

# Developing and Testing a Mobile Learning Games Framework

Carsten Busch, Sabine Claßnitz, André Selmanagić and Martin Steinicke

Creative Media Research Group, University of Applied Sciences HTW-Berlin, Germany

[carsten.busch@htw-berlin.de](mailto:carsten.busch@htw-berlin.de)

[sabine.classnitz@htw-berlin.de](mailto:sabine.classnitz@htw-berlin.de)

[andre.selmanagic@htw-berlin.de](mailto:andre.selmanagic@htw-berlin.de)

[martin.steinicke@htw-berlin.de](mailto:martin.steinicke@htw-berlin.de)

**Abstract:** In 2010 1.1 million pupils took private lessons in Germany, with 25% of all German children by the age of 17 having attended paid private lessons at some point in their school career (Klemm & Klemm, 2010). The high demand for support for learning curricular content led us to consider an integrated solution that speeds up both the design of mobile learning games as well as their implementation and adaptation. This paper describes the iterative development of a game development framework for touch-based mobile learning games. The framework focuses on touch-controlled interaction due to the fact that in 2014 more than 87% of German teenagers possessed a smart phone with touch input (Feierabend, Plankenhorn, Rathgeb, 2014) as well as the possibility to engage in short bursts of learning experiences during their idle time, e.g. when commuting. The framework consists of a conceptual component that specifies five different game modes for casual mobile learning games. The technical part of the framework builds on the Unity game engine and offers an architecture that mirrors the game modes and objects from the conceptual part as well as a layer of service building blocks that cover generic functionality like logging, high score management or social media integration. The development of the framework is iterative and cyclic in that each produced game enriches the framework, which in turn accelerates the prototyping and development of further games. Additionally the games themselves are developed and tested iteratively – both concerning usability/user-experience and transfer, which is described in this paper.

**Keywords:** mobile learning games, touch interfaces, private lessons, usability, software framework, transfer

---

## 1. Introduction

Private tutoring – sometimes referred to as shadow education (Entrich, 2014) – is a huge and growing market. In 2010 1.1 million pupils took private lessons in Germany, with 25% of all German children by the age of 17 having attended paid private lessons at some point in their school career (Klemm & Klemm, 2010). Though Austria shows similar numbers (20%), the amount of paid private lessons varies greatly across Europe. While only 8% of secondary pupils from the UK receive tutoring, surveys in Lithuania showed that 62% of university students had received private supplementary tutoring in their last year of secondary school (Bray, 2011). The reasons for this development are diverse and often viewed in a negative light, in that they foster inequality, performance pressure and a competitive attitude (Klemm & Klemm, 2010; Bray, 2011). Other authors like Entrich (2014) highlight that private tutoring might be beneficial in a social sense, as it could have a neutralizing effect on disadvantaged family backgrounds.

When success in learning is to a large extent coupled with knowing the basics, a vicious cycle is easily set into motion. Failure creates anxiety, thus motivation (Csikszentmihalyi, 2010) and perceived self-efficacy (Bandura, 1977) declines. Negative emotion avoidance strategies (Persons, 2010) are developed, and thus future success in school becomes less and less likely. In today's performance culture, parents commonly seek professional help even for their well performing children (Klemm & Klemm, 2010). From a motivational standpoint this may actually further foster the decline of pupils' intrinsic motivation through the perceived parental, thus extrinsic, motivation (Ryan & Deci, 2000:56) and the potential loss of social/peer status.

Digital game-based learning may be used to counter these points. Clearly associative approaches fit into the specific use-case of repetitive training. Pupils need to memorize curriculum-specific content like facts and figures as well as develop a seemingly intuitive grasp of methods and concepts that, like in basic math or language lessons, are often only acquired by such repetitive training. With these approaches facts and figures can be learned in less time compared to systemic or constructivist approaches. Additionally, as the learning content is pretty apparent, parents understand why their children should play these games as a learning aid. And from the child's perspective, even a game, which is clearly learning-oriented, may be an appealing alternative compared to learning from other sources. Furthermore, such games may be helpful to break the

above described vicious circle of failure, self-efficacy reduction, avoidance strategies resulting in renewed failure alternately creating moments of success that may foster content-specific self-efficacy.

Compared to such conceptually embedded games (Clark & Gaza, 2012), which embed learning content into existing game mechanics that have a weak or no relation to the learning goals, conceptually integrated games facilitate deep systemic but oftentimes implicit learning. Due to this, the latter do need a teacher or a special game mechanic to support the transfer of the primarily implicit understandings (Clark & Gaza, 2012:280). Thus while the former are ideal for rote learning at home or while commuting as well as relieving the burden of testing by the tutor, the latter engage in deep learning during individual idle time so that the tutoring sessions can be used for discussing concepts, problems and motivational aspects instead. An approach that is rather similar to the concept of the flipped classroom (Tucker, 2012).

## **2. The Conceptual Component of the Game Framework**

With the aim of using mobile learning games in collaboration with private lessons and tutoring, the question arose whether to create an integrated learning experience that teaches multiple school subjects at once or to focus on a palette of games that each teaches a specific subject. Due to the fact that parents commonly only enroll children for one or two courses, we decided to create a game framework that enables both the design and implementation of small games that focus on the latter. In this context “framework” does not denote a pedagogical framework as in (Park, 2011) or a conceptual framework as in (Yusoff, Crowder, Gilbert, & Wills 2009), but rather a software framework, which “is an architecture or infrastructure intended to support and enable the integration and interoperation of software components. The essential idea is that components developed to be compliant and consistent with a framework may be combined, connected, and composed within that framework, and that such compositions may be assembled more readily and with more likelihood of correct operation than would be possible without the framework.” (Petty, Kim, Barbosa, & Pyun, 2014)

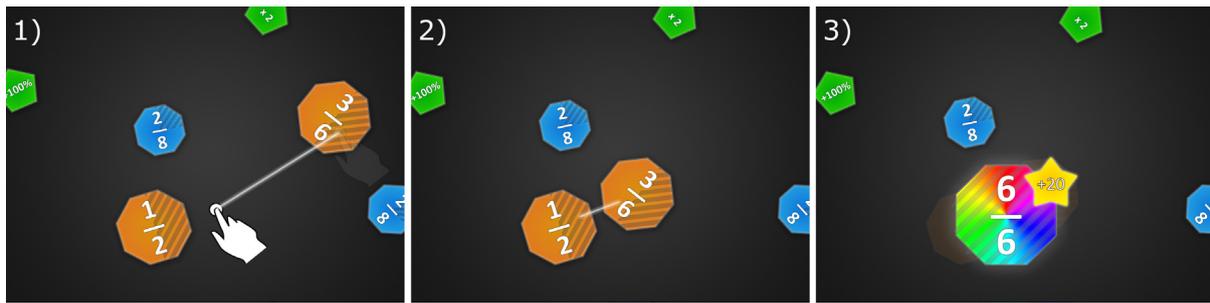
The conceptual part of this framework, which provides building blocks for designing touch-based games, will be described below.

### **2.1 Inspiration**

To reach our goal of creating games that support the client’s educational goals while simultaneously being sufficiently entertaining so that students aged 10 to 16 would play them as exercises in their spare time, we started with a market research followed by brainstorming sessions. Although we were well aware of (and for some time even shared) the critical attitude towards learning games that do not integrate the learning content into the games mechanics (see e.g. Egenfeldt-Nielsen, 2007), we came to see that the approach of choice depends primarily on the context and learning goals.

One entertainment game that we identified as promising in our market research is the iOS game Super 7 (No Monkeys, 2010), which inspired the framework described in this paper. The core game mechanic of Super 7 is simple: by drawing paths on the touchscreen, the player has to bring floating digits together until they add up to the number 7. Reaching the summation of 7, scores points. However numbers greater than 7 result in “Game Over”. Super 7 provides a challenging gameplay while teaching basic mathematical concepts like addition and division along the way.

With mathematics being one of the most problematic subjects for many students in our target group (Stiftung Warentest, 2006), our initial idea was to transfer Super 7’s game mechanic to more advanced mathematical problems like fractional arithmetic and percentages. In such a game the goal would be to sum fractions up to 1 (see figure 1). Using color-coding and varying sizes, a game like this should give students a sense of how decimal values and fractions relate to each other and show them how the same value can be represented by multiple fractions. It could also depict how fractions and percentages are related.



**Figure 1:** Mock-up example of the basic game mechanic with fractional arithmetic.

For our goal to create a number of mobile learning games, we decided to abstract this basic game mechanic, and reconstruct it into a framework consisting of a conceptual as well as technological component. It features a handful of different touch-based input methods in addition to the creation of paths used in Super 7, and a set of different ways in which objects can interact with each other. This framework would serve as a foundation for the game instances, enabling us to mix these game parameters and create similar, but diverse, gaming experiences.

## 2.2 Core Game Mechanic and Game Objects

In its current state the framework consists of five different *game modes*, which are loosely based on the same core game mechanic. Game parameters were classified into groups, including variables of game objects and global parameters. Each game mode represents a different combination of these classified parameters (mixture of the global parameter values and game objects that have mixed parameters as well). The game instances, hereinafter called *game modules*, are implementations of the abstract game modes.

### 2.2.1 Core Game Mechanic

The playing field is static and two-dimensional. It can be populated by visual objects, which may either float freely or are placed at fixed positions. Depending on their types, the player's objective is to bring these objects together and combine them, or to keep them apart using touch gestures, for instance by drawing paths for the objects to follow or by swiping them in specific directions. Encounters of the "correct" (contextually associated) objects scores points, whereas collisions of "wrong" objects whether intended or not, drains health points and eventually leads to the end of the game. The player's objective is to score as many points as possible before all health is depleted.

For example: In a game, in which the player's objective is to associate countries with their respective continent, bringing "Denmark" and "Europe" together would score points, whereas matching "China" and "Africa" would drain health.

### 2.2.2 Game Objects

We classified the possible variables and values for game objects so that every game object could be built upon the same foundation. All types of game objects share the same basic variables, but with varying values. Each game mode (and game module) has a unique gameplay due to its different mix of these factors. The main variables are as followed:

**Table 1:** Game object variables and their allowed values

Variable	Description and possible values
Mobility	Game objects can be placed at fixed positions or move freely. Moving objects have a fixed speed and direction, with the exception of objects that the player interacts with, e.g. by providing a path for the objects to follow.
Limited by playing field boundaries	Objects may be restricted by the boundaries of the playing field. If so, they are deflected by the borders. Otherwise they simply leave the play space.
Appearance / Spawning	Objects either exist at the beginning of the round or enter the playing field from the outside during the game.
Player influence	<p>The player may alter the position of objects in various ways.</p> <p><u>Values:</u></p> <ul style="list-style-type: none"> <li>● <i>None</i>: The player cannot influence the object in any way.</li> <li>● <i>Follow path</i>: The player can draw a path on the screen. If the path crosses the object, the object will follow it. (Touch gesture: drag)</li> <li>● <i>“Holdable”</i>: The player can hold a moving object (keep it from moving). Once the player lets go, the object begins to move again (Touch gesture: press)</li> <li>● <i>Follow path and “holdable”</i>: Mixture of “Follow path” and “Holdable” (Touch gestures: drag, press)</li> <li>● <i>Directly controlled</i>: The player can directly change the position of the object by dragging it to a different location. (Touch gesture: press &amp; drag)</li> <li>● <i>Deflectable</i>: The player can change the direction of a moving object. (Touch gesture: swipe)</li> <li>● <i>Tap action</i>: When the player taps the object, a custom action is executed. (Touch gesture: single tap)</li> <li>● <i>Custom action on path encounter</i>: When the player draws a path over the object, a custom action is executed. (Touch gesture: drag)</li> </ul>
Collisions	<p>Depending on the types of two colliding game objects, different results may occur.</p> <p><u>Values:</u></p> <ul style="list-style-type: none"> <li>● <i>Pass through</i>: No physical interaction. No change in points or health.</li> <li>● <i>Deflection</i>: Both game objects bounce off of each other. No change in points or health.</li> <li>● <i>Dissolution (both)</i>: Both game objects dissolve. Depending on the types of the objects, points may be given or health is drained.</li> <li>● <i>Dissolution (one)</i>: One of the two game objects dissolves. Depending on the types of the objects points may be given or health drained.</li> <li>● <i>Dissolution with creation</i>: Both game objects dissolve, but one or more new game objects are created. Depending on the types of the objects, points may be given or health drained.</li> </ul>

### 2.2.3 Basic Variables of a Game Mode

The game modes themselves also have a set of variables, including the maximum number of paths the player can draw for objects to follow, the maximum number of objects on the same path, in which direction of the path objects are moving (towards the start, the end or the centre) and whether paths are shrinking with time. Another important variable is the number of health points, e.g. the number of mistakes a player can make before the round is lost.

## 2.3 Game Modes

As stated, all game modes are basically combinations of the different possible values provided by the game objects and the basic variables of a game mode. For the framework five game modes have been specified. Based on these game modes we created concepts for several games (game modules), targeting various subjects from fractional arithmetic to musical history to counting in different languages or numeral systems. Some of these are described with each game mode.

### 2.3.1 Game Mode: "Categories"

The game mode "Categories" defines two different types of game objects: categories and elements. Categories exist from the beginning of the game, and have fixed positions that cannot be altered by the player. Elements are moving and enter the playing field from the outside. Each element belongs to a specific category. The player's objective is to guide the elements to their respective categories, but also circumvent unintended encounters of non-associated elements and categories. When a category and an element collide, the element dissolves. The algorithm may replace categories with other categories during the game. The previous example of associating continents and countries is a representative of this game mode with continents being the categories and countries the elements.

The game mode is divided into three different variations ("sub game modes"):

In *Variation A*, once an element has entered the playing field, it cannot leave the play area. The player can draw paths, which the elements will then follow to the end; however, the player can also hold the elements. With new elements entering the field and old ones unable to leave (unless associated with a category), the players will find themselves under increasing pressure.

*Variation B ("Comets")* features elements that enter the screen like comets flying in a random direction. They are not limited by the playing field boundaries and can therefore leave the screen. The player can deflect them by swiping them in the desired direction, e.g. towards a category or out of the playing field. By letting the correct elements leave the screen, the player loses the chance to score points. On the other hand it reduces the risk of false associations if the player is in doubt.

In *Variation C ("Boxes")* coincidental collisions of elements and categories do not count as an association of the two. Elements will just bounce off. To bring an element and a category together, the player needs to drag it onto the category. In contrast to the other two variations, where elements and categories can unintentionally collide, the player only loses health if the wrong decision was made.

#### Game Module: "Continents & Countries"

The educational objective of "Continents & Countries" is to teach the player, in which continent a given country is. The game module is based on the game mode "Categories" variation B ("Comets"). A globe is placed in the center of the screen with the active continent facing the player while individual countries enter the playing field from outside. The player can use swipe gestures to guide the countries in specific directions, either towards or away from the continent. For correctly associating a continent with a country, the player scores points. If the player sends the wrong country to the displayed continent (e.g. "France" to "North America"), the player loses one health point. The game is lost once all health points are depleted. The active continent is repeatedly swapped.

As the game progresses the difficulty increases, which has three consequences. First, with increasing difficulty, the countries will begin to move faster and spawn more often. Secondly, the "wrong" countries are more likely to fly towards the continent, forcing the player to react. Lastly, the set of continents and countries, which

enter the field, are related to the chosen difficulty level, being divided into groups from countries that are most commonly known (e.g. industrial countries and Germany's neighboring countries) to lesser known countries (e.g. smaller island countries). Lesser known countries score more points, although this can be adapted easily as the sorting of countries into difficulty levels is located in a CSV text file. The prototypical implementation of this game module, called "Crazy Robots", will be described in section 3.4.

#### Game Module: "One, Deux, Tres"

"One, Deux, Tres" uses a similar concept for teaching counting in different languages. Instead of countries, numerals in word-form from varying languages are entering the playing field. Four categories are shown simultaneously in the corners of the screen, representing numbers in the form of western Arabic numerals (symbols). The player's task is to associate the words with the correct Arabic numerals. The numerals in word-form are color-coded by their respective language (figure 2). The module utilizes sub game modes in Variation A or C (see 2.3.1) depending on the specific implementation of the rule system.

In addition, this module can be enhanced with a mathematical context so that new numeral objects can be created by combining two or more numeral words with mathematical operators such as addition, subtraction, multiplication and division. The result of such a game action is an object that contains the result of the mathematical operation with the two Arabic numerals, which will then disappear from the game field. The languages used in a game session can be chosen individually via options.



**Figure 2:** Mock-up example of "One, Deux, Tres".

#### 2.3.2 Game Mode: "Combination"

In the game mode "Combination" the player needs to draw paths to combine objects, similar to the gameplay of Super 7.

Objects in this mode are specified as mobile and can hold different types. They enter the game field from outside and keep moving within the dimension of the play area. The player can combine elements by drawing a path between them or can keep items apart by swiping them to change the direction of their motion. A long press on a game object results in "holding" the object – meaning it remains on its current position until released. Once a path was created, it will shrink from both sides, which over time results in a collision of the involved items. Other independently moving objects can collide as well. The specific handling of these events e.g. evaluation within the game context is part of the respective game module implementation.

For the creation of game modules the game mode "Combination" provides two sub variants regarding the basic handling of collisions as well as in the maximum number of objects a path can hold:

In *Variation A* two colliding game objects merge to a new game object and a path can hold an arbitrary number of elements. This variant is suitable for games like the previously mentioned math game "Super 7". In "Super 7", game objects are represented by integer numbers, algebraic signs (+/-) and mathematical operators such as multiplication or division by a specific value. When two or more numbers collide, they merge and sum up their values to a new integer number. Combinations of number objects with algebraic signs or mathematical operators change the current value of the number object and merge with it.

In *Variation B*, the colliding items of different types dissolve and a path can hold two objects at maximum. An example would be an application where one-to-one allocations have to be found for example the correct combination of a flag with a country. Two colliding flags will reflect from each other whereas the collision of a country with a flag will result in a merging process.

#### Game Module: “Fractional Arithmetic”

As mentioned previously the “Fractional Arithmetic” game (see figure 1) teaches about the relation of decimal values and fractions and how multiple fractions form the same value. For this, the game mode in *Variation A* is utilized. Players will combine fractions by drawing paths between them with the goal of forming the value of “1”. On object collisions the values of the involved elements are summed up and the result is the value of a new fraction item (see figure 1). If the new value is 1 the merged fraction object disappears and the player is rewarded with points. If the result exceeds 1 the game is over. Objects with values smaller than 1 remain floating in the playing area. Similar to the gameplay of “Super 7” other elements such as mathematical operators and algebraic signs are available to modify the value of existing fractional items.

#### 2.3.3 Game Mode: “Positioning”

The main task of the game mode “Positioning” is to place certain objects in the “correct” position on the playing field. Two types of game objects are specified in this mode: backgrounds and elements. Similar to “category” game objects (see 2.3.1) backgrounds are available from game start and cannot be modified by the player directly. Backgrounds provide the context to which one or more elements relate. Elements can be initially fixed or mobile. The player will move them to a position in the playing area where they are locked. While mobile elements enter the playing area from outside and keep moving within its dimension initially fixed elements are apparent in a specific location from the beginning of a game session. The player's goal is to drag the elements in the right positions a background provides e.g. placing floating punctuation marks in the correct positions in a displayed sentence.

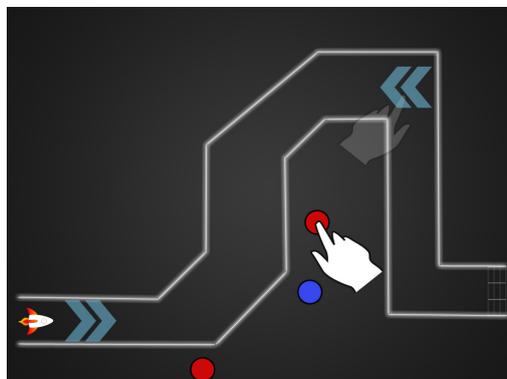
The game mode offers two variants with respect to object collision and movement settings.

*Variation A* provides elements, which are initially mobile and do bounce on collisions with each other.

In *Variation B* the elements are initially displayed in a fixed location e.g. within a menu where the player can choose from by tapping on an item. Once an element was selected the player can place this item by tapping on a new position in the playing area.

#### Game Module: “SURGE Clone”

Whereas the previous examples use learning by association to teach facts and figures, more constructivist approaches are also imaginable. SURGE II is a learning game that teaches Newtonian mechanics, in which the player needs to navigate a spaceship through a narrow pathway by placing commands (forces, impulses, acceleration, etc.) in the game world, thereby modifying the motion of the spaceship (Clark & Gaza, 2012:282).



**Figure 3:** Mock-up example of a “SURGE II” clone.

The previously named game mode “Positioning” provides all necessary functionality for placing commands in the world (see figure 3), albeit the specific physical consequences of these commands need to be defined in the game module itself.

### 2.1.1 Other Game Modes

The other two game modes are constructed in a similar manner. In “Order” the player’s objective is to connect objects in the right sequence, for example combining syllables to make a whole word. In contrast “Separation” requests the player to split up existing objects and form new objects from the pieces. This could be used for teaching subjects like chemistry or engineering.

## 3. Technical Framework and Game Prototype

Following we will describe the architecture of the technical framework. We strived for an iterative development with a lean base, which could be fleshed out by each produced game module so that – over time – development of new game modules would be accelerated. How this works is described in 3.3 and 3.4 using the example of the “Crazy Robots” game module. A further point in this iterative concept was to use the rapid prototyping approach as in game design, design thinking and design based research (see e.g. Barab & Squire, 2004; Wang & Hannafin, 2005; Pernin et al., 2012) to quickly identify potential issues that need to be addressed. This stands in contrast to the Waterfall or ADDIE (analysis, design, development, implementation & evaluation) approach to software development that focuses on distinct, separated and sequential phases (Kapp, 2012:199). Especially the Waterfall Model is often connected to the low success rate (1995: 16,2%) of ICT projects (Rajlich, 2006). This proved to be a valuable approach as described in 3.5 and 4.

### 3.1 Architecture of the Framework

In its current version, the technological framework (see figure 4) is modelled closely to the hierarchical structure of the conceptual part of the game framework – supplying a class hierarchy for game modes and a base class for game objects that provides the functionality described in section 2.2.2 (Game Objects). The framework builds upon the Unity game engine 4.1.2 (Unity Technologies, 2013).

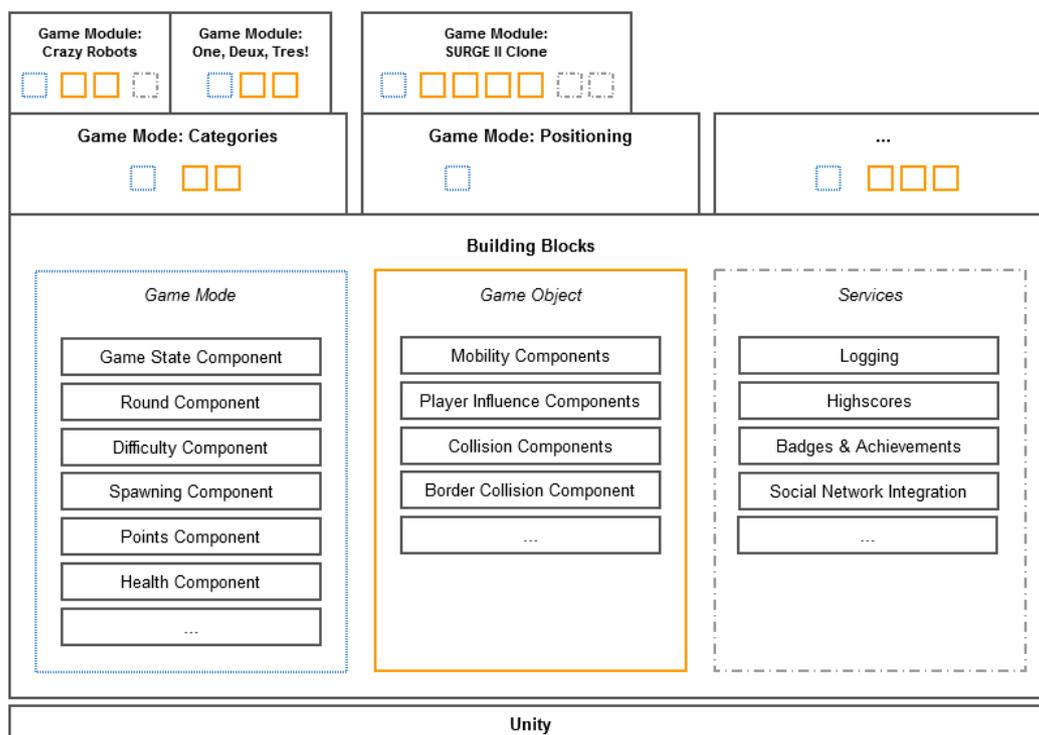


Figure 4: High level overview of the technical framework

The game modules and components can additionally make use of services like the logging system, which is going to be described in the following section. Services provide commonly used functionality, which is not tied to the core game mechanics and abstract this functionality behind coherent interfaces. Besides the “Logging” service we intend to add other reusable services like “Badges & Achievements”, “Highscore” (local and online leader boards) and “Social Network Integration”.

### 3.2 Logging & Analytics

In order to enhance our analysis of the possible learning outcomes, logging capabilities were added to the service layer of the technological framework – producing log files that can be analysed using a custom web application.

#### 3.2.1 Logging

The logging system allows the developer to create player sessions (identified by the players name and starting time of the session), log arbitrary events with custom attributes as well as a timestamp in game-time (being the time actively playing - excluding pause and menu time) and save all this session information on the device. This information enables the researcher to reconstruct the complete session of every player.

```
<message title="New Round started" timestamp="0.000000" unix_timestamp="1425379850" />
<message title="Continent Activated" timestamp="2.028221" continent="Africa" />
<message title="Country Spawned" timestamp="2.110053" country="Ethiopia" oncurrentcontinent="True" />
<message title="Country Success" timestamp="5.020000" country="Ethiopia" pointschanged="5"
currpoints="5" playertouch="False" />
```

**Figure 5:** Excerpt of a log file from a participant’s session of the “Crazy Robots” game module

The log file is saved as XML (see figure 5) whenever the game is paused or over. This is a compromise between not doing IO while the player is actively playing and saving often enough to prevent losing log data due to unforeseen software crashes. Saving the log data on the device ensures that the test participants do not need to be online while playing, but has the obvious drawback of the researcher needing access to the device after the test was conducted. Optionally sending the log files to a server once a device gets online is a feature for a future version of the logging system.

#### 3.2.2 Analytics

With XML as a standard format, the log data can easily be analysed using existing applications like Microsoft Excel or statistics software like IBM SPSS. Depending on the type of the targeted visualization it may also be feasible to create custom charts using self-written software, for example by using the statistical computing and graphics environment R (The R Foundation, 2015).

To ensure maximum flexibility when analysing the log data of “Crazy Robots” (an implementation of “Continents & Countries”, which will be described in section 3.4), we decided to develop a small web application that does all the necessary computations on the data and generates graphs using the visualization library D3.js (Bostock, 2015) and the reusable chart library C3.js (Tanaka, 2015).

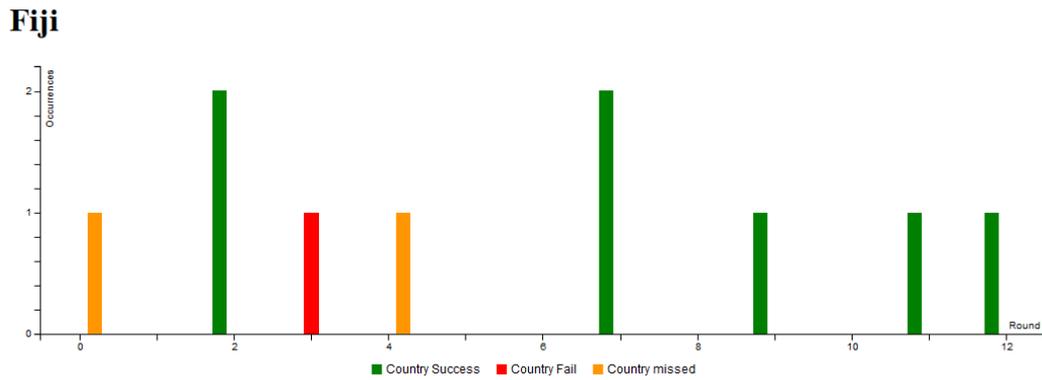


Figure 6: Visualization: participant’s correct, wrong & missed matchings over number of rounds played for Fiji

This web application is currently still strongly tied to the log events produced by “Crazy Robots”, but the idea is to abstract it in a way that it is useful for visualizing the log data of arbitrary game modules that have been created using the framework.

Besides this web application, a command-line utility that converts the XML files into segment.io events, is in the making. Segment.io (Segment, 2015) is a service that provides a unified interface to many different online analytics services. Some of these services like the open source analytics platform Piwik (Piwik, 2015) can be hosted locally, in case privacy is an issue. This will provide researchers with more tools for analysing player sessions and learning outcome.

### 3.3 How to Create new Games

To implement a new game module, a new class must be derived from a game mode or variant class, which provides most of the necessary game logic. This new class needs to deal with all the functionality that is unique to the game, such as loading the specific necessary data. The visual game objects must be implemented as Unity’s component-based game objects and use the *BaseGameObject* class, which handles movement, collisions and the like. Unity allows us to expose these game parameters to the user interface of its editor. This way, game parameters like movement speed of the game objects, spawn rate, and difficulty levels can easily be fine-tuned by the game designer. Creating a new game module is mostly a task of adjusting values in the Unity Editor (figure 7) with programming only required for module specific behaviours, which are not already provided by the existing game mode classes.

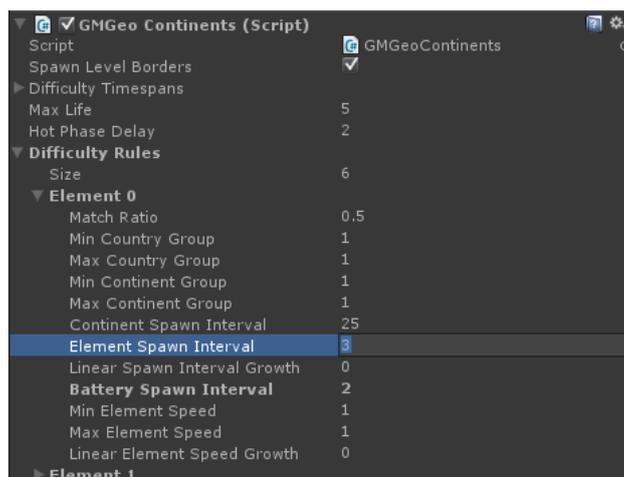


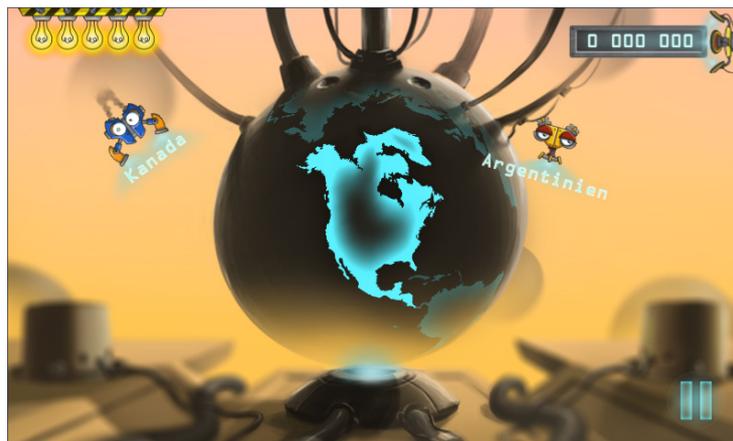
Figure 7: Properties exposed to Unity’s user interface

### 3.4 From Game Design to Game Instance: A Quick Tour

As a proof of concept we developed a prototypical version of the game module “Continents & Countries” called “Crazy Robots”, which is built upon the technical framework. The following description of the prototype underlines how the framework enables rapid prototyping and development of new game modules for a given learning goal.

#### 3.4.1 Conceptual Design

Even with a technical framework at hand, the visual appearance and story of a game module must still be crafted. However, once created, arts and animations can be reused for other game modules. With our young target group in mind, we designed a comic-based style that features animated robots (figure 8) in the spirit of WALL-E (Walt Disney Pictures & Pixar Animation Studios, 2008), with the intention of targeting both genders.



**Figure 8:** Screenshot of the game module “Crazy Robots”

#### 3.4.2 Technical Development

The creation of the game prototype started with an empty Unity scene. All the necessary elements (background, the globe with its continents, the robots) were created as Unity game objects. Some of them, like the globe, were placed in the scene while the robots were saved as prefabs (basically game object templates) that are instantiated dynamically. The robots that represent countries are assembled from different components that handle animations and gameplay (movement, collisions), the latter in form of the *BaseGameObject* class. We created the new game module’s main class (*GM\_GeoContinents*) as a subclass of the variation *GameMode\_Categories\_Comet*.

The game module’s class handles everything specific to “Continents & Countries”: loading of the country-data from a CSV file, rotating the globe, changing the active continent, and setting all parameters related to difficulty and points. The class is also connected with the prefabs of the robots so that it knows what prefab instances to instantiate. With the exception of setting the textual content to be displayed, the creation of the country game objects is handled by the game module’s super classes (*GameMode\_Categories* and *GameMode\_Categories\_Comet*).

Having created the necessary game objects and the game module’s main class, we balanced the gameplay by adjusting the game parameters in Unity’s editor. We created a set of difficulty levels with difficulty increasing over given time intervals, for instance robots will move faster and lesser known countries appear more often.

### 3.5 Transitioning to a more modular framework

The development of the “Crazy Robots” prototype proved that the framework is flexible enough to create game modules that combine the existing game object behaviours in new and interesting ways, but we found that the object oriented class hierarchy makes it harder than necessary to create new game modes that require alternate behaviours, for example means of player-game-object interactions that are currently not provided. In the next version of the framework we thus intend to make greater use of Unity’s component-

based architecture by disentangling the current functionality into even smaller single-purpose components with as little interdependency as possible. These “building blocks” (see figure 4) can then be combined to easily create new game modes, game modules and game objects.

#### **4. Expert and User Tests**

We built the “Crazy Robots” game to evaluate the ease of use and potential of both the conceptual and the technical components of the framework. Additionally we carried out heuristic usability and user tests with this prototype to identify further issues that may need to be considered for the development of the framework and the “Crazy Robots” prototype itself.

##### **4.1 Heuristic Usability Tests**

We evaluated the prototype with in-house usability and game-based learning experts (heuristic evaluation) in an iterative way to avoid problems with late coming but inevitable changes, which may present serious problems in Waterfall/ADDIE approaches (Kapp, 2012:199). Jeffries and Desurvire (1992) argue that heuristic evaluation is a valuable method, but is no substitute for usability testing with the target group (see 4.2). Nonetheless the “valuable inspection method” (Jeffries & Desurvire, 1992) of heuristic evaluation produced a wealth of information. Following we will highlight three exemplary findings related to the framework and prototype.

The first finding concerns Unity’s realistic simulation of game physics, which is handling movement, deflection and collision detection. Even touch events like drawing paths and deflecting game objects using a swipe gesture act as forces on these objects. Contrary to our expectations the realistic simulation of physical laws harmed the gaming experience. Therefore we adjusted the physics of the game objects to provide a smoother gameplay allowing frictionless and thus steady movement and making colliding objects deflect like billiard balls, despite their real shape. Furthermore, after an expert’s proposal, we decided to give the objects a tiny temporary boost when deflected by the player in order to provide better feedback.

Secondly, instead of the globe, the first version of the game showed only the current continent's outline, which faded over to the next continent after a period of time. However, the continent's outline proved to be hardly recognizable. The 3D globe was introduced to provide a better context for the current continent, rotating the globe instead of fading images and having the other continents still visible helped increase recognition and reaction time.

One question that was raised by an expert, but still needs to be evaluated in future user tests, is whether more information could be taught by fine-tuning the feedback’s level of detail. Currently the game only provides feedback as to whether or not a country is located on the given continent. Without altering the game-play, the game could teach more geographical information, e.g. the locations and shapes of countries, by highlighting the outlines of the country on the continent. Furthermore, it might be beneficial to give feedback, where the incorrectly associated countries are actually located.

##### **4.2 User Tests**

As Jeffries and Desurvire (1992) argue, testing software with experts is valuable but not enough for a holistic evaluation. This is even truer for learning software, as these additionally need to be tested concerning learning and transfer effects – although this is tricky at best, not only because the concept of “transfer” (Thorndike & Woodworth, 1901, Brown 1994) itself is debated (Schaffer, 2012:403; Anderson, Reder, & Simon, 1996 vs Greeno, 1997) but also because it is questionable whether “gold standard” quantitative science works in educational settings (Squire, 2011:228-234). The latter is especially a problem when using conceptually integrated games (Clark & Gaza, 2012) or applying commercial of the shelf games (Squire, 2011) as well as game-based learning approaches that focus on creating/modding games (Busch, Conrad & Steinicke, 2013) or epistemic learning (Schaffer, 2012). This is due to the fact, that these approaches require extensive time and framing. Squire (2011:229) for example proclaims that one typically needs an eight-week curriculum to find a statistically significant result – which is still burdened with identifying confounders like intra and inter group differences and dynamics, different teachers or to counter this a special moderator, etc (see also Squire, 2011:228-234).

By nature this is a slightly lesser issue with learning games that focus on teaching associations as well as facts and figures by repetition. Here a pre-test/post-test setting with a questionnaire covering the content seems like a low hanging fruit. To test the game we are currently organizing tests with target users. In accordance with (Nielsen & Landauer, 1993) we did not start with a big cohort of testers that all test the same version, but with an iterative approach, which integrates the lessons learned from tests with up to five users into a new version before starting new tests. This approach – although not without its critics – is economically sound as it is expected that especially early versions will contain bugs and issues that can be identified by most users so that using 100 users to find the same problem is a waste of resources especially because the issue might mask further problems that can only become visible after the (first) issue is fixed. Additionally we anticipated that even low hanging fruits might have thorns so we started the first user tests with the main goal not of testing the game but testing the tests and test setting.

To test for learning in a pre-test/post-test setting we used the CSV-file that contains the mapping of continents and countries as well as the attributed difficulty and probability of spawning. As this list contained 254 items (countries, islands, city-states and countries without sovereignty), we decided to reduce this to a number that would fit a somewhat cramped two-page table (110 countries). We first subtracted countries with a very low spawning probability, as these would mostly not be seen or even have a chance to be correctly associated in 30 to 60 minutes game time. Additionally we reduced the number of countries with high priorities as testing for 60 instead of 20 of these would probably not add much value. From the remainder we subtracted countries so that all continents would be presented relatively evenly into a list that was sorted by country name. Additionally we create a page that asked the test participant to name the continents on a world map, to test whether the depiction of continent image and name in the game strengthens this association.

To test the user experience (UX) and usability we decided to try the ISONORM-short (Pataki, Sachse, Prümper & Thüning, 2006) with 21 items and the User-Experience-Questionnaire (Laugwitz, Held & Schrepp, 2008) with 26 items both with a seven point Likert scale. While the latter uses only polar word pairs (e.g. “conservative” <- > “innovative”) and thus fits on one page, the ISONORM questionnaire uses a sentence and its negation instead and thus fills three plus two explanatory pages. The standard UX questionnaires are flanked by a one-page questionnaire that asks for some biographical information, perceived difficulty and likeability as well as some questions for a short semi-structured interview.

Up to now we did two user tests both with the sequence of pre-test, some 25 minutes game play, followed by usability/UX tests and the first post-test. Then two days of free play followed by a second post-test two days after the last play session. Results seem positive as can also be seen on the basis of the selected data in table 2 and the chart from figure 9 that was created by our analytics tool. The chart shows the success/fail events for the country of Samoa, which was answered wrongly in the pre-test, but correctly in the post-test.

	pre-test correct	post-test 1 correct	post-test 2 correct	play time	saw grade improvement potential	Liked playing
User 1	15 / 110	24 / 110	27 / 110	27m up-to 40m	4/5	4/5
User 2	74 / 110	80 / 110	94 / 110	3,2h	4/5	4/5

**Table 2:** first test results of the “Continents & Countries” game “Crazy Robots”

## Samoa



Figure 9: Success and fail events for the country of Samoa over time (user 2)

Nonetheless two problems of the transfer evaluation were identified. Firstly as the 110 items are only a subset of the country list in the game, users encountered countries in the game that are not listed in the questionnaire and vice versa. Thus learning gains might very well be higher than measured. This is especially true for the first user as she had a very low level of knowledge and might have had high learning gains for the often-appearing simple countries that have mostly been eliminated from the questionnaire. On the other hand, even the 110 items were found to be exhausting and time consuming thus a further reduction seems to be needed. An additional problem with the list-/table-like presentation might skew results potentially both to the positive and negative. At least with user 2 a negative influence was observed. When she reached the African country “Mauretania” she saw that the following item would be “Macedonia”, audibly told herself that “Macedonia” would be in Europe and then marked both countries as “in Europe”. A similar error must have occurred for the country of “Tunisia”. Although it was successfully associated to “Africa” most of the playing time (see figure 10), user 2 answered it wrongly in the second post-test. Upon inquiry one week after the test user 2 instantaneously answered correctly.

Both errors seem to be induced by the long list and the visibility of following items. The latter might be countered by an item-by-item approach. Using the mobile device and a questionnaire that presents items sequentially could solve this problem. Thus a further service building block for the mobile learning games framework could prove valuable especially as this could use data from the logging component to tailor the post-test to encountered items, potentially reducing errors that are due to huge number of questionnaire items.

## Tunisia

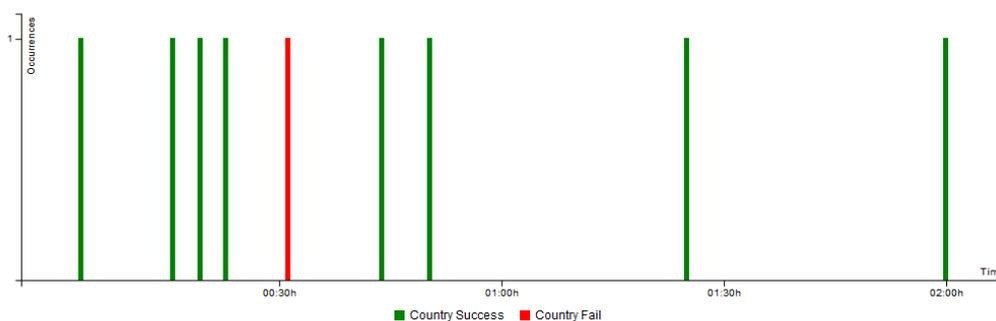


Figure 10: Success and fail events for the country of Tunisia over time

A further issue we identified in the first user test was a potential bug in the logging system. User 1 played some 27 minutes in the first accompanied play session and declared that she played the game for at least 5 minutes on both following days. But the game log-file covered only a playtime of 27 minutes. When asked whether she contrary to the agreement did not play the game in the two days of free play both she and her parents stated that she indeed had played the game. Thus we reworked the logging system and are currently testing for logging leaks.

The ISONORM and User-Experience-Questionnaire results are positive up to very positive but also show some issues. While the former is an inventory that focuses on usability for generic software systems and thus contains questions that both users found inapt – especially IK1-3 – the latter shows a strong tendency of medium values for dimensional labels with complicated adjectives – e.g. “erwartungskonform” which is a very technical term for “conformity with user expectations”. Thus both questionnaires need to be reworked for our target group.

### 4.3 Future Testing

The iterative small sample testing – even though it clearly does not generate representative data – has proven very valuable and will be continued. Additionally we are preparing a quasi-experimental follow up trail using the “Crazy Robots” prototype in a school setting with 5<sup>th</sup> and 6<sup>th</sup> graders from a Berlin elementary school. While the latter is not a perfect match with our original target group of children in private tutoring sessions as well as the contained content of continents and countries which is 7<sup>th</sup> to 8<sup>th</sup> grade in the Berlin curriculum (LISUM, 2006:12), we nonetheless hope to gain valuable data concerning usability and learning gains.

## 5. Conclusion

In this paper we described our approach to developing a mobile learning games framework in an iterative fashion. We first developed the conceptual part of the framework that covers general interaction modes for touch-based games. This enabled us to lay the groundwork for the technical part of the framework that is based on the Unity game engine and additionally enables game instances to reuse existing services, game mechanics and objects. The iterative and cyclic approach thus enriches the framework with each produced game, which in turn accelerates the development of further games – closing the loop. While the framework enables both the creation of conceptually integrated and conceptually embedded games (Clark & Gaza, 2012), we focused on the latter to produce a mobile learning game that fits our current context of private tutoring. The prototype “Crazy Robots” – a geography game that trains the association of countries to continents – was developed in an iterative fashion, too. Experience from both heuristic evaluation by usability experts as well as user testing for usability and transfer lead to the evolution of the game, its tests and in consequence of the framework. Next steps will be a quasi-experimental trial of the game in an elementary school as well as further development of game modules and thus the enrichment of the mobile learning games framework.

## References

- Anderson, J. R., Reder, L. M., & Simon, H. A. (1996). Situated learning and education. *Educational researcher*, 25(4), 5-11.
- Bandura, A. (1997). *Self-efficacy: The exercise of control*. New York: Freeman.
- Barab, S. & Squire, K. (2004). Design-Based Research: Putting a Stake in the Ground. *Journal of the Learning Sciences* Volume 13, Issue 1, 2004, S.1-16
- Bostock, M. (2015). D3.js - Data-Driven Documents. Retrieved 12.03.2015 from <http://d3js.org>
- Busch, C.; Conrad, F. & Steinicke, M. (2012). Digital Games and the Hero's Journey in Change and Innovation Management Workshops. *Proceedings 6th European Conference on Games Based Learning*, pp.72-81
- Busch, C.; Conrad, F. & Steinicke, M. (2013). Digital Games and the Hero's Journey in Management Workshops and Tertiary Education. *Electronic Journal of e-Learning*, 11(1).
- Bray, Mark (2011). *The Challenge of Shadow Education: Private Tutoring and Its Implications for Policy Makers in European Union. NESSE and the European Commission*. Available online: <http://www.nesse.fr/nesse/activities/reports/activities/reports/the-challenge-of-shadow-education-1>
- Brown, A.L. (1994). The Advancement of Learning. *Educational Researcher*, Vol. 23, No. 8 (Nov., 1994), pp. 4-12
- Clark, D.B. & Martinez-Garza, M. (2012). Prediction and Explanation as Design Mechanics in Conceptually Integrated Digital Games to Help Players Articulate the Tacit Understandings They Build Through Game Play. in Steinkuehler, C.; Squire, K. & Barab, S. (2012) *Games, learning, and society: Learning and meaning in the digital age*. Cambridge: Cambridge University Press. S. 279 - 305
- Csikszentmihalyi, M. (2010). *Flow: Das Geheimnis des Glücks*. Klett-Cotta; Edition:15.
- Egenfeldt-Nielsen, S. (2007). Third Generation Educational Use of Computer Games. In *Journal of Educational Multimedia and Hypermedia*; 2007; 16(3); Research Library pg. 263
- Entrich, S.R. (2014). German and Japanese Education in the Shadow – Do Out-of-School Lessons really contribute to Class Reproduction? *IAFOR Journal of Education*, Vol 2.2. p. 17-53
- Greeno, J. G. (1997). On claims that answer the wrong questions. *Educational researcher*, 26(1), 5-17.
- Feierabend, S.; Plankenhorn, T. & Rathgeb, T. (2014). *JIM 2014: Jugend, Information, (Multi-) Media – Basisuntersuchung zum Medienumgang 12- bis 19-Jähriger in Deutschland*. Stuttgart: Medienpädagogischer Forschungsverbund Südwest

- Jeffries, R. & Desurvire, H. (1992). Usability Testing vs. Heuristic Evaluation: Was there a contest? In SIGCHI Bulletin; 24(4)
- Kapp, K.M. (2012). *The Gamification of Learning and Instruction – game-based methods and strategies for training and education*. San Francisco: CA: Pfeiffer, John Wiley & Sons, Inc.
- Klemm, K. & Klemm, A. (2010). *Ausgaben für Nachhilfe – teuer und unfairer Ausgleich für fehlende individuelle Förderung*. Güterloh: Bertelsmann Stiftung
- LISUM – Berliner Landesinstitut für Schule und Medien (2006). *Geografie - Berliner Rahmenlehrplan für die Sekundarstufe I*. Senatsverwaltung für Bildung, Jugend und Sport Berlin (Hrsg.), Retrieved 01.03.2015 from [https://www.berlin.de/imperia/md/content/sen-bildung/schulorganisation/lehrplaene/sek1\\_geografie.pdf](https://www.berlin.de/imperia/md/content/sen-bildung/schulorganisation/lehrplaene/sek1_geografie.pdf)
- Laugwitz, B., Held, T., & Schrepp, M. (2008). Construction and evaluation of a user experience questionnaire. In A. Holzinger (Edit.), *USAB 2008, LNCS 5298* (p. 125-134). Springer
- Nielsen, J. & Landauer, T.K. (1993) A Mathematical Model of the Finding of Usability Problems. *Proceedings of the ACM INTERCHI '93*. Addison-Wesley. Pp. 206-213
- No Monkeys (2010). *Super 7*. [Download] iPhone, iPad, iPod, iOS 4.3+
- Park, Y. (2011). A pedagogical framework for mobile learning: Categorizing educational applications of mobile technologies into four types. *The International Review of Research in Open and Distance Learning*, 12(2), 78-102.
- Pataki, K.; Sachse, K.; Prümper, J.; Thüring, M. (2006). ISONORM 9241/10-S: Kurzfragebogen zur Software-Evaluation. In: *Berichte über den 45. Kongress der Deutschen Gesellschaft für Psychologie*, p. 258-259, Pabst Science Publishers, Lengerich,
- Piwik (2015). Piwik Open Analytics Platform. Retrieved 12.03.2015 from <http://piwik.org/>
- Pernin, J.-P.; Michau, F.; Mandran, N. & Mariais, C. (2012). ScenLRPG, a Board Game for the Collaborative Design of Gbl Scenarios: Qualitative Analysis of an Experiment. *Proceedings of the 6th European Conference on Game-based Learning (ECGBL)*
- Petty, M. D., Kim, J., Barbosa, S. E., & Pyun, J. J. (2014). *Software Frameworks for Model Composition. Modelling and Simulation in Engineering, 2014*.
- Rajlich, V. (2006). Changing the Paradigm of Software Engineering. *Communications of the ACM*. August 2006/Vol. 49, No. 8, 67-70
- Ryan, R.M. & Deci, E.L. (2000). *Intrinsic and Extrinsic Motivations: Classic Definitions and New Directions*. University of Rochester. Retrieved 01.02.2014 from <http://mmrg.pbworks.com/f/Ryan,+Deci+00.pdf>
- Schaffer, D.W. (2012). *Models of Situated Action: Computer Games and the Problem of Transfer*. In: Steinkuehler, C; Squire, K. & Barab, S. (2012) *Games, Learning, and Society: Learning And Meaning In The Digital Age*. Cambridge University Press. S. 403-431
- Segment (2015). *Segment - We make customer data simple*. Retrieved 12.03.2015 from <https://segment.com/>
- Stiftung Warentest (2006). *Nachhilfe: Büffeln für bessere Noten*. Stiftung Warentest 2006(4). p. 80-84
- Squire, K. (2011). *Video Games And Learning – Teaching and Participatory Culture in the Digital Age*, Teachers College Press
- Tanaka, M. (2015). *C3.js | D3-based reusable chart library*. Retrieved 12.03.2015 from <http://c3js.org/>
- The R Foundation (2015). *The R Project for Statistical Computing*. Retrieved 12.03.2015 from <http://www.r-project.org/>
- Thorndike, E. L. & Woodworth, R. S. (1901). The Influence of Improvement in One Mental Function Upon the Efficiency of Other Functions (I). *Psychological Review*, 8, 247-261.
- Tucker, B. (2012). The flipped classroom. *Education Next*, 12(1), 82-83.
- Unity Technologies (2013). *Unity3D Game Engine Version*. Retrieved 05.06.2013 from <http://unity3d.com/>
- Walt Disney Pictures & Pixar Animation Studios (2008). *WALL-E*. (Movie). Distributed by Walt Disney Studios Motion Pictures
- Yusoff, A., Crowder, R., Gilbert, L., & Wills, G. (2009, July). A conceptual framework for serious games. In *Advanced Learning Technologies, 2009. ICALT 2009. 9th IEEE International Conference on Advanced Learning Technologies* (pp. 21-23). IEEE.