Additional questions

Now the last 3 questions will be analyzed. The first question asked whether the activity between the players was balanced. Half of the players gave a positive answer and the other half a negative. In the dashboard the data shows that the activity with the game was not balanced in a single session. In the case where the players gave a positive answer it can be caused by a perceived balanced workload as both players discussed the strategy and played the game together. Question 14 asks whether the modules were selected sensible. This question had only 2 negative answers. The data in the dashboard shows that the modules were split either 4:4 or 5:3, which both can be considered a good division of the modules. In not a single case one player could play the game alone, thus the result of the self-assessment and the dashboard align. The last question asked how often corrections of the values needed to be made. Half of the players gave a positive and the other half a negative review.

Self-assessment result

All in all most of the answers from the self-assessment can also be seen from the dashboard analysis. Some differences in the answers are present, but most of them can be explained by the lacking of a comparison of the players with other runs and a perfect run.

Chapter 8 Dashboard Usability Study

Chapter overview

This chapter first describes how the study of the dashboard was created. Then the results of the study will be analyzed.

8.1. Study

Usability study

In this section the dashboard usability study that was conducted will be explained. The study will use the evaluation schema by Brooke [27]. It consists of 10 questions that the participants are asked after they have worked with a system. The questionnaire can be found in appendix B.4. Each question has a Likert Scale to give an answer on. The range of the Likert Scale is from Strongly Disagree to Strongly Agree with 5 steps where each answer has a value of 0 to 4 for the usability of the system. All values of the answers are then summed and multiplied by 2.5 to get the usability score between 0 and 100. The usability score of the system is the arithmetic middle of all participants. Higher values indicate better usability. In this master thesis the system to be evaluated is the dashboard.

Questions for dashboard usability study

In parallel to the questionnaire of the dashboard usability study the participants are given 14 questions and talk aloud how they go on answering the questions. Notes are taken if participants had problems to answer the questions. It is important to note that the questions focus on the usability and not on the interpretation of the dashboard. The questions are:

1. General: Which was the most used activity in the "Activity Usage" graph and how often was it used?
2. General: What does the y-axis of the "Session Capital" graph show?
3. General: Why are the players ordered this way in the "Player Activity" graph?
4. Function: Why are the colors in the "Personal and Machine Management" table different?
5. Function: What does an area in the "Composition of Car Production" graph show?
6. Function: What does each different colored bubble represent in the "Loan Taken" graph with color and size?
7. Strategy: How do you change the current session to look at?
8. Strategy: Did session 3748 finish above or below the mean capital?
9. Strategy: Show a session and period where more cars where produced than sold.
10. Collaboration: Which player had the procurement module?

11. Collaboration: What does the x-axis in the "User Activity per 30 seconds" show?
12. Collaboration: What does the y-axis in the "Changes of Values" graph show?
13. Graphic Test: Which type to represent the "Changes of Values" graphic do you like more?
14. Overall: Do you have any remarks?

Reason for question selection
Participant selection

These questions evaluate the understandability of each graph type, as well as the understandability of the navigation through the sessions with the filter and menu. For each graph type at least one question was asked. To guarantee that the participants have a general knowledge of the data that is displayed in the dashboard, the participants of the usability study have all participated in the study to collect learning data or played the game in a test run.

8.2. Analysis

Usability questions

At first the answers to the questions that were asked during the dashboard study are analyzed. The first question checks whether the players notice by themselves that the graphics can be hovered. Four players did not find the hover functionality. Thus it seems to be necessary to inform them in advance of the hover functionality. The second question was answered correctly by all players. For the third question only two players were not able to answer that the different coloring is based on player one and player two of a session. This is probably the case, because all participants played *transaction MTLG*. For the fourth question only one participant was not able to notice that the cell background color opacity is based on the value in the cell. Only 4 participants were able to answer question 5 correctly. It seems that the "Composition of Car Production" graph needs to be explained to the players. After explaining this graph, all participants were able to see the added value of the graph. Two players were not able to correctly answer question 6. All players were able to answer question 7, 8, and 9 correctly. For these questions they had to explore the dashboard by themselves as the graphic that they needed to look at was not mentioned in the question. Question 10 is also answered correct by all players, as well as question 11, which is similar to question 2 that also was answered correctly by everyone. Question 12 is only answered correctly by 6 participants, thus an explanation of the graphics y-axis seems to be necessary.

Graphic test question

For question 13 about which way to display the "Changes of Values" graph 8 participants preferred displaying the values with the three individual graphs. Only 2 participants said that the single graph can be better for experts to deduct information from it. The reasons that the participants preferred the top graphs were that it is easier to follow the lines and that the color can be distinguished better. On the first glance it seems not to be possible to the advantages of the graphic that combines all lines, as the graphic looks to intimidating. During the discussion with the participants about the graphic it was discovered that the bottom graphic can be enhanced with interactivity. If it is possible to highlight specific lines, that can be selected either by hovering or multiple with a selection over the legend. This way the benefits for which the participants chose the top 3 graphs over the bottom graph can be added to the bottom graph. From an analysis perspective the lower graph has the advantages that the full height of the graph is used by all lines, the graph can be wider, and all lines are in one graph and can be compared to each other.

Participants remarks

Values	4	3	2	1	0

Scores:	0-64: Not Acceptable
	65-84: Acceptable
	85-100: Excellent

	Ev. 1	Ev. 2	Ev. 3	Ev. 4	Ev. 5	Ev. 6	Ev. 7	Ev. 8	Ev. 9	Ev. 10
Q 1	4	3	3	4	2	3	4	2	3	3
Q 2	4	2	3	2	3	3	3	3	3	4
Q 3	4	4	3	3	3	4	4	3	4	3
Q 4	4	4	3	4	4	4	4	4	4	4
Q 5	4	3	4	1	4	3	4	2	4	3
Q 6	4	4	4	4	4	4	4	4	4	3
Q 7	4	3	3	4	4	3	3	3	3	4
Q 8	4	2	3	4	4	4	4	3	4	4
Q 9	4	2	3	4	3	4	4	2	4	3
Q 10	4	4	2	2	3	3	4	2	4	4
Scores	100	77,5	77,5	80	85	87,5	95	70	92,5	87,5
Average	85,25									

Figure 8.1.: Survey Overview

Over the course of the study the participants commented on some possible minor improvements of the dashboard. Two participants commented on the personal and machine management table. First that "Session" in the top left corner is misleading as the periods can be interpreted as the sessions this way and that the order of the sessions is reversed compared to the other graphics in the same row. Then some comments on alternative naming of axis or graphics were made. These suggestions were incorporated to the dashboard after the study was conducted.

Usability study procedure

The answers on the System Usability Scale by Brooke [27] are analyzed analog to the analysis by McLellan et al. [28]. The scoring instrument they used will be adapted to work on the value interpretation of the answers as well as using a different color for each possible value. The evaluation can be seen in figure 8.1.

Usability study analysis

Figure 8.1. shows how each participant of the study answered each question. Each participants has its own column and each question its own row. The value 4 is the best possible answer, while a lower value represents a worse answer to the question. The score is the sum of the values of the participant multiplied by 2.5, which represents the system usability score the participant gave the dashboard. The average of all players is shown at the bottom. The most values in the table are positive responses to the question, either 4 or 3. The 2 in the table is the neither agree nor disagree, which half of the participants gave as answer to at least one question. Only one negative response was given. This was for the question 5 by participant 4. This question deals with the integration of the functions in the dashboard. The main critic of this user was that no notice informs the user on the hover functionality. The overall score the users gave the dashboard is either in the acceptable category (4 results) or the excellent category (6 results). The lowest score is 70 and the highest is 100. The average of all scores is 85.25, which is just above the threshold for an excellent score.

Question analysis

Next the average Score for each question will be plotted like McLellan et al. [28] presented it in their study. The y-axis will range from 0 to 4 to represent the values of each answer plus the 95% confidence interval by the error bars. This can be seen in figure 8.2.

The mean of all questions is at or above 3. The first two questions have the lowest average. These questions ask whether the participants would use the system frequently (question 1) and whether the system was unnecessarily complex (question 2). Question 4 and 6 have a mean of 3.9, which is almost at maximum. This shows that they were able to understand the dashboard without help of a technical person as well as not seeing inconsistency in the dashboard. Question 5 has the highest confidence interval, because of the

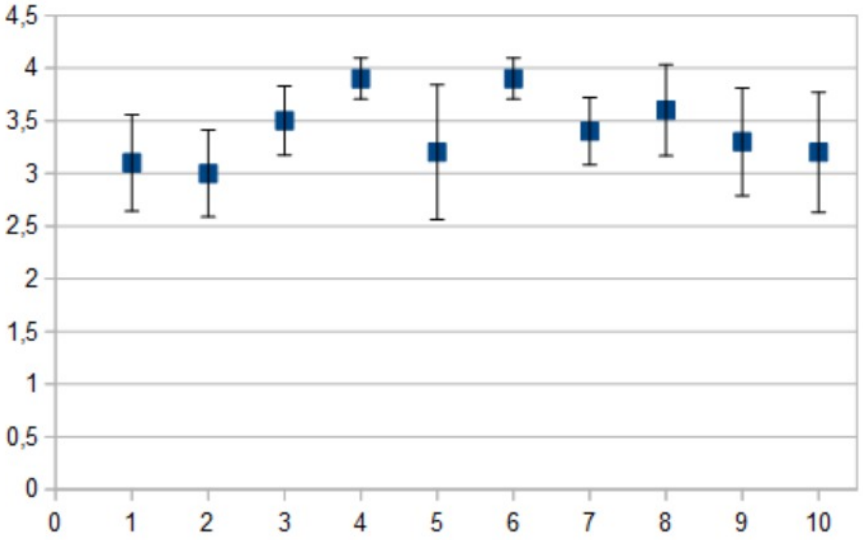


Figure 8.2.: Survey Mean

Mistake in answers

value 1 in the evaluation. Overall the system received a positive evaluation where even the lower ends of the confidence interval are all above 2.5, signaling that the dashboard will very likely get a positive evaluation when more participants are added to the study. After starting the analysis of the collected questionnaires, the answers of three participants seemed likely to contain errors that were made while answering the questions. The first is participants 1 that either choose strongly disagree and strongly agree and gave a perfect score. The second was participant 3 that answered question 9 and 10 with value 1. The third is participant 4 that answered question 4 with a value of 1. After contacting the participants of the study again it was discovered that participant 3 did indeed understand the questions 9 and 10 wrongly and corrected the answers to 3 and 2. The other two participants confirmed the answers that they had given. This is in accordance with the findings of Sauro et al. [29], who said that 13% of the System Usability Study questionnaires they analyzed were answered with mistakes. In the SUS study of this master thesis 10% of the questionnaires were initially answered with mistakes. The values presented in the analysis above are with the corrected values. Without the correction of the mistake the overall score is 84.5, which would mean the dashboard has an overall score of acceptable instead of excellence, but either way the score is just between these two results. The mean for question 9 and 10 would also be slightly lower.

Part IV

Result

Chapter 9 Conclusion

Overview

The goal of this master thesis is enabling Learning Analytics for the game *transaction MTLG* with the usage of xAPI. To achieve this goal *transaction MTLG* is upgraded to enable the collection of learning data with xAPI in a Learning Record Store. A Learning Analytics dashboard is created based on defined learning goals to display the learning data collected from the LRS in different graphics. These graphics can be used to analyze the achievement of the learning goals for the players. To evaluate the dashboard a user study is conducted to generate the learning data that is displayed in the dashboard. And finally a usability study of the dashboard is conducted.

Extension of transaction MTLG

transaction MTLG is extended in various ways to enable the learning data collection. At first an online multiplayer is added to be able to conduct a user study during COVID-19, as *transaction MTLG* is originally played collaboratively on one device. During testing bugs are detected in the base game that are fixed to guarantee a stable game execution. Fixing these bugs was challenging, as they are diverse and program changes needed to be done on a structural level. The known bugs of the base game are worked around as fixing them would require changing the concepts for these functionalities.

User study

The learning data is collected with a user study on the upgraded game in xAPI format and sent to Learning Locker for storing. For this predefined xAPI definitions are used and the xAPI data collection is added to *transaction MTLG*. In order to know which data to collect learning goals of the game are set. The learning goals are defined based on the original game *transaction*. The configuration of the game is tuned so that the players are challenged with situations that show whether they reached the learning goals. After they played the game the players answered a self-assessment about their game performance. It was difficult to find enough players to take part in the study, because the study took 1 to 2 hours and two participants are needed at the same time.

Learning Analytics dashboard evaluation

The Learning Analytics dashboard is created based on the learning goals. The learning data is fetched from Learning Locker and transformed into graphics. These graphics are used to apply Learning Analytics to the data and evaluate whether the players reached the learning goals. The focus of the Learning Analytics is one specific game run. The self-assessment of the players is also compared to the analysis via the dashboard. The analysis of the learning data shows that it is possible to deduct a lot of information about the players that can be used to evaluate their achievement of learning goals. Even the collaboration between the players can be analyzed even though the collaboration happened not in the game, but through a voice channel.

Dashboard usability study

After showing an analysis with the dashboard, the dashboard is evaluated with a usability study. In this study the participants were asked questions that they answered with using the dashboard. Some participants had problems understanding some of the graphics in the dashboard. After the graphics were explained to them they were able to understand them. Recommendations of the participants are added to the dashboard.

Achievement of goal

After they finished exploring the dashboard, they rated the dashboard on the System Usability Scale. The result of this evaluation is a score of 85.25, which is just above the threshold for an excellent system. Thus the dashboard has a high usability.
All in all the goal to enable Learning Analytics with xAPI for the game *transaction MTLG* is reached within this master thesis and it is shown with a user study, that meaningful information about the players can be deducted in the analysis. Furthermore, the usability study of the dashboard shows that the usability and understandability of the dashboard is high.

Chapter 10 Future Work

This chapter collects ideas of future work based on this master thesis. The topics of future work are ordered from base game to dashboard.

Enhancement of transaction MTLG

One basic element future work is the enhancement of the base game. The two bugs moving the modules and reloading a session are still in the base game and were not fixed. In the bug fixing section of this thesis the workaround for these bugs is explained. Thus one essential future work is fixing these two bugs. While fixing these bugs it is also recommended that the control flow of the program is refactored to make it easier and faster to understand the program. For this the execution should not jump from one file to another with different emitters all over the place, but the control flow should be more centralized in one file. In addition a refactoring of the game code can be helpful for better readability.

Additional logging

Another part is that in addition to the logged events in the game it could also be logged with xAPI when and how often the players open the manual as well as the position of the selected modules in addition to the order in which they were selected.

Message handling

To reduce the amount of messages sent to the server the sliders only send their data if they were not moved for a small time frame. For the buttons this functionality is not added, so if a button is mashed really fast a lot of xAPI statements are sent to the server and can overpower the server, which causes slight delays in the game.

Slider optimization

In the analysis of the game it was noted that it was difficult for the players to enter the exact values. Thus another topic for future work is how the input of the values can be improved.

xAPI-statement analysis

The analysis of the xAPI statements was challenging, because it is not possible to select and transform the data with queries that are SQL-like. Thus the data needs to be collected from all xAPI-statements by looping through them manually. It can be interesting to develop a SQL-like approach to work on xAPI-statements to make it easier and faster to access the data.

Metrics for learning goals

The goal of this master thesis was to develop a dashboard that shows graphics that can be used as an indicator to answer how good the learning goals were achieved. The dashboard can be enhanced by building metrics and scores out of the learning data to evaluate the learning goals in a score. This makes the comparison between different sessions easier.

Module handling

In the game the modules were chosen by the players at the start of the game. Even if it is possible for the players to change the modules by themselves, it can still be interesting to see what happens if the modules are completely swapped each period. Especially how this will impact the collaboration between the players. Another possibility is to give preset modules for each player in order to balance the activity between the players. The player that has the activity heavy procurement module also gets the storage, finance and personal and machine management and the other player the rest.

Collaboration analysis

The collaboration of the players is analyzed with the data that is collected from *transaction MTLG*. However the data of the voice communication between the players could not be analyzed as it is not recorded. It can be interesting to analyze these data as well to see connections between the game interactions and voice communication. Furthermore, the contents of the conversation can be analyzed to get further insight into the challenges the players face while playing the game.

Mode comparison

The study conducted to collect the learning data was executed with the mode of greyed-out modules. It can be informative to compare runs between the greyed-out mode and the no-display mode to find out which way the collaboration between the players is better. Is it better when they can see the values of the other modules or is it better when they cannot see the modules of the other players?

Dashboard interactivity

In the dashboard interactivity can be added to make it possible to explore the graphics. For example zooming or highlighting can be added. This seems to be especially promising for the "Changes of Values" graphic as it is a finding of the dashboard study.

Gaming hypothesis

Another interesting hypothesis is to verify whether the participants that already played the game or play strategy computer games have a better performance in the game. Or maybe an analysis based on other parts of the background of the player. A correlation analysis between the background of the players and the game performance can be interesting to see what prerequisite knowledge help successfully playing this game.

Change target group

In this master thesis the target group for the studies was not fixed. Thus it could be interesting to let business administration students, which are the target group of *transaction*, play the game. This can result in additional feedback from a business perspective as well as runs by players that have more background knowledge. This can result in faster play times as well as a better usage of the different functionalities.

Color palette

Another future work is to allow the user to change the color palette, which is used to draw the graphics to have a better integration of users that have problems to distinguish colors.

Appendix A Bibliography

[1] D. Schablowsky, "Kollaborative planspiele für große multitouch-displays," 2019. Master-Thesis, RWTH Aachen University, The Learning Technologies Research Group.

[2] C. S. Loh, Y. Sheng, and D. Ifenthaler, "Serious games analytics: Theoretical framework," in *Serious games analytics*, pp. 3–29, Springer, 2015.

[3] J. Alvarez, D. Djaouti, *et al.*, "An introduction to serious game definitions and concepts," *Serious Games & Simulation for Risks Management*, vol. 11, no. 1, pp. 11–15, 2011.

[4] G. Siemens and P. Long, "Penetrating the fog: Analytics in learning and education.," *EDUCAUSE review*, vol. 46, no. 5, p. 30, 2011.

[5] D. Ifenthaler, "Learning analytics im hochschulkontext–potenziale aus sicht von stakeholdern, datenschutz und handlungsempfehlungen," in *Digitale Bildung und Künstliche Intelligenz in Deutschland*, pp. 519–535, Springer, 2020.

[6] K. Verbert, N. Manouselis, H. Drachsler, and E. Duval, "Dataset-driven research to support learning and knowledge analytics.," *Educational Technology & Society*, vol. 15, no. 3, pp. 133–148, 2012.

[7] M. A. Chatti, V. Lukarov, H. Thüs, A. Muslim, A. M. F. Yousef, U. Wahid, C. Greven, A. Chakrabarti, and U. Schroeder, "Learning analytics: Challenges and future research directions," *eleed*, vol. 10, no. 1, 2014.

[8] T. Elias, "Learning analytics," *Learning*, pp. 1–22, 2011.

[9] M. Freire, Á. Serrano-Laguna, B. Manero, I. Martínez-Ortiz, P. Moreno-Ger, and B. Fernández-Manjón, "Game learning analytics: learning analytics for serious games," in *Learning, design, and technology*, pp. 1–29, Springer Nature Switzerland AG, 2016.

[10] C. Alonso-Fernandez, A. Calvo-Morata, M. Freire, I. Martinez-Ortiz, and B. Fernández-Manjón, "Data science meets standardized game learning analytics," 02 2021.

[11] Advanced Distributed Learning Initiative, "Experience api (xapi) standard." `https://adlnet.gov/projects/xapi/`. Accessed: 2021-05-27.

[12] M. Ehlenz, B. Heinemann, T. Leonhardt, R. Röpke, V. Lukarov, and U. Schroeder, "Eine forschungspraktische perspektive auf xapi-registries," *DELFI 2020–Die 18. Fachtagung Bildungstechnologien der Gesellschaft für Informatik eV*, 2020.

[13] Advanced Distributed Learning Initiative, "xapi-spec." https://github.com/adlnet/xAPI-Spec. Accessed: 2021-05-27.

[14] Learningpool, "Learing locker." https://learningpool.com/solutions/learning-record-store-learning-locker-draft/learning-locker-community-overview/. Accessed: 2021-05-27.

[15] W. Barodte, "transaction information." https://www.controlling.rwth-aachen.de/cms/Controlling/Studium/eLearning/~jtfn/transAction/. Accessed: 2021-05-27.

[16] W. Barodte and M. Gerards, "transaction game." https://transaction.medien.rwth-aachen.de/ebwlp/. Accessed: 2021-05-27.

[17] Learning Technologies Research Group, RWTH Aachen University, "Mtlg framework." https://mtlg-framework.gitlab.io/. Accessed: 2021-05-27.

[18] Learning Technologies Research Group, RWTH Aachen University, "xapi definitions." https://gitlab.com/learntech-rwth/xapi. Accessed: 2021-05-27.

[19] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," in *The craft of information visualization*, pp. 364–371, Elsevier, 2003.

[20] M. Bostock, "Observable d3 gallery." https://observablehq.com/@d3/gallery. Accessed: 2021-05-27.

[21] Y. Holtz, "d3-graph-gallery bubble scatter plot." https://www.d3-graph-gallery.com/graph/bubble_basic.html. Accessed: 2021-05-27.

[22] M. Bostock, "Observable d3 bar chart." https://observablehq.com/@d3/bar-chart. Accessed: 2021-05-27.

[23] M. Bostock, "Observable d3 grouped bar chart." https://observablehq.com/@d3/grouped-bar-chart. Accessed: 2021-05-27.

[24] M. Bostock, "Observable d3 multiline graph." https://observablehq.com/@d3/multi-line-chart. Accessed: 2021-05-27.

[25] M. Bostock, "Observable d3 normalized stacked area chart plot." https://observablehq.com/@d3/normalized-stacked-area-chart. Accessed: 2021-05-27.

[26] I. Maboud, "Simple express server." https://github.com/ipenywis/simple-express-server. Accessed: 2021-05-27.

[27] J. Brooke, "Sus: A 'quick and dirty' usability scale," in *Usability evaluation in industry*, pp. 189–195, Taylor & Francis, 1996.

[28] S. McLellan, A. Muddimer, and S. C. Peres, "The effect of experience on system usability scale ratings," *Journal of usability studies*, vol. 7, no. 2, pp. 56–67, 2012.

[29] J. Sauro and J. R. Lewis, "When designing usability questionnaires, does it hurt to be positive?," in *Proceedings of the SIGCHI conference on human factors in computing systems*, pp. 2215–2224, 2011.

Appendix B User Studies

B.1. Declaration of Consent

Einwilligungserklärung gemäß DSGVO

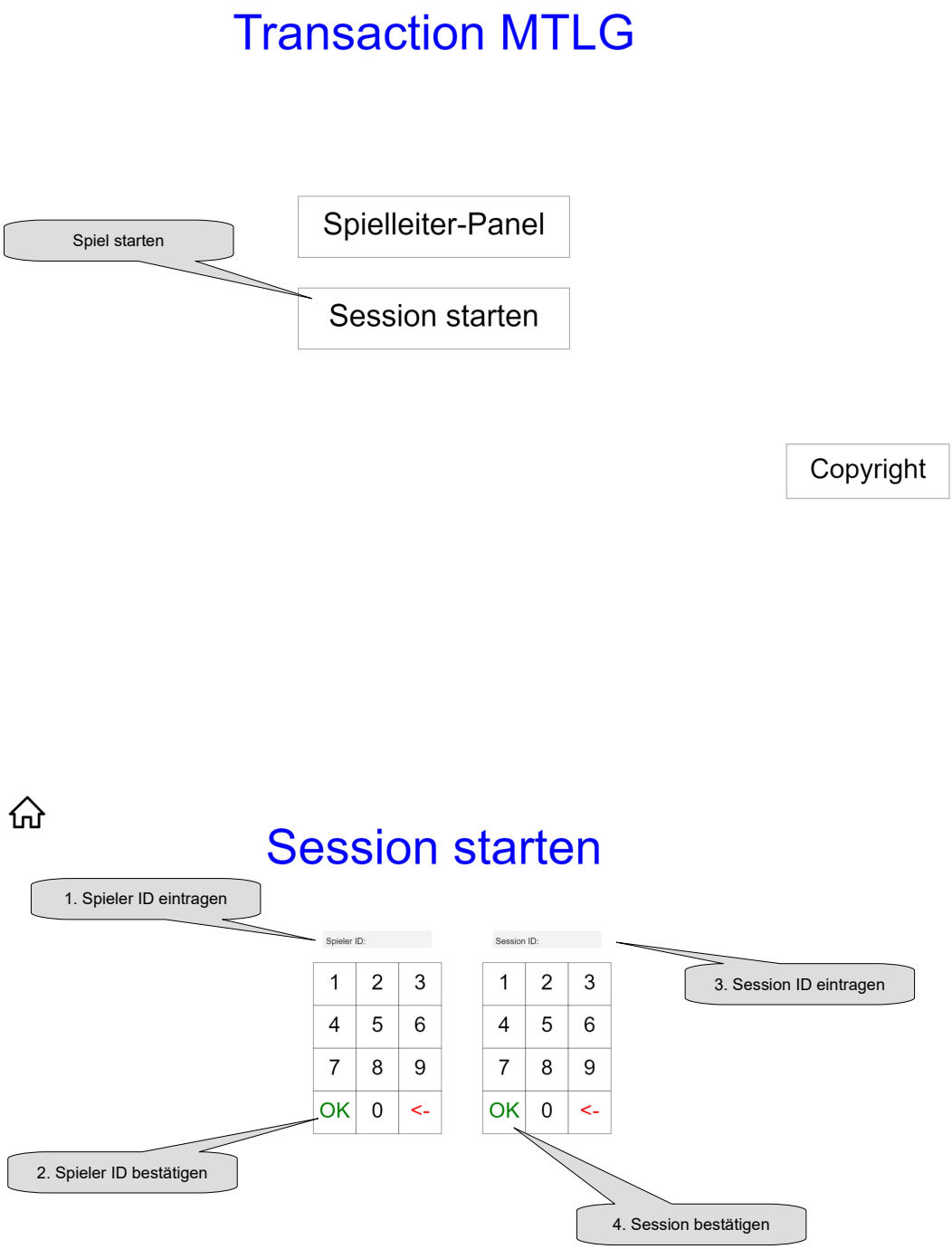
Hiermit willige ich ein, dass die personenbezogenen Daten die während der Studie der Masterarbeit „Learning Analytics in Serious Games mit xAPI“ von Jan-Patrick Schulz erhoben werden im Rahmen der Masterarbeit verwendet und ausgewertet werden.
Die personenbezogenen Daten werden in der Arbeit verwendet um die Gruppe der Befragten bestimmen zu können. Dabei werden nur die aggregierten Informationen über die Befragten in der Arbeit erwähnt.

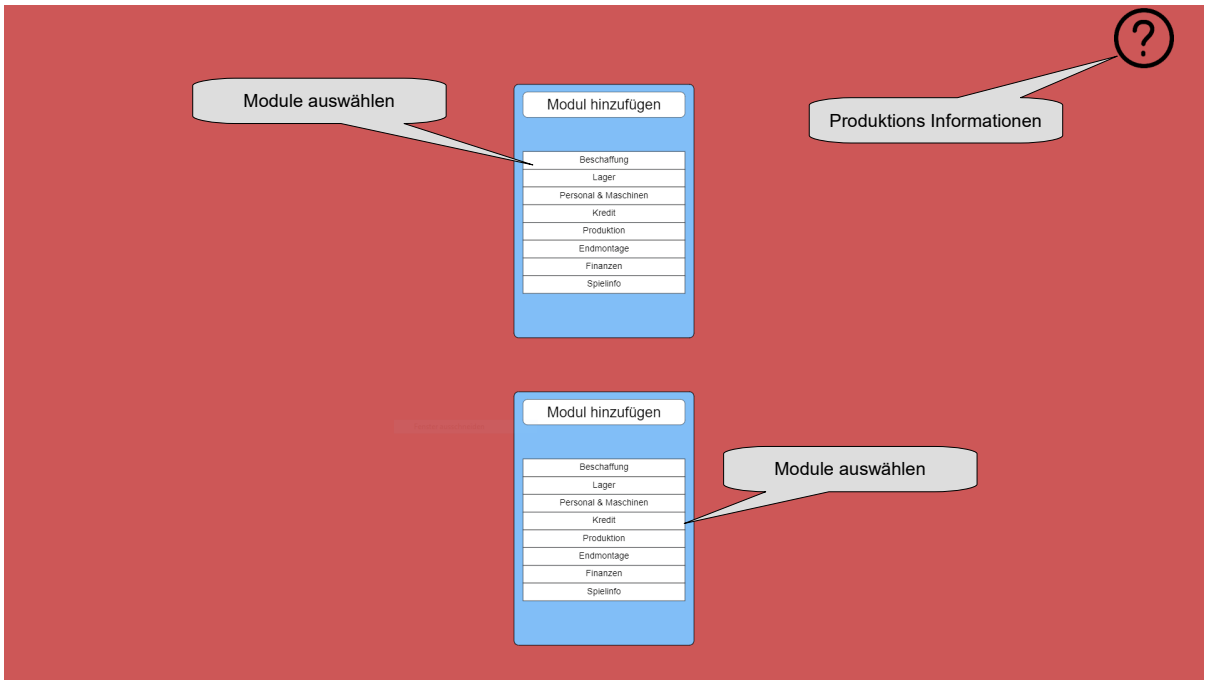
Dieser Einverständniserklärung dürfen Sie jederzeit widersprechen.

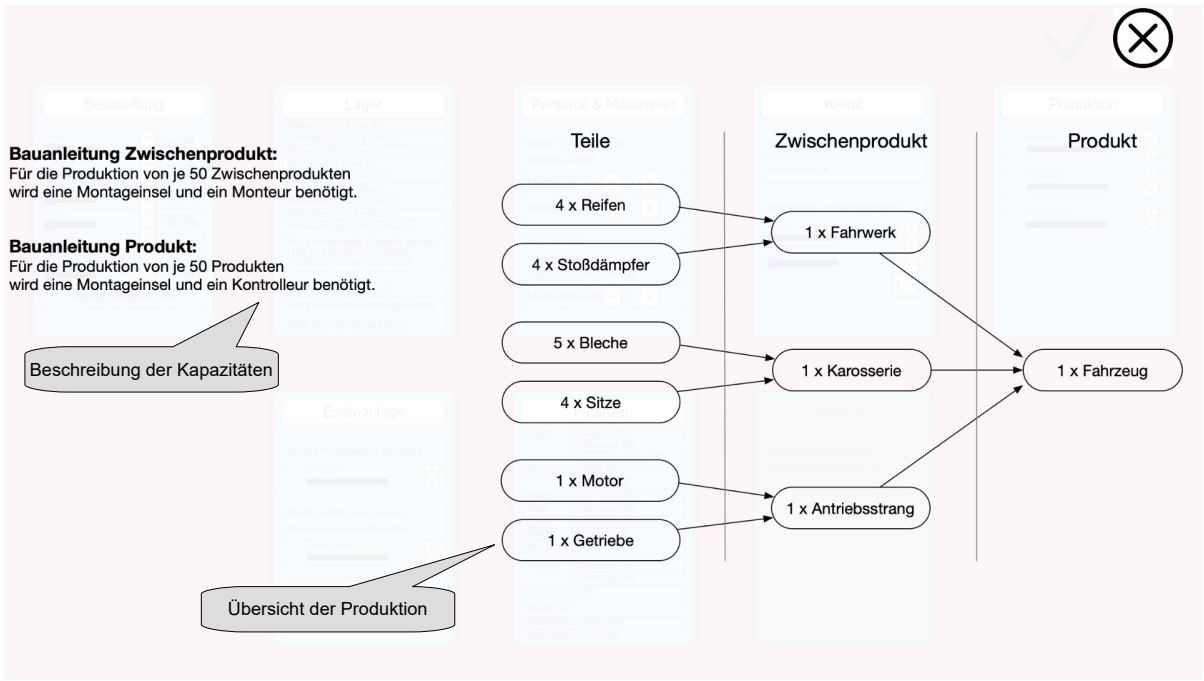
Die Teilnahme an der Studie kann jederzeit beendet werden.

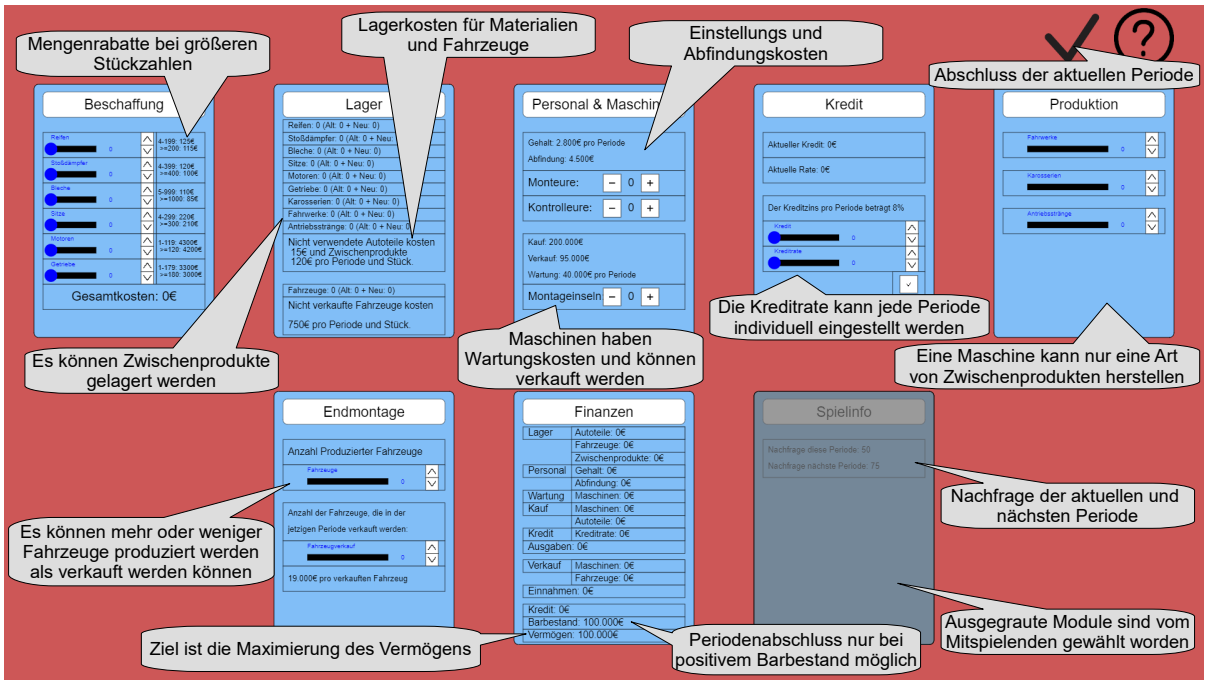
_____	_____	_____
Ort, Datum	Name	Unterschrift

B.2. Tutorial


Transaction MTLG
Spiel starten
Spielleiter-Panel
Session starten
Copyright
Session starten
1. Spieler ID eintragen
Spieler ID:
Session ID:
3. Session ID eintragen
1 2 3
4 5 6
7 8 9
OK 0 <-
2. Spieler ID bestätigen
4. Session bestätigen


Module auswählen
Modul hinzufügen
Beschaffung
Lager
Personal & Maschinen
Kredit
Produktion
Endmontage
Finanzen
Spielinfo
Produktions Informationen
Modul hinzufügen
Beschaffung
Lager
Personal & Maschinen
Kredit
Produktion
Endmontage
Finanzen
Spielinfo
Module auswählen


Bauanleitung Zwischenprodukt:
Für die Produktion von je 50 Zwischenprodukten
wird eine Montageinsel und ein Monteur benötigt.
Bauanleitung Produkt:
Für die Produktion von je 50 Produkten
wird eine Montageinsel und ein Kontrolleur benötigt.
Beschreibung der Kapazitäten
Übersicht der Produktion
Teile
Zwischenprodukt
Produkt
4 x Reifen
4 x Stoßdämpfer
5 x Bleche
4 x Sitze
1 x Motor
1 x Getriebe
1 x Fahrwerk
1 x Karosserie
1 x Antriebsstrang
1 x Fahrzeug


Mengenrabatte bei größeren Stückzahlen
Lagerkosten für Materialien und Fahrzeuge
Einstellungs und Abfindungskosten
Abschluss der aktuellen Periode
Beschaffung
Gesamtkosten: 0€
Lager
Nicht verwendete Autoteile kosten 15€ und Zwischenprodukte 120€ pro Periode und Stück.
Nicht verkaufte Fahrzeuge kosten 750€ pro Periode und Stück.
Personal & Maschin
Gehalt: 2.800€ pro Periode
Abfindung: 4.500€
Monteure:
Kontrolleure:
Kauf: 200.000€
Verkauf: 95.000€
Wartung: 40.000€ pro Periode
Montageinseln
Kredit
Aktueller Kredit: 0€
Aktuelle Rate: 0€
Der Kreditzins pro Periode beträgt 8%
Produktion
Es können Zwischenprodukte gelagert werden
Maschinen haben Wartungskosten und können verkauft werden
Die Kreditrate kann jede Periode individuell eingestellt werden
Eine Maschine kann nur eine Art von Zwischenprodukten herstellen
Endmontage
Anzahl Produzierter Fahrzeuge
Anzahl der Fahrzeuge, die in der jetzigen Periode verkauft werden:
19.000€ pro verkauften Fahrzeug
Finanzen
Lager
Autoteile: 0€
Fahrzeuge: 0€
Zwischenprodukte: 0€
Personal
Gehalt: 0€
Abfindung: 0€
Wartung
Maschinen: 0€
Kauf
Maschinen: 0€
Autoteile: 0€
Kredit
Kreditrate: 0€
Ausgaben: 0€
Verkauf
Maschinen: 0€
Fahrzeuge: 0€
Einnahmen: 0€
Kredit: 0€
Barbestand: 100.000€
Vermögen: 100.000€
Spielinfo
Nachfrage diese Periode: 50
Nachfrage nächste Periode: 75
Es können mehr oder weniger Fahrzeuge produziert werden als verkauft werden können
Nachfrage der aktuellen und nächsten Periode
Ziel ist die Maximierung des Vermögens
Periodenabschluss nur bei positivem Barbestand möglich
Ausgegraute Module sind vom Mitspielenden gewählt worden

B.3. Questionaire for Data Collection

Session ID: _____

Fragebogen zur Datenerhebung

im Rahmen der Masterarbeit Learning Analytics in Serious Games mit xAPI

Angaben zur Person

Geschlecht: ☐ männlich ☐ weiblich ☐ k.A.

Alter: ☐ <18 ☐ 18-26 ☐ 27-35 ☐ 36-44
☐ 45-53 ☐ 54-62 ☐ 63-70 ☐ 71<

Ausbildung: ☐ Studium abgeschlossen ☐ im Studium
☐ Ausbildung abgeschlossen ☐ in Ausbildung ☐ Schulabschluss

Ausbildungsrichtung:_____

Vorerfahrung

Haben Sie schon an einem Planspiel teilgenommen? ☐ ja ☐ nein

Haben Sie an einem Testlauf des Planspiels transaction MTLG teilgenommen? ☐ ja ☐ nein

Fragen	--	-	+	++
Wie gut haben Sie den Warenfluss des Unternehmens verstanden?	☐	☐	☐	☐
Wie konnte Ihr Team mit den folgenden Funktionen umgehen?				
– Einkauf	☐	☐	☐	☐
– Kreditaufnahme	☐	☐	☐	☐
– Kreditrückzahlung	☐	☐	☐	☐
– Produktion auf Lager	☐	☐	☐	☐
– Personal und Maschinenmanagement	☐	☐	☐	☐
Wie gut funktionierte Ihre Strategie im Bezug auf folgende Situationen?				
– Anpassung der Produktion an die Nachfrage	☐	☐	☐	☐
– Budgetverwendung	☐	☐	☐	☐
– Maschinenauslastung	☐	☐	☐	☐
– Vermögenswachstum	☐	☐	☐	☐
Wie lief die Kollaboration in Ihrem Team?	☐	☐	☐	☐
Wie gut haben Sie sich mit ihrem Mitspielenden abgesprochen?	☐	☐	☐	☐
War die Aktivität des Teams bei Ihnen ausgeglichen?	☐	☐	☐	☐
Haben Sie die Module sinnvoll aufgeteilt?	☐	☐	☐	☐
Mussten Sie häufig Korrekturen Ihrer Werte vornehmen?	☐	☐	☐	☐

Anmerkungen

B.4. Questionaire for Dashboard Usability

Questionaire for Dashboard Usability

as part of the master thesis Learning Analytics in Serious Games with xAPI

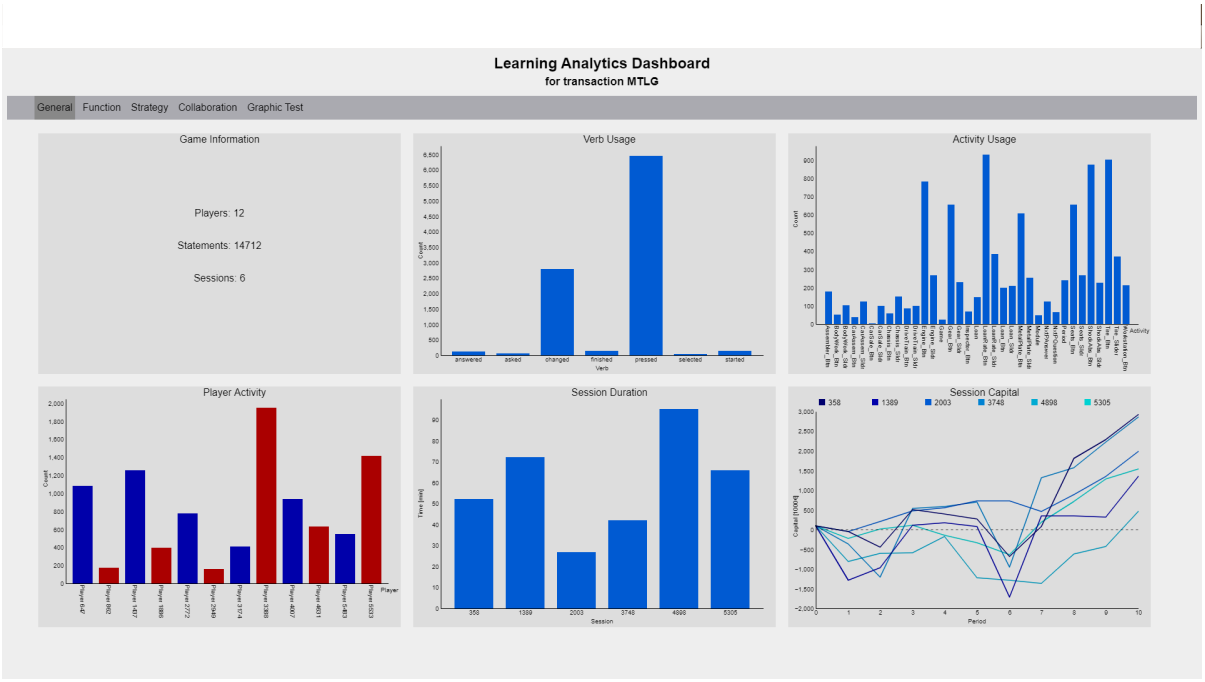
Previous Experience

Did you play the serious game transaction MTLG? ☐ yes ☐ no

Questions	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently	☐	☐	☐	☐	☐
2. I found the system unnecessarily complex	☐	☐	☐	☐	☐
3. I thought the system was easy to use	☐	☐	☐	☐	☐
4. I think that I would need the support of a technical person to be able to use this system	☐	☐	☐	☐	☐
5. I found the various functions in this system were well integrated	☐	☐	☐	☐	☐
6. I thought there was too much inconsistency in this system	☐	☐	☐	☐	☐
7. I would imagine that most people would learn to use this system very quickly	☐	☐	☐	☐	☐
8. I found the system very cumbersome to use	☐	☐	☐	☐	☐
9. I felt very confident using the system	☐	☐	☐	☐	☐
10. I needed to learn a lot of things before I could get going with this system	☐	☐	☐	☐	☐

Comments

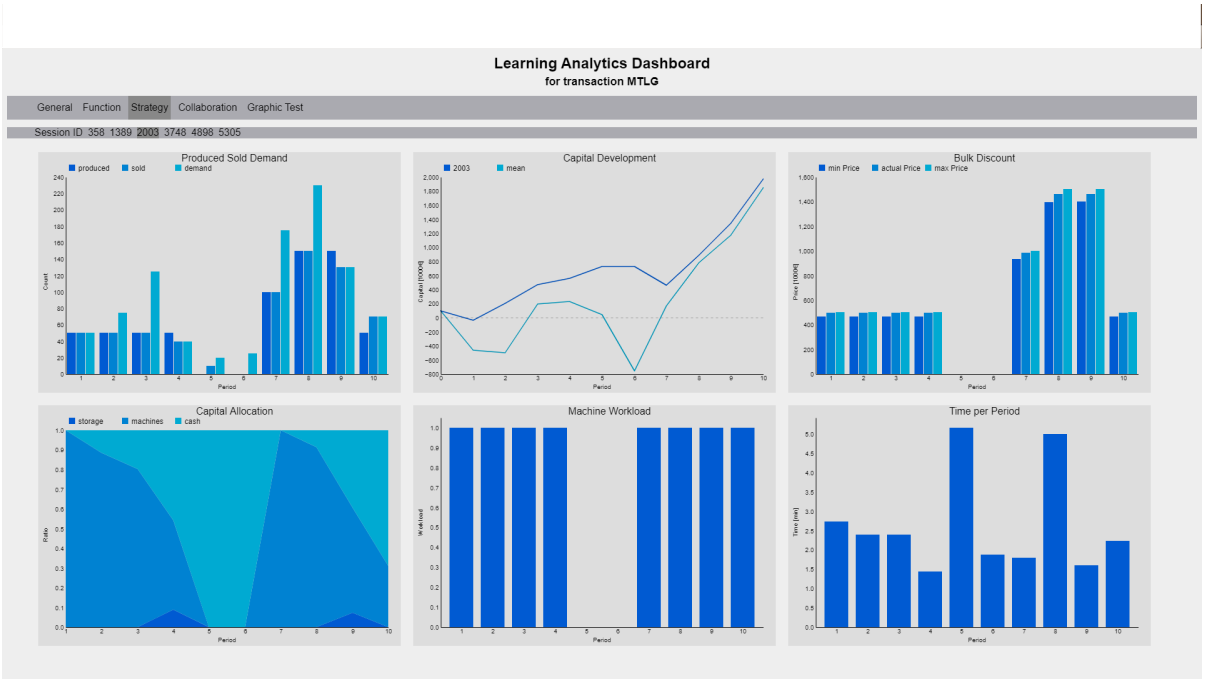
Appendix C Dashboard


Learning Analytics Dashboard
for transaction MTLG
General
Function
Strategy
Collaboration
Graphic Test
Game Information
Players: 12
Statements: 14712
Sessions: 6
Verb Usage
Activity Usage
Player Activity
Session Duration
Session Capital


Learning Analytics Dashboard
for transaction MTLG
General
Function
Strategy
Collaboration
Graphic Test
Cars for Storage
Loan Taken
Personal and Machine Management
Assembler
Inspector
Machine
Composition of Car Production
Session 358
Session 1389
Session 2003
Session 3748
Session 4898
Session 5305


Learning Analytics Dashboard
for transaction MTLG
General Function Strategy Collaboration Graphic Test
Session ID 358 1389 2003 3748 4898 5305
Produced Sold Demand
produced sold demand
Capital Development
358 mean
Bulk Discount
min Price actual Price max Price
Capital Allocation
storage machines cash
Machine Workload
Time per Period


Learning Analytics Dashboard
for transaction MTLG
General Function Strategy Collaboration Graphic Test
Session ID 358 1389 2003 3748 4898 5305
Produced Sold Demand
produced sold demand
Capital Development
1389 mean
Bulk Discount
min Price actual Price max Price
Capital Allocation
storage machines cash
Machine Workload
Time per Period


Learning Analytics Dashboard
for transaction MTLG
General Function Strategy Collaboration Graphic Test
Session ID 358 1389 2003 3748 4898 5305
Produced Sold Demand
produced sold demand
Count
Period
Capital Development
2003 mean
Capital [1000$]
Period
Bulk Discount
min Price actual Price max Price
Price [1000$]
Period
Capital Allocation
storage machines cash
Ratio
Period
Machine Workload
Workload
Period
Time per Period
Time [sec]
Period

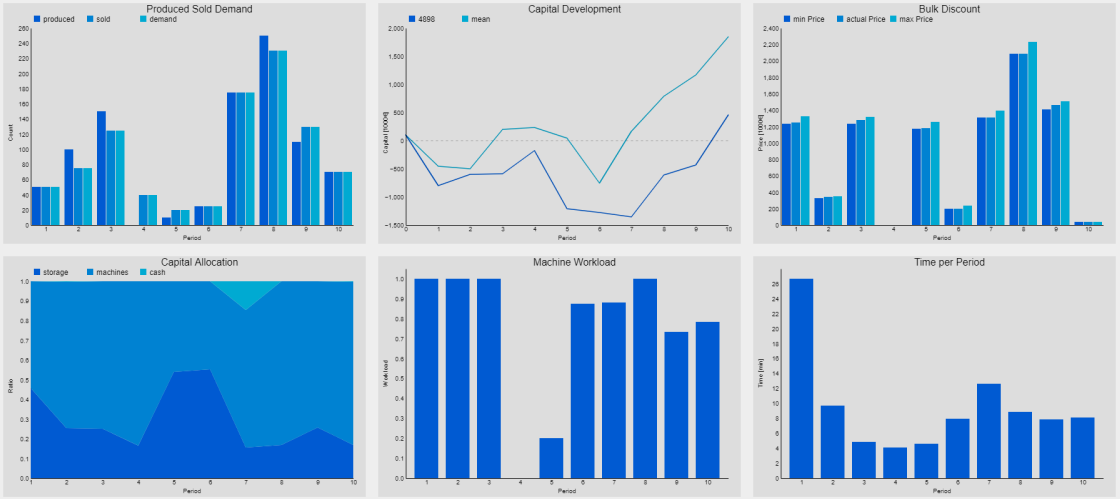

Learning Analytics Dashboard
for transaction MTLG
General Function Strategy Collaboration Graphic Test
Session ID 358 1389 2003 3748 4898 5305
Produced Sold Demand
produced sold demand
Count
Period
Capital Development
3748 mean
Capital [1000$]
Period
Bulk Discount
min Price actual Price max Price
Price [1000$]
Period
Capital Allocation
storage machines cash
Ratio
Period
Machine Workload
Workload
Period
Time per Period
Time [sec]
Period

Learning Analytics Dashboard

for transaction MTLG

General Function Strategy Collaboration Graphic Test

Session ID 358 1389 2003 3748 4898 5305

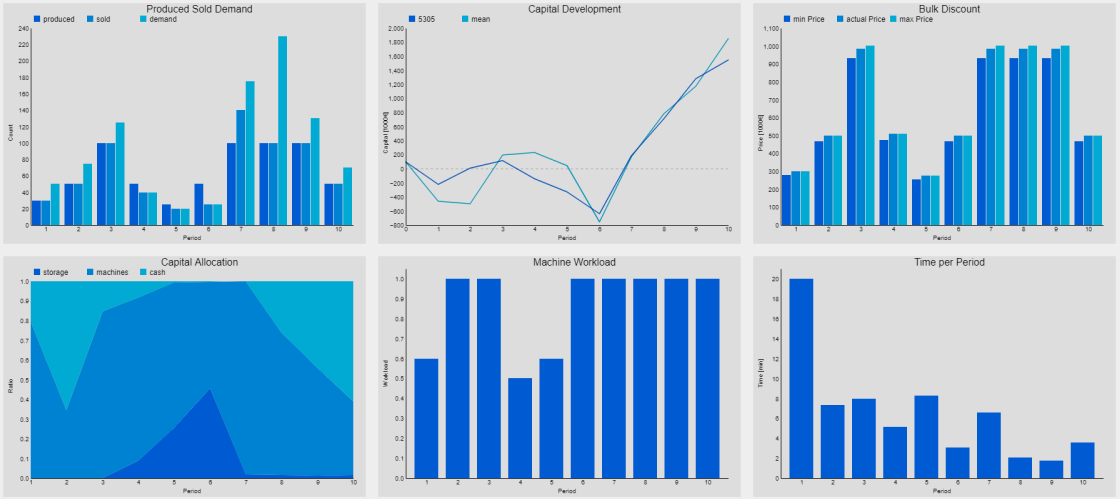


Learning Analytics Dashboard

for transaction MTLG

General Function Strategy Collaboration Graphic Test

Session ID 358 1389 2003 3748 4898 5305




Learning Analytics Dashboard
for transaction MTLG
General Function Strategy Collaboration Graphic Test
Session ID 358 1389 2003 3748 4898 5305
Module Selection
Player 647
- Finance
- Loan
- Procurement
Player 862
- Demand
- Storage
- Staff and Machines
- Production
- Assembly and Sale
User Activity per Period
Player 647 Player 862
Number of Statements
Period
User Activity per 30 seconds
Player 647 Player 862
active
inactive
Activity
Time [30s]
Next Period Question
Count
Player 647
Player
Changes of Values
Tire ShockAbs. Sheet Seat Engine Gear Assembler Inspector Machine Loan Installment LoanConf. Chassis
BodyWork DriveTrain Production Sold
Percent of max
Time [min]


Learning Analytics Dashboard
for transaction MTLG
General Function Strategy Collaboration Graphic Test
Session ID 358 1389 2003 3748 4898 5305
Module Selection
Player 1437
- Procurement
- Storage
- Loan
- Finance
Player 1886
- Demand
- Production
- Staff and Machines
- Assembly and Sale
User Activity per Period
Player 1437 Player 1886
Number of Statements
Period
User Activity per 30 seconds
Player 1437 Player 1886
active
inactive
Activity
Time [30s]
Next Period Question
Count
Player 1437
Player
Changes of Values
Tire ShockAbs. Sheet Seat Engine Gear Assembler Inspector Machine Loan Installment LoanConf. Chassis
BodyWork DriveTrain Production Sold
Percent of max
Time [min]

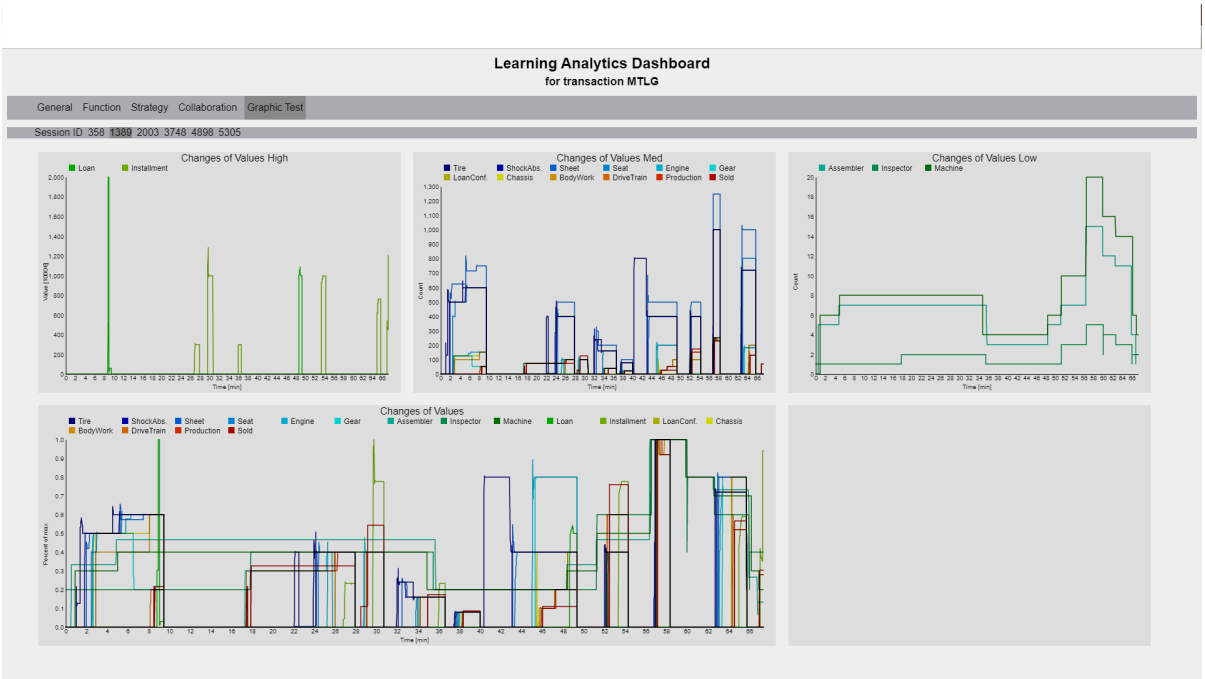

Learning Analytics Dashboard
for transaction MTLG
General Function Strategy Collaboration Graphic Test
Session ID 358 1389 2003 3748 4898 5305
Module Selection
Player 2772
- Procurement
- Storage
- Production
- Assembly and Sale
Player 2949
- Demand
- Staff and Machines
- Finance
- Loan
User Activity per Period
Player 2772 Player 2949
Number of Statements
Period
User Activity per 30 seconds
Player 2772 Player 2949
active
inactive
Activity
Time [30s]
Next Period Question
Count
Player 2772
Player 2949
Player
Changes of Values
Tire ShockAbs. Sheet Seat Engine Gear Assembler Inspector Machine Loan Installment LoanConf. Chassis
BodyWork DriveTrain Production Sold
Percent of max
Time [min]


Learning Analytics Dashboard
for transaction MTLG
General Function Strategy Collaboration Graphic Test
Session ID 358 1389 2003 3748 4898 5305
Module Selection
Player 3174
- Storage
- Loan
- Assembly and Sale
- Demand
Player 3388
- Procurement
- Staff and Machines
- Production
- Finance
User Activity per Period
Player 3174 Player 3388
Number of Statements
Period
User Activity per 30 seconds
Player 3174 Player 3388
active
inactive
Activity
Time [30s]
Next Period Question
Count
Player 3174
Player 3388
Player
Changes of Values
Tire ShockAbs. Sheet Seat Engine Gear Assembler Inspector Machine Loan Installment LoanConf. Chassis
BodyWork DriveTrain Production Sold
Percent of max
Time [min]


Learning Analytics Dashboard
for transaction MTLG
General Function Strategy Collaboration Graphic Test
Session ID 358 1389 2003 3748 4898 5305
Module Selection
Player 4007
- Procurement
- Staff and Machines
- Demand
- Assembly and Sale
Player 4631
- Storage
- Loan
- Production
- Finance
User Activity per Period
Player 4007 Player 4631
Number of Statements
Period
User Activity per 30 seconds
Player 4007 Player 4631
active
inactive
Activity
Time [30s]
Next Period Question
Count
Player 4007
Player
Changes of Values
Tire ShockAbs. Sheet Seat Engine Gear Assembler Inspector Machine Loan Installment LoanConf. Chassis
BodyWork DriveTrain Production Sold
Percent of max.
Time [min]


Learning Analytics Dashboard
for transaction MTLG
General Function Strategy Collaboration Graphic Test
Session ID 358 1389 2003 3748 4898 5305
Module Selection
Player 5483
- Staff and Machines
- Loan
- Finance
- Storage
- Demand
Player 5533
- Production
- Assembly and Sale
- Procurement
User Activity per Period
Player 5483 Player 5533
Number of Statements
Period
User Activity per 30 seconds
Player 5483 Player 5533
active
inactive
Activity
Time [30s]
Next Period Question
Count
Player 5483
Player 5533
Player
Changes of Values
Tire ShockAbs. Sheet Seat Engine Gear Assembler Inspector Machine Loan Installment LoanConf. Chassis
BodyWork DriveTrain Production Sold
Percent of max.
Time [min]


Learning Analytics Dashboard
for transaction MTLG
General Function Strategy Collaboration Graphic Test
Session ID 358 1389 2003 3748 4898 5305
Changes of Values High
Loan Installment
Changes of Values Med
Tire ShockAbs. Sheet Seat Engine Gear
LoanConf. Chassis BodyWork DriveTrain Production Sold
Changes of Values Low
Assembler Inspector Machine
Changes of Values
Tire ShockAbs. Sheet Seat Engine Gear Assembler Inspector Machine Loan Installment LoanConf. Chassis
BodyWork DriveTrain Production Sold
Time [min]

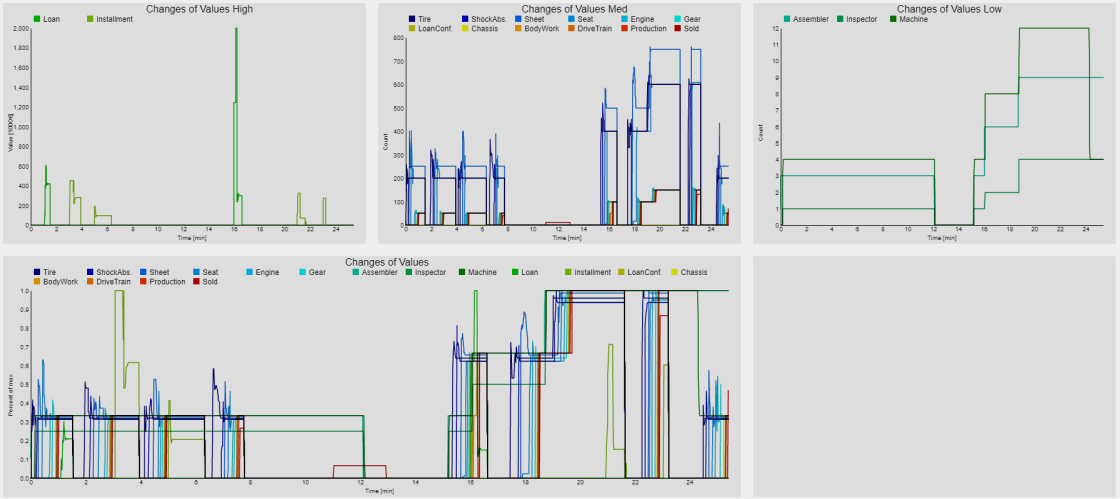

Learning Analytics Dashboard
for transaction MTLG
General Function Strategy Collaboration Graphic Test
Session ID 358 1389 2003 3748 4898 5305
Changes of Values High
Loan Installment
Changes of Values Med
Tire ShockAbs. Sheet Seat Engine Gear
LoanConf. Chassis BodyWork DriveTrain Production Sold
Changes of Values Low
Assembler Inspector Machine
Changes of Values
Tire ShockAbs. Sheet Seat Engine Gear Assembler Inspector Machine Loan Installment LoanConf. Chassis
BodyWork DriveTrain Production Sold
Time [min]

Learning Analytics Dashboard

for transaction MTLG

General Function Strategy Collaboration Graphic Test

Session ID 358 1389 2003 3748 4898 5305

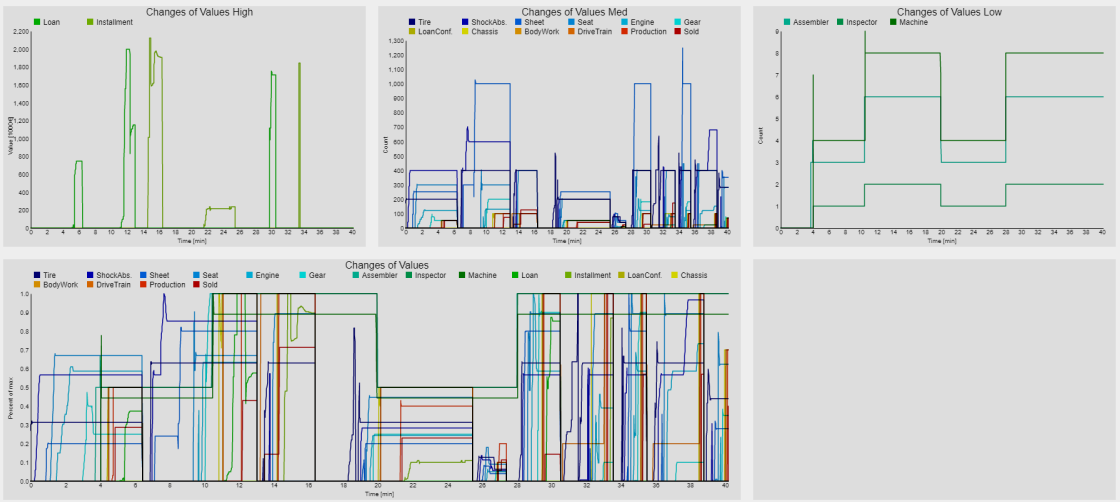


Learning Analytics Dashboard

for transaction MTLG

General Function Strategy Collaboration Graphic Test

Session ID 358 1389 2003 3748 4898 5305

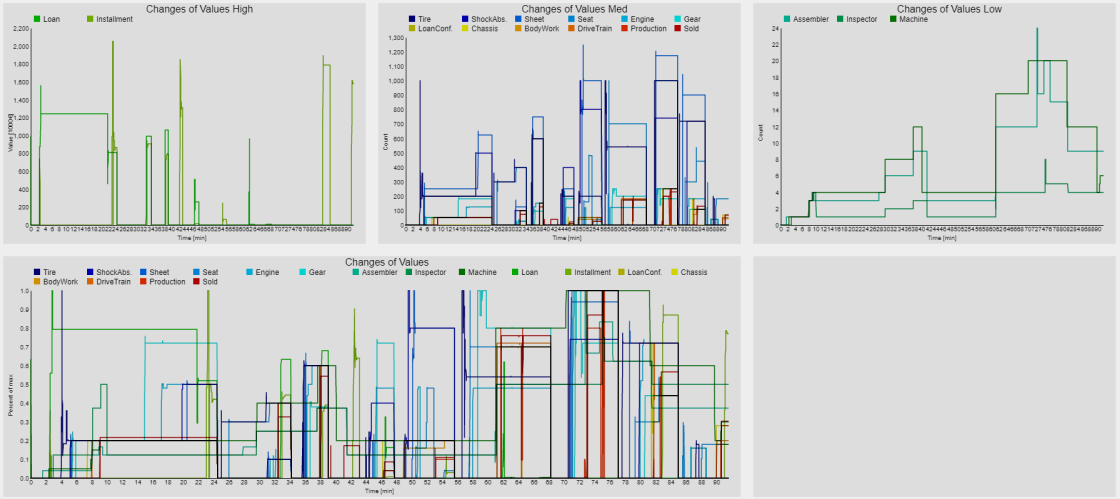


Learning Analytics Dashboard

for transaction MTLG

General Function Strategy Collaboration Graphic Test

Session ID 358 1389 2003 3748 4898 5305

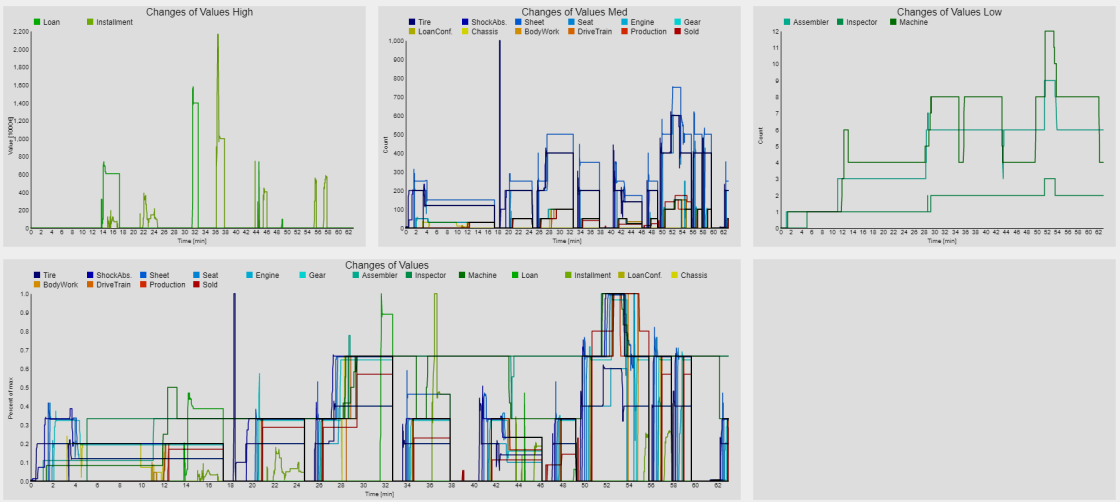


Learning Analytics Dashboard

for transaction MTLG

General Function Strategy Collaboration Graphic Test

Session ID 358 1389 2003 3748 4898 5305



Eidesstattliche Versicherung

Schulz, Jan-Patrick 354467

Name, Vorname Matrikelnummer (freiwillige Angabe)

Ich versichere hiermit an Eides Statt, dass ich die vorliegende Arbeit/Bachelorarbeit/ Masterarbeit* mit dem Titel

Learning Analytics in Serious Games with xAPI

selbstständig und ohne unzulässige Hilfe erbracht habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt. Für den Fall, dass die Arbeit zusätzlich auf einem Datenträger eingereicht wird, erkläre ich, dass die schriftliche und die elektronische Form vollständig übereinstimmen. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Aachen , May 28, 2021

Ort, Datum Unterschrift

*Nichtzutreffendes bitte streichen

Belehrung:

§ 156 StGB: Falsche Versicherung an Eides Statt
Wer vor einer zur Abnahme einer Versicherung an Eides Statt zuständigen Behörde eine solche Versicherung falsch abgibt oder unter Berufung auf eine solche Versicherung falsch aussagt, wird mit Freiheitsstrafe bis zu drei Jahren oder mit Geldstrafe bestraft.

§ 161 StGB: Fahrlässiger Falscheid; fahrlässige falsche Versicherung an Eides Statt
(1) Wenn eine der in den §§ 154 bis 156 bezeichneten Handlungen aus Fahrlässigkeit begangen worden ist, so tritt Freiheitsstrafe bis zu einem Jahr oder Geldstrafe ein.
(2) Straflosigkeit tritt ein, wenn der Täter die falsche Angabe rechtzeitig berichtigt. Die Vorschriften des § 158 Abs. 2 und 3 gelten entsprechend.

Die vorstehende Belehrung habe ich zur Kenntnis genommen:

Aachen , May 28, 2021

Ort, Datum Unterschrift