

**Player Evaluation of the Level Design  
of a Platform Game Using Automated  
Questions Based on Context-Specific  
Game Metrics**

ANGELIKA BUGL

MASTERARBEIT

eingereicht am  
Fachhochschul-Masterstudiengang

INTERACTIVE MEDIA

in Hagenberg

im September 2014

© Copyright 2014 Angelika Bugl

This work is published under the conditions of the *Creative Commons License Attribution–NonCommercial–NoDerivatives* (CC BY-NC-ND)—see <http://creativecommons.org/licenses/by-nc-nd/3.0/>.

# Declaration

I hereby declare and confirm that this thesis is entirely the result of my own original work. Where other sources of information have been used, they have been indicated as such and properly acknowledged. I further declare that this or similar work has not been submitted for credit elsewhere.

Hagenberg, September 29, 2014

Angelika Bugl

# Contents

|  |             |
|--|-------------|
| <b>Declaration</b>   | <b>iii</b>  |
| <b>Acknowledgements</b>  | <b>vii</b>  |
| <b>Abstract</b>  | <b>viii</b> |
| <b>Kurzfassung</b>   | <b>ix</b>   |
| <b>1 Introduction</b>  | <b>1</b>    |
| 1.1 Motivation . . . . .   | 1           |
| 1.2 Objective . . . . .  | 1           |
| 1.3 Structure . . . . .  | 2           |
| <b>2 Platform Games</b>  | <b>3</b>    |
| 2.1 Definition . . . . .   | 3           |
| 2.2 Components . . . . .   | 6           |
| <b>3 Reasons for Evaluations</b>                                   | <b>10</b>   |
| 3.1 The Perspective of the Gaming Industry . . . . .               | 10          |
| 3.2 Experience . . . . .   | 10          |
| 3.3 Flow and Immersion . . . . .                                   | 12          |
| 3.3.1 Flow . . . . .   | 13          |
| 3.3.2 Immersion . . . . .  | 14          |
| 3.4 Problems in Games . . . . .                                    | 16          |
| 3.4.1 Problems in Platform Games . . . . .                         | 17          |
| 3.5 Conclusion . . . . .   | 18          |
| <b>4 Game Evaluation Methods</b>                                   | <b>19</b>   |
| 4.1 Evaluation of Traditional Products . . . . .                   | 19          |
| 4.1.1 Differences between Traditional Products and Games . . . . . | 19          |
| 4.1.2 Classic Evaluation Methods . . . . .                         | 20          |
| 4.2 Questionnaires . . . . .                                       | 22          |
| 4.3 Heuristics . . . . .   | 23          |
| 4.4 Gameplay Metrics . . . . .                                     | 24          |

|          |  |           |
|----------|--|-----------|
| 4.4.1    | Data for Analytics . . . . .   | 25        |
| 4.4.2    | From Data to Metrics . . . . .   | 26        |
| 4.4.3    | Classification of Metrics . . . . .  | 27        |
| 4.4.4    | Gameplay Metrics for Platform Games . . . . .                                | 28        |
| 4.4.5    | Feature Selection . . . . .  | 28        |
| 4.4.6    | Tracking Strategies . . . . .  | 29        |
| 4.4.7    | Analysis . . . . .   | 30        |
| 4.5      | Combining Qualitative and Quantitative . . . . .                             | 30        |
| <b>5</b> | <b>State of the Art</b>  | <b>32</b> |
| 5.1      | TRUE . . . . .   | 32        |
| 5.1.1    | Case Study: Halo 2 . . . . .   | 33        |
| 5.1.2    | Case Study: Halo 3 . . . . .   | 34        |
| 5.1.3    | Case Study: Shadowrun . . . . .  | 35        |
| 5.2      | EIDOS Metrics Suite . . . . .  | 36        |
| 5.2.1    | Tomb Raider: Underworld: Clustering into Different<br>Player Types . . . . . | 37        |
| 5.2.2    | Fragile Alliance . . . . .   | 38        |
| 5.3      | Volition's Telemetry System . . . . .  | 39        |
| 5.4      | More Studies . . . . .   | 41        |
| 5.5      | Conclusion . . . . .   | 41        |
| <b>6</b> | <b>Project</b>   | <b>43</b> |
| 6.1      | Concept . . . . .  | 43        |
| 6.1.1    | Problem, Idea and Requirements . . . . .                                     | 43        |
| 6.1.2    | Solution . . . . .   | 44        |
| 6.2      | Architecture and Implementation . . . . .                                    | 44        |
| 6.2.1    | xis-engine . . . . .   | 44        |
| 6.2.2    | Components . . . . .   | 45        |
| 6.3      | The Game: Elements . . . . .   | 51        |
| 6.4      | Integration of the System into Elements . . . . .                            | 53        |
| 6.4.1    | Objective . . . . .  | 53        |
| 6.4.2    | Architecture . . . . .   | 54        |
| 6.4.3    | Tracked Features . . . . .   | 57        |
| 6.4.4    | Asked Questions . . . . .  | 58        |
| 6.5      | Conclusion . . . . .   | 59        |
| <b>7</b> | <b>User Study</b>  | <b>60</b> |
| 7.1      | Evaluation Design . . . . .  | 60        |
| 7.1.1    | Test Levels . . . . .  | 60        |
| 7.1.2    | Starting the Test Session . . . . .  | 61        |
| 7.1.3    | Menu Options . . . . .   | 61        |
| 7.1.4    | Triggering of Questions . . . . .  | 62        |
| 7.1.5    | Types of Questions . . . . .   | 65        |

|                   |  |            |
|-------------------|--|------------|
| 7.2               | Preliminary Study . . . . .                              | 67         |
| 7.2.1             | GameStage . . . . .                                      | 67         |
| 7.2.2             | Advantages of the Setting . . . . .                      | 67         |
| 7.2.3             | Problems with the Setting . . . . .                      | 68         |
| 7.2.4             | Detected Problems in the Tool . . . . .                  | 69         |
| 7.2.5             | Execution . . . . .                                      | 69         |
| 7.2.6             | Further Improvement Opportunities for the Tool . . . . . | 69         |
| 7.2.7             | Data Cleaning . . . . .                                  | 72         |
| 7.3               | Main Study . . . . .                                     | 72         |
| 7.3.1             | Changes in the Questions . . . . .                       | 73         |
| 7.3.2             | Setting . . . . .  | 74         |
| 7.4               | Lessons Learned . . . . .                                | 75         |
| 7.5               | Conclusion . . . . .                                     | 75         |
| <b>8</b>          | <b>Evaluation</b> . . . . .                              | <b>76</b>  |
| 8.1               | Results of the User Study . . . . .                      | 76         |
| 8.1.1             | General . . . . .  | 76         |
| 8.1.2             | Test Session Aborted and Level Skipped . . . . .         | 77         |
| 8.1.3             | Difficulty . . . . .                                     | 79         |
| 8.1.4             | Level Popularity . . . . .                               | 83         |
| 8.1.5             | Coins . . . . .  | 87         |
| 8.1.6             | Controls . . . . .                                       | 88         |
| 8.1.7             | Understanding . . . . .                                  | 88         |
| 8.1.8             | Heatmaps and Lines . . . . .                             | 90         |
| 8.1.9             | Predominance of Fire . . . . .                           | 92         |
| 8.1.10            | Feedback . . . . .                                       | 93         |
| 8.1.11            | Summary . . . . .  | 94         |
| 8.2               | Further Ideas . . . . .                                  | 95         |
| 8.2.1             | Analysis . . . . .                                       | 95         |
| 8.2.2             | Tracking . . . . .                                       | 95         |
| 8.3               | Summary . . . . .  | 96         |
| <b>9</b>          | <b>Conclusion</b> . . . . .                              | <b>97</b>  |
| <b>A</b>          | <b>Content of the CD-ROM</b> . . . . .                   | <b>98</b>  |
| A.1               | PDF-File . . . . .                                       | 98         |
| A.2               | Study Results . . . . .                                  | 98         |
| A.3               | Miscellaneous . . . . .                                  | 99         |
| <b>References</b> |  | <b>100</b> |
|                   | Literature . . . . .                                     | 100        |
|                   | Online sources . . . . .                                 | 108        |

# Acknowledgements

I would like to thank my supervisor, Jeremiah Diephuis, who supported me in every single aspect concerning this thesis. I am grateful for the enormous amount of time he invested into providing me detailed feedback and explanations as well as helping me with my user studies.

I also want to thank my parents who enabled me to study and who always encouraged me in my work. Moreover, I want to thank all my relatives for helping me at anytime and in any situation when I need them.

Finally, gratitude is owed to my boyfriend, Christoph Lipphart, for his supportive advice, his patience and for being always there for me.

# Abstract

This thesis discusses the evaluation of the level design of platform games using game metrics in combination with questionnaires. Thereby the tracked metrics data are analyzed directly in the game at run-time and used to ask appropriate questions depending on the current game situation. This approach aims to identify problems with the level design. It is examined which questions may be useful and which events could trigger their appearance. The developed system was included into the platform game *Elements* to facilitate an evaluation. It was tested in a user study to discover if it is actually able to detect problems and provide useful results for improving the game.



# Kurzfassung

Diese Arbeit beschäftigt sich mit der Evaluierung des Level Designs von Plattformer-Spielen unter Verwendung von *Game Metrics*, welche mit Fragebögen kombiniert werden. Die gesammelten Daten werden dabei direkt im Spiel zur Laufzeit analysiert und dazu benutzt, um abhängig von der jeweiligen Spielsituation passende Fragen zu stellen. Diese Methode zielt darauf ab, Probleme im Level Design zu finden. Es wird darauf eingegangen, welche Fragen dafür nützlich sein könnten und aufgrund welcher Ereignisse diese gestellt werden können. Um eine Evaluierung zu ermöglichen, wurde das entwickelte System in das Spiel *Elements* integriert. In einer Nutzerstudie wurde getestet, ob dieses Tool tatsächlich Probleme entdecken und für die Verbesserung des Spiels nützliche Ergebnisse liefern kann.

# Chapter 1

## Introduction

### 1.1 Motivation

As the video game industry is constantly growing, the importance of game evaluation methods is increasing. Big game studios usually have their own dedicated test departments to assure high quality and good player experiences. The systems used to serve this purpose commonly include game metrics tracking. This method makes it possible to automatically collect a multitude of various information such as the amount of time spent for a specific task, the weapons used or the areas visited. This allows detailed insight into the happenings during gameplay and the decisions players make.

But this approach is not able to report information about the intentions and feelings of users. A possible method for obtaining such data is directly asking the users themselves using questionnaires.

### 1.2 Objective

The intention of this thesis was to develop a system which combines metrics tracking with in-game questionnaires to maximize the outcome and the usefulness of the evaluation. In addition, the tool includes the possibility of an immediate in-game analysis. This enables it to ask appropriate questions depending on the tracked metrics data and gain deeper insight into how players experience particular situations and what caused major difficulties.

This thesis tries to identify if such a system can be used to detect problems in platform games especially concerning their level design. It was implemented as a **service** for the *xis-engine* and integrated into the previously developed platform game *Elements*.

It was tested using two user studies. The preliminary study was dedicated to finding problems concerning the program itself while the results from the main study were analyzed in detail and offered interesting and valuable information which can be used for improving the game and its levels.

### 1.3 Structure

To start with, the platform game genre is introduced. The main components used for the level design of such games as well as some of their numerous variation opportunities are briefly described.

Subsequently, it is discussed why it is important to care about evaluations and why they form a useful possibility for the video game industry. In this context, the importance of the experience a game provides is emphasized. Two related concepts, *flow* and *immersion*, are introduced and some problems which can occur in computer games are listed to further highlight the relevance of evaluation methods.

The next chapter deals with various possibilities game developers can use to test their games. In addition to several traditional evaluation methods, such as direct observation and thinking-aloud, questionnaires and heuristics are also described and it is especially focused on gameplay metrics.

Subsequently, three evaluation systems which also have some similarities to the system proposed in this thesis, are presented. Three exemplary case studies highlight their potential and usefulness for the evaluation of video games.

In the practical part, the developed evaluation prototype and its architecture are described. Particular emphasis is placed on the integration of the system into the platform game *Elements*.

Chapter 7 deals in detail with the two performed user studies. It presents the used evaluation design and explains the test flow. In addition, the problems which occurred during the preliminary user study concerning the prototype are listed and it is illustrated how they were fixed in order to increase the quality for the main study.

Several results of the main user study are presented and analyzed to provide an insight into the possibilities the collected data offer and to highlight the potential of this particular evaluation method. Several ideas for improving the game, which were derived from the results, are proposed. In addition, further tracking and analysis prospects are mentioned.

Finally, a conclusion summarizes this work and presents some possibilities for further enhancements of the evaluation system.

## Chapter 2

# Platform Games

This chapter aims to explain the main characteristics of platform games<sup>1</sup>. Several examples of platformers are mentioned and components which can frequently be found in games belonging to this genre are described.

### 2.1 Definition

Currently there is not *one* generally accepted genre taxonomy, but several very different theories have been proposed. Some of them do not list platform games as separate genre. As a result there is also not *one* clear definition of this game genre.

In [70, p. 9] Rogers defines the platformer genre as a subgenre of action games:

A platform game often features a mascot character jumping (or swinging or bouncing) through challenging “platform” environments. Shooting and fighting may also be involved.

A very similar definition can be found in [93]:

A platformer is a game in which a character runs and jumps around a level consisting of platforms floating in the air.

In [96] this game genre is briefly defined as:

A platformer (sic!) is a video game in which the game-play revolves heavily around players running and jumping onto platforms, floors, ledges, stairs or other objects depicted on a single or scrolling game screen.

A more comprehensive description can be found in [82, p. 270]:

---

<sup>1</sup> In German usually the term *Jump 'n' Run* (from jump and run) is used instead of platformer or platform game [95].

Games in which the primary objective requires movement through a series of levels, by way of running, climbing, jumping, and other means of locomotion. Characters and settings are seen in side view as opposed to top view, thus creating a graphical sense of “up” and “down” as is implied in “Platform.” These games often also can involve the avoidance of dropped or falling objects, conflict with (or navigation around) computer-controlled characters, and often some character, object, or reward at the top of the climb which provides narrative motivation. This term should not be used for games which do not involve ascending heights or advancement through a series of levels (see Adventure), nor for games which involve little more than traversing a path of obstacles (see Obstacle Course).

Although there are several differences between these definitions, they all include the basic mechanic of moving across platforms. It may also be added that these games are frequently clustered into smaller parts, called levels, and that most games have one starting point and one end point. According to [70, 82], platformers are closely connected to adventure games.

Within this genre there is much freedom for variations. For example, the type of movement can differ; in addition to jumping and running the character may, for example, also swing, slide, climb or fly (see for instance *Little Big Planet*, *Rayman Origins* or *FEZ*). The goal may be completing the level as fast as possible (for example in *Sonic the Hedgehog*) or exploring it and collecting rewards (for instance in *Donkey Kong Country 2*). There may be multiple paths (e.g. *Sonic*) or just a single one (e.g. *Super Mario World*) [77].

Furthermore, developers can use different perspectives and degrees of freedom [95]. In *Super Mario Brothers*, a typical side perspective was used. Very similar are 2.5D games in which the game world consists of 3D objects while the physics are calculated in 2D. The camera can use the side perspective such as it was done in the mobile application *Manu Ganu* (see figure 2.1). But platformers can also be in 3D; there the player can move in a three-dimensional space as this is for instance the case in *Super Mario 64*.

There are games which handle the number of dimensions in a very innovative way: In *Little Big Planet* there is, for example, a third dimension which is split into three different layers (see figure 2.2). The players can always decide, depending on the objects in each layer, which one to use. Another innovative game is *FEZ* in which the game world is always seen from one (of four) side perspectives. The main character itself only runs in one 2D perspective. The player can change the 2D perspective by pressing a button which rotates the 3D world around 90°.

In many platform games, the running direction is from left to right [95] but there are also games in which it changes somewhere within a level as is



**Figure 2.1:** This screenshot is taken from the 2.5D platform game *Manu Ganu* [101].

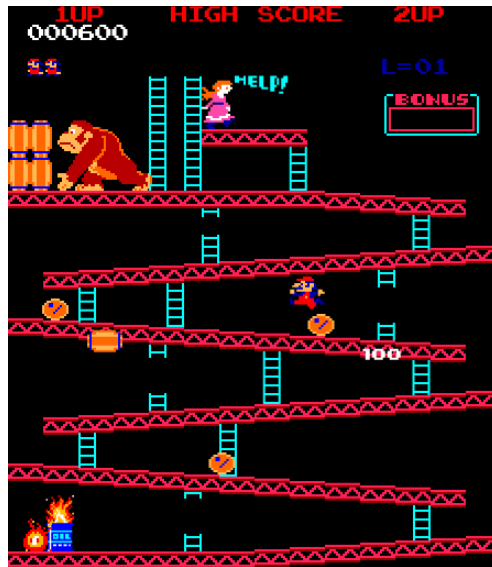


**Figure 2.2:** In the game *Little Big Planet*, the character can move on three different depth layers.

the case in *Manu Ganu*.

One of the first platform games was Nintendo's *Donkey Kong* (1981) (see figure 2.3). It was created by Shigeru Miyamoto who subsequently also created the famous *Mario* titles<sup>2</sup> [70, pp. 5, 28][53, p. 26][64, p. 12].

<sup>2</sup>According to [82, p. 275], *Super Mario Bros.* was the best-selling home video game of all time. This may be also because it was sold in combination with the *Nintendo Entertainment System*.



**Figure 2.3:** This image shows a screenshot of Nintendo's *Donkey Kong* (1981) which was one of the first platform games.

## 2.2 Components

Although platform games can vary in many aspects, some components can be found frequently within different games. The following collection is based on the categorization used in [77]. A different classification can, for example, be found in [70, pp. 69–71]. Although it is not platformer specific, it partially describes the same game elements.

In this context, several variations of object properties will be listed. Instead of explicitly providing an example game for each of them, which would have resulted in many repetitions, it was decided to list them as global references. Within the scope of this section, it is suggested to have a look at the following platform games which contain most of the addressed items: *Little Big Planet*, *Super Mario Bros.*, *Super Mario 64*, *Rayman Origins* and *Donkey Kong Country Tropical Freeze*.

### Avatar

The avatar is the main character which is controlled by the player. It can be possible to choose one out of several avatars or to change it within a level, which can extend the game by adding a puzzle component. Usually only one character is controlled by the player at any moment in time.

As already discussed, there are several different movement possibilities such as running, climbing, jumping, crouching, swimming or swinging to mention just a few. A character may be able to use several different methods,

which can also change in the course of a level. It is important to note, that in all games of this genre, players have control over the horizontal and vertical movement of their character, although the latter may be limited.

Furthermore, the character can also have additional abilities such as double jumping, wall jumps [70, p. 102][79, pp. 258–262] or the power of shooting, fighting or destroying things.

### **Platforms**

In [77] a platform is defined as any object that the avatar can walk or run across safely. Platforms can be temporary, controlled, for example, by a timer or a limited number of touches executed by the character. They may be static or moving. They can be straight, occasionally rotated, or perhaps in a limited number of angles as well as forming arbitrary curves. Varying their size by creating smaller platforms can be used to increase the challenge of a passage. Furthermore, their friction values can differ, resulting, for instance, in slippery ice worlds. They may also be flexible, bending, shifting or invisible and there are also platforms which serve multiple purposes such as being item boxes (e.g. *Super Mario World*) at the same time. It can also be possible for the character to change them by moving or destroying them.

### **Obstacles**

Obstacles can harm anytime they collide with the character. They may be static, such as spikes, or moving. It can be possible to eliminate them, but they may also respawn once, for a finite or unfinite number of times. Obstacles can also be equipped with some sort of awareness and/or artificial intelligence [70, p. 71][93] which can, for example, enable them to shoot into the direction of the avatar.

More complex enemies may also be used for possible boss battles [92][70, p. 71] at the end of a level.

### **Movement Aids**

Movement aids help the character to move through the level in an other way than jumping or running. Examples are ladders, springs, ropes or slings. They may also be moveable by the character. Unlike some power-ups they do not alter the players' powers permanently but only as long as they interact with it.

### **Collectible Items**

Collectible items such as coins, stars, points, extra lifes, weapons or power-ups are objects which offer a reward. In [77] it is claimed that every platformer has a reward system and that collectible items are in many cases



used for this purpose. Collectibles can also be non-player characters which extend the avatar's moving abilities such as enabling it to fly, for instance. These items can also be used to guide players through the level. This can, for example, be helpful when they should be encouraged to jump down when they are not able to see the ground as this uncertainty can feel unfair to users [94]. Furthermore, collecting things can also just be fun [93].

### Triggers

Interactive trigger objects can be used by the player to change parts of the level such as the physics or the opening status of a door. These effects may be only temporary; an entrance may, for example, close itself automatically briefly after the player releases the button. Triggers can be used to add a puzzle character to the game. This is, for example, possible in *Media Molecule's*<sup>3</sup> *Little Big Planet*. This game is equipped with a level editor which allows users to build their own levels using a wide range of different interactive objects and triggers. These elements may also be adjusted and combined to create complex and interesting puzzles.

### Checkpoints

Although checkpoints are not mentioned explicitly in [77], they are also used frequently in platform games. Usually they are automatically activated when the avatar touches them. Should players die after reaching a checkpoint, they do not have to restart the level from the beginning but they are respawned at the position of the last activated checkpoint. In this way, checkpoints help gamers to succeed in long and difficult levels.

The number of times how often a player can be respawned at a certain checkpoint can be limited, as is the case for some checkpoints in *Little Big Planet*.

### Other Elements

These were some of the major components used to build platformers, but there are of course more things involved in the creation of (platform) games. Other essential parts include, for example, the camera [84, 94][70, pp. 121–153], the user interface [93][70, pp. 171–196] [73, pp. 221–244], the controls [70, pp. 155–169] [73, pp. 221–244], the overall physics [92] or music [70, pp. 394–405] to mention just a few.

Furthermore, a platform level is more than just the sum of all its components. Especially the rhythm is emphasized often in the literature [14, 92, 77, 78]. Hoffstein claims that jumping puzzles are not fun per se but that they have to be well-designed. In his opinion, these feel best when they

---

<sup>3</sup>[www.mediamolecule.com](http://www.mediamolecule.com)

are built in a rhythmic sequence, such as *Jump-Jump-Run-Jump-Jump-Run*. This may also make the puzzles learnable [92] and the jumps easier to time. Furthermore, being *in the rhythm* may also help users to reach the *flow* state (see section 3.3.1) [14, 92].

After having defined what platform games are, the next step is to address evaluations. The next chapters explain why they are used in the video game industry and which possibilities developers have to playtest their games.

## Chapter 3

# Reasons for Evaluations

Game evaluations can be time consuming and require resources such as people performing them or computers to test the game (see chapter 4). Therefore, it could be questioned why they should be done and if they are worth the efforts. This chapter tries to explain why evaluations are meaningful and necessary in the game development process. Furthermore, some problems which can affect the quality of digital games will be mentioned.

### 3.1 The Perspective of the Gaming Industry

The growth of the market of digital games is remarkably constant and games are becoming an important part of the mainstream software development industry [27, 33] and a very popular spare-time activity [60]. Thus, also the competitive pressure is heightened and the requirement for high quality, interesting content, innovations and an immersive experience increases [27, 33, 58, 60]. To achieve these goals, game evaluations and testing, which have already been performed for decades, are essential [59]. The scientific methods for measuring and analyzing player experiences have further become very important [59] as digital games become more complex [20, 45, 55].

### 3.2 Experience

When asking why it makes sense to evaluate a game, it is also important to know why a game is created per se. What is the goal of the game development process and what do designers want to achieve with their creation?

Video games should create a good user experience [88], entertain and engage the players [66, 67] and they have to be fun [46, p. 66][45]. The game's quality is directly linked to the perceived experience [20].

Computer games *always* create an experience regardless if this is intended or not [65]. Playing a game means *experiencing* it, including for instance visual, auditory and tangible input [71, p. 314].

In the game design context, a couple of variations of the term experience such as user experience [5, 6, 35, 58, 80], player experience [5, 59, 71], play experience [71], gameplay experience [25, 50, 58, 60, 61], game experience [50, 71, 80] or gaming experience [35, 80] show up in literature. They are difficult to separate as there is yet not *one* general definition [6, 25, 35, 80]. It is also questionable if it is even possible to clearly differentiate between them and most probably they also intersect each other. Going into details about what which term explicitly specifies would go beyond the scope of this thesis. Therefore it is not clearly distinguished between the diverging terms although different wordings may be used in different contexts to emphasize various aspects of experience.

Playtesting has to be taken very seriously as the whole experience can be destroyed by one single sub-system in the game [46, pp. 66–67]. In *Call of Duty 3*, for example, much effort was invested into the fine-tuning of bikes as they were not fun to drive because they were too slow and their handling did not feel good [46, p. 67].

Furthermore, the experience a game creates can never be completely predicted. It is necessary to test if the game is fun to play, if the gamers understand what to do and if they want to play the game again [71, p. 12]. Moreover, it may also be important to discover if the game creates the intended experience [45, 65] and how the player's experience emerges [65]. This can help to evaluate design decisions which were already made and to further improve the game design. For example, in a game in which the players have to learn and improve some skills, it may be important to test if they are able to achieve and understand these, as confusion caused by unintuitive game elements or bad design can be a problem for the game [1]. Moreover, if a game becomes too hard or too easy according to the user's skills, this can influence the *flow* state, which will be described in detail in section 3.3.1, in an undesirable manner.

There is also not only *one* experience but there are a couple of very different experiences. Games of different genres will result, for example, in different experiences [35]. Every single experience is different [86] and experiences can have many different forms [71, p. 314].

Currently there are no fixed genres and no general knowledge about what people prefer [19]. There is not *one* single experience which should be provided by every game [71, p. 314]. Furthermore, there are no reliable measurements of gaming experiences [60]. Indeed they are very difficult to measure and describe [35]. The reasons for this are that experiences are based on an unconscious process and that they are fleeting. Moreover, there is currently no common terminology to describe the concept of experiences. Nevertheless, the design of experience is a very fundamental principle for game development [71, p. 314]. Understanding the experiences of the players may help to understand how to create pleasure [19] and why people play digital games [60].

Furthermore, it is necessary to understand what happens when a video game is played and what player experiences occur in order to understand games, as the interactivity of the game is very fundamental and there is no game without a player [25].

Game experiences consist of the same parts all other experiences do. In [25] gameplay experiences are defined as “[...] an ensemble made up of the player’s sensations, thoughts, feelings, actions and meaning-making in the gameplay setting.” Thus, experience is not directly created by game elements but it evolves from the interaction between the user with the game. Therefore, the users with their individual desires, anticipations, previous experiences and abilities form a fundamental part. This also causes different people to experience the same game differently. Furthermore, also the context in which a game is played can have an impact on the experience, for example, the reason why a game is played such as avoiding boredom or to vent anger. Moreover, the experience can also be different when there are other people in the room, maybe also participating in the game [25].

### Related Concepts

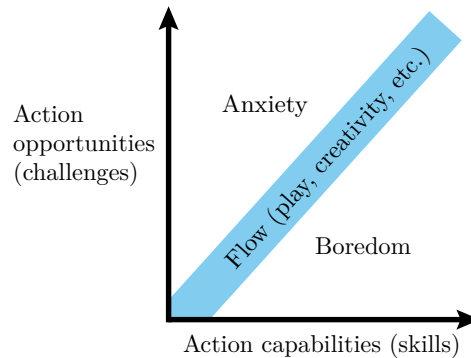
There are several concepts which are related to the topic of experience. Often the terms fun, immersion and flow are mentioned in literature [6, 19, 52, 61, 65]. Furthermore, also the terms playability [6, 56, 59], engagement [19, 35, 52, 75] and involvement [35, 98] can be found frequently. The game design and evaluation model proposed by Pereira and Roque includes the perspectives *playfulness*, *challenge*, *embodiment*, *sensemaking*, *sensoriality* and *sociability* [65]. In the following section two selected concepts, flow and immersion, are described in more detail.

## 3.3 Flow and Immersion

In [61] it is claimed that flow and immersion are seen as the holy grail of digital game design. Unfortunately, there are currently no generally accepted definitions for these terms [10, 61].

Flow and immersion are related to each other [3, 10, 35]: Both may be necessary to characterize the concept of gameplay [35]. Immersion may be required in order to experience flow [61] as in total immersion people forget about their surroundings [10, 61] whereas people in the flow state are completely absorbed by the activity [61]. But, on the other hand, flow may also be a precondition for immersion [3]. Furthermore, some components of flow and immersion are similar such as the requirement of attention, knowledge and skills as well as the losing of the sense of time and self [10].

This demonstrates that the different understandings of these concepts make it difficult to clearly separate and compare them. Nevertheless, this section tries to give a rough insight into their original ideas.



**Figure 3.1:** Csikszentmihalyi's model of the flow state [16, p. 49][62].

### 3.3.1 Flow

Csikszentmihalyi has intensively studied positive experiences [62][17, p. xi] by interviewing people such as chess players, dancers or rock climbers. He discovered that there is a state of optimal experience he called *flow* in which people feel a deep sense of enjoyment. It is a state of consciousness in which people are so concentrated that they are completely absorbed in the activity. They typically forget about time and problems and feel strong, alert, in control and unselfconscious. Through his studies, Csikszentmihalyi found out that this state is experienced the same way independent of culture, social class, age, gender and the performed activity. He mentioned some characteristics of flow: People perform a task with clear goals, rules and immediate feedback. The activity is challenging and cannot be done without skills but people are able to finish it. They feel a sense of control over what they are doing and are not concerned about losing this control. They can concentrate on this activity, are deeply involved and forget about their problems and worries as well as about themselves. They lose their sense of time: After the activity, they do not know where the time went and during it it seems that time stands still [83, 62][17, pp. 43–71].

Flow can be seen as a state of balance between skill and challenge [35] (see figure 3.1).

Some video game players may have already experienced this state [83, 35, 38]. Games can include some of the requirements of flow: For example, they may have a goal, are challenging, require skills and provide immediate feedback [25, 35]. An effective game can offer players an almost trance-like experience in which they focus only on the game and forget about everything else [35, 38]. In [15], definitions of flow were compared to concepts of play. There were some similarities found which led to the conclusion that these two systems may intersect each other and that games *can* give flow.

Challenge is very important for good game design. Ideally it adequately matches the individual skills of the player. Thereby it has to be taken into account that these skills will increase during playing, potentially increasing the degree of flow [35]. But this flow state is very fragile. If the challenges become too difficult compared to the individual skills, people may become anxious or frustrated. When the challenges become too easy, they may become bored [35, 62]. Therefore one very important challenge when designing a game is to adjust the difficulty appropriately in order to keep the players in the flow state [35].

### 3.3.2 Immersion

The term immersion is often used, but it lacks a common, generally agreed upon definition [3, 10, 25, 52]. Immersion is a powerful experience and often very important for enjoying a game. But players can also feel enjoyment when they are not immersed. Unfortunately, it is not exactly known what causes it. Game characteristics can create but also ruin immersion. This term has been used in several different contexts but most of the time it is used in relation to virtual reality and game software [3, 10]. Bartle claims that players want affirmation of identity and that immersion offers this [3].

Immersion describes how involved a player is in a game [10]. It is the sense of being in the virtual world [3]. Players are immersed when they feel like they are in the game [3, 52]. Murray defined immersion very figuratively as follows [54, p. 98]:

The experience of being transported to an elaborately simulated place is pleasurable in itself, regardless of the fantasy content. We refer to this experience as immersion. *Immersion* is a metaphorical term derived from the physical experience of being submerged in water. We seek the same feeling from a psychologically immersive experience that we do from a plunge in the ocean or swimming pool: the sensation of being surrounded by a completely other reality, as different as water is from air, that takes over all of our attention, our whole perceptual apparatus.

Immersion is related to the concept of presence [3, 10, 52] which can be defined as *the feeling of being there* [52]. Often both terms are used synonymously [52].

McMahan defined three conditions for immersion concerning virtual reality in 3D games: The expectations the player has have to match with the conventions of the virtual reality, the actions the player can take have to have a non-trivial impact on this world and the conventions have to be consistent [52].

In order to receive more information about immersion, Brown and Cairns interviewed gamers [10]. Based on their study, they suggested a division into

three different levels: *engagement*, *engrossment* and *total immersion*. They also discovered that it can be difficult to experience total immersion as there are several barriers. Some of them can be removed by the game design, others can only be removed by the player such as for instance concentration.

The lowest level of immersion is *engagement*. To enter this level, players have to play the game. Therefore having access is one barrier. The gamers also have to start the game, which depends on their individual preferences concerning, for example, game genres. The users have to spend time; someone who has already played a certain game for a very long time may become more involved into it. Another barrier is that they have to invest effort and use their energy to learn the game. They have to pay attention to the game and be willing to concentrate on it. Therefore the game should provide something which encourages people to keep on playing. When these barriers are overcome, players start feeling engaged.

The next level is *engrossment*. One of the barriers therefore is that during playing, the emotions of the users have to be affected. In the study, it was discovered that game construction was important: Components, such as tasks, visuals and plots, when well designed, are able to support such feelings. When in engrossment, players partly forget about the surrounding area and want to keep on playing as they have already invested a considerable amount of time and effort.

The final stage is *total immersion*. Brown and Cairns equated this level of immersion with presence. People completely forget about reality. The only thing that matters is the game. Participants of the study described their feelings as *being in the game*. One problem is that total immersion is elusive. One of the barriers to this experience is empathy; the players have to feel attached to the game character. Atmosphere is another very important aspect to enter this level of immersion. Atmosphere is, as engrossment, created using game construction. But in this case they also have to be relevant to the game. When it is necessary to carefully perceive every single piece of the game, more effort and attention is required which leads to deeper immersion [10].

Bartle also proposes different levels of immersion [3]. He separates them into *unimmersed*, *avatar*, *character* and *persona*. They can be differentiated using the way the players see the main character. If they regard it as just an object, they are unimmersed. When it is seen as their representative, it is an avatar. When they start projecting their personality onto it, it is called character and the last level is when they consider it as being themselves. In this case it is their persona [3].

After having illustrated the concept of experience and its importance in video games, the following section will go one step further and explain some problems in games which can lead to poor gameplay experiences.



### 3.4 Problems in Games

When testing games, it can be helpful to know which problems may occur to focus on these aspects for the purpose of maximizing the evaluation's outcome. One very valuable resource for this task are heuristic sets as they contain collections of problems. They will be explained in detail in section 4.3.

The following heuristics are extracted from [18, 26, 39, 40, 42, 45, 66]. The problems listed are derived from [66] in which a heuristic evaluation of 108 different games was performed.

**Interface:** The game's interface and menu should be consistent, intuitive, logical as well as easy and efficient to use. It should be considered as part of the game. This also includes the navigation of the menu. Furthermore, the terminology used should be understood by the player.

It can be problematic when there is too much information, too many characters or game elements on the screen or when it is difficult to differentiate between interactive and non-interactive elements [66]. A poor user interface can completely destroy a gaming experience [45].

**Goals:** The goals in the game should be clear, meaning that the players can easily understand and identify them. It should also be possible to create one's own goals. Moreover, the goals should be reachable.

**Help:** The game should also provide context-sensitive help to avoid that users might get stuck. They should not be forced to read the game's manual. Example problems are missing instructions, tutorials and training missions.

**Control:** Players should feel in control. The controls should be convenient, intuitive, consistent and flexible and they should follow standard conventions. The inputs should be easy to manage and should respond appropriately. Problems are, for example, poor hit detection, poor physics, bad input mappings or inconsistent input responses. Furthermore, slow response time, oversensitive, unresponsive and unnatural controls can (unintentionally) increase the game's difficulty level.

**Challenge:** The game should be easy to play at the beginning. Later on, the difficulty may increase, but the learning curve should be shortened. At the beginning of the game, tutorials should help the gamers to make it possible for them to start playing immediately. Strategy, challenge and pace should be balanced. Pace can apply pressure but players should not be frustrated or bored at anytime. Challenges should be experienced as positive

rather than negative experiences. Furthermore, variable difficulty levels are recommended.

**Consistency:** The game, its elements and the story should be consistent and predictable. When a player moves, the world should react accordingly and the generated changes should also be persistent.

**Understanding:** The players should understand what is happening in the game, the failure conditions, their current status and it should be clear to them what the next goal is. It should be possible to easily identify game elements. These should *look like what they are for*; an enemy should, for example, not be misinterpreted as a power-up. It is a problem when a player's confusion is caused by poor game design [1].

These were just a few excerpts which can help increasing a game's user experience. Most of them are very general and can therefore be applied to games of very different genres. Finding lists of problems for games of a specific genre is currently very difficult. But the usefulness of such collections can also be questioned, as games, even though they belong to one genre, can differ in many aspects [13] as every game may have a USP<sup>1</sup>[70, p. 62], for instance.

### 3.4.1 Problems in Platform Games

To get detailed information about which problems may occur in an evaluated game, it is useful to know more about problems of games of the specific game genre. In this section, some problems of the platform game genre are listed. Some of them could also be detected using the heuristic sets which were previously mentioned.

Jonkers [94] claimed that it may be better to use several moving platforms instead of just a single one to bridge a wide gap, to avoid that the players have to wait for a long period of time which could bore them. Furthermore, pointless dead ends [92] and leaps of faith should be avoided. Players should see under the platforms so that it is clear to them if they would survive when they fall down. When they are supposed to jump down somewhere in the level and they are not able to see where they may land, there should be at least an indicator which tells them that they can safely jump down [94][70, p. 106].

In platform games it can be very frustrating when an enemy who is still off screen attacks or when there are very difficult passages very late in a level. Moreover, it can be annoying when a long period of time was spent to collect certain game elements and then the player dies and all items are lost [94].

---

<sup>1</sup>Unique Selling Point

### 3.5 Conclusion

The problems listed in this section were just a few examples to form an idea of what may cause problems. More tips about creating platform games and levels can be found in [28, 92–94, 70].

In any case, it is necessary to consider that not every difficulty in a digital game is a problem per se. It is necessary to distinguish between intended challenges which are very important in games, as they are essential to avoid boring the players and difficulties which decrease the game's fun factor by making it unpleasant and unnecessary hard to play. It is the job of the game designer and the evaluators to decide for each detected issue individually if changing or removing it may enhance the game's quality.

In the following chapter, it will be described how such problems can be found using different scientific evaluation methods.

## Chapter 4

# Game Evaluation Methods

This chapter presents different game evaluation methods and how they can be used for the evaluation of computer games.

### 4.1 Evaluation of Traditional Products

As empirical research has typically focused on the evaluation of traditional products, such as functional software or websites [33], several methods for the evaluation of such applications already exist [27]. But as there are differences between games and traditional software, these methods may not always fit for the evaluation of games, but may have to be at least adapted [55]. The following section aims to address some main differences which are necessary to consider in order to understand the requirements of evaluation methods for games.

#### 4.1.1 Differences between Traditional Products and Games

To determine to what extent an evaluation method may fit or how it has to be adapted, it is necessary to know the differences and commonalities between traditional products and games [27, 66]. Issues concerning the functionality, such as the menu, the controls and the user interface, are equally important for both areas. Therefore, these can be evaluated in a similar way. But in games, the user experience is much more important. When working with a functional application, such as an email program, the user can achieve the goal (for instance sending a mail) without an enjoyable user experience. But when a game is played, an enjoyable user experience is essential, because it *is* the goal per se [88, 27].

A game focuses on recreational and not on functional interaction and is designed for creating pleasurable experiences, in contrast to productivity software, which operates solely according to general usability principles, such as task efficiency [55, 57]. Games are created to entertain while a player

solves tasks that are not created to be as easy to solve as possible, whereas productivity applications are created to help people accomplish tasks [80].

Game design is not about usability per se, but usability can establish a good foundation for an enjoyable user experience. Productivity software is outcome-oriented, whereas games are process-oriented. Therefore, the user experience is most important, as games are usually designed to generate positive emotions or enjoyment [55, 57]. This results in the necessity of developing concepts particularly for the evaluation of games.

#### 4.1.2 Classic Evaluation Methods

Classic evaluation methods have been extensively tested and offer a number of ways to examine the quality of an application. They have already been used in a broad variety of research fields such as, for example, in anthropology [4] or for analyzing research and innovations. In the latter, they have already been applied since the 1960s [51, p. 7].

Some of these methods can be used by indie game developers very easily, but others require some expert knowledge or expensive equipment. In this section, some of them are described briefly and their advantages and disadvantages for the use of evaluating computer games are highlighted.

##### Direct Observation

A direct observation includes one or more users and a supervisor whose task it is to *merely* supervise the participants without interacting or talking to them. In this way, it should be avoided that the users are disturbed to achieve that they interact in a natural way with the tested subject [69]. Therefore, direct observation has a high degree of ‘ecological validity’, which means that the observed behavior is very similar to the user’s natural behavior which would appear if there would not be any laboratory test situation [69][74].

If, for example, a game would be evaluated, this would allow the player or players to interact with it in a non-intrusive manner. Whereas it should be noted that the so-called *Hawthorne Effect* could occur. This describes the phenomenon that participants most likely achieve better results when they are monitored [69].

To get qualitative results from a direct observation, it is necessary to have a highly qualified supervisor. This person has to observe the participant, for example in matters of facial expressions, interaction with the product, conversations and reactions, which requires very good skills [27]. In games, in which the interactions between the player and the computer can be very complex and quick, the challenges for the supervisor may even further increase.

### Free Interaction

The classic evaluation method called free interaction separates the testing into two parts which are taken in turns. In the first part, the participant executes a (given) task and then the supervisor asks him or her content-related questions. This method poses a significant complication when used for the evaluation of games, as it can destroy the gameplay experience and therefore lead to biased results, which are distorted by the testing itself. The participant's capability to sink into the game, to experience immersion and flow, is restricted because of the communication or even just the presence of the supervisor [27]. Therefore, evaluations in which the test persons play on their own without anyone disturbing them may lead to less distorted output.

### Thinking-Aloud

For the classic thinking-aloud method, the player verbally communicates thoughts, ideas and decisions while playing. This can be used for the evaluation of functionality, to find out if the players understand what they should do and how they should do it. It helps to reveal *why* users do something. [27, 31]. One problem with using this method for evaluating games is that the player and gameplay experience may be altered. The participant has to concentrate on telling every single thought while playing, which influences the individual perception of the game. The use of this technique is highly dependent on the game concept [27, 35]. For a very quick game, for which the player has to be highly concentrated all the time, it may be better to choose a different testing method, as the influence of articulating may completely destroy the original game experience.

### Retrospective Thinking-Aloud

Retrospective thinking-aloud is similar to the thinking-aloud technique, but the commenting is only done at the end of the test. First, the player can play the game unhindered and without being distracted. Afterwards the participant and the supervisor analyze the video which was captured during the test session. Fierley and Engl discovered that the execution of this technique can be difficult for some people [27], especially because the method highly relies on the player's memory. Moreover, retrospective thinking-aloud is even more time consuming than the original thinking-aloud technique [31].

### Constructive Interaction

Another variant of thinking-aloud is constructive interaction. It involves two test persons who use the system together, which makes the test situation more natural. They talk to each other while solving a problem, therefore

they may make more comments as their talking is not purely for the supervisor [31].

### Conclusion

There are already a couple of classic evaluation methods which are generally useful for capturing player feedback and subjective user experiences. But all of them have specific strengths and weaknesses. Most of them are very time consuming and have drawbacks when it comes to the evaluation of games, as they can influence the gameplay experience [18, 20, 27].

All of the presented play-testing techniques require user participation. This has the advantage that direct information about how people play the game is provided [31]. But this necessitates the organization of a test session, which may be time consuming and may require some resources, such as the test environment (the room) per se or the game setup (for instance a computer or game console). Moreover, people have to be invited to the test and a supervisor has to guide them through the test session. Therefore, these techniques may not be ideal for indie game developers, who usually have a rather limited budget [44].

## 4.2 Questionnaires

Questionnaires represent an indirect method of evaluation, as not the system itself, but the information and opinions of users using the system, are collected. A problem thereby is that user statements cannot always be taken on trust. Therefore, data about player behavior should be given a higher priority than user claims. An advantage of questionnaires is that they offer the possibility to create statistics and to acquire subjective user input and the opinions of real end users. But, the fact that this information is subjective results in the requirement of numerous participants (in [31] a minimum of 30 people is suggested) to get significant results. The biggest difficulty in preparing a survey is to decide the wording of the asked questions. They have to be understandable, but what seems to be straightforward to one person may seem very difficult to another person [2].

This may also be caused by the differences between the participants such as gender, age, language, level of education or income. Moreover, it is also necessary to consider that the question length and the order of the questions may also influence the evaluation. Questions for which the participants are supposed to select one out of a couple of options require a particularly well-considered design, because there are several things which have to be taken into account. Questionnaire designers should consider, for example, if there is a *Don't know* option, the number of response options, if there is an even or odd number of options or the order and direction of the options [47]. Therefore, questionnaires require some experience to design [31]. But these

difficulties can be avoided by using an already existing questionnaire such as the *Game Experience Questionnaire (GEQ)* [34, 36]. This questionnaire consists of three modules: the core questionnaire, the social presence module and the post-game module. All three modules have to be evaluated directly after the gameplay session. The core questionnaire analyzes the game experience regarding seven components: immersion, flow, competence, positive and negative affect, tension and challenge. The social presence module measures psychological and behavioral involvement and should only be used when co-playing is involved in the game. Co-playing in this context can mean that virtual (in-game) characters, online players and/or real people are involved in the gameplay. In contrast to the first and second module, which investigate the feelings of the participants during playing, the post-game module assesses how the players felt after they had stopped playing. For the *GEQ* also an in-game version, which has an identical component structure, was developed. This should assess the game experience at multiple intervals during the game session [34].

For evaluations with children, the *Kids Game Experience Questionnaire (KidsGEQ)*, which is based on the *GEQ*, can be used. The *KidsGEQ* is adapted to a child-friendly format and wording. As a self-report instrument it measures in-game experiences of children at the age of 8–12 years [68].

The *Social Presence in Gaming Questionnaire (SPGQ)* is another questionnaire specifically created for games. It was developed based on focus group interviews of gamers. It studies their awareness of and involvement with their co-players as gaming is often not only about interaction with the game content, but also about social interactions [43].

Moreover, there is also the *Game Engagement Questionnaire*, which was developed to measure the engagement of a player in playing video games [9] and the *Networked Minds Measure of Social Presence* [7].

### 4.3 Heuristics

Nielsen defines the heuristic evaluation as a usability engineering method for finding usability problems in the user interface design [63, 99] in which a set of evaluators examine the interface and judge its compliance with recognized usability principles, called heuristics.

But heuristics cannot only be used for the evaluation of an interface. In [18] Desurvire, Caplan and Toth define heuristics as “[...] design guidelines which serve as a useful evaluation tool for both product designers and usability professionals.” They propose four categories for game heuristics: *game play*, for problems and challenges the user has to confirm to win the game; *game story* including the plot(s) and the character development, *game mechanics* consisting of the programming that enables the interaction between units and the environment and *game usability* involving the interface and



the input and output elements, which are used to interact with the game, such as a mouse, a keyboard or a head-up display.

Usually three to five expert evaluators are used to perform a heuristic evaluation. The original approach is for each one to inspect the system alone. After everybody has finished, they are allowed to communicate their findings. This is important to ensure that each evaluation is done uninfluenced by one another. During the test session, the experts analyze the system several times and compare it to the carefully selected list of heuristics [31].

As heuristic evaluations do not require user participation, they can also be used to evaluate early mockups. They are inexpensive and can be performed in a short amount of time, but, on the other hand, they require skilled evaluators [59, 66].

There are multiple heuristic sets available for the evaluation of video games and especially their playability [18, 26, 42, 66]. The research of playability heuristics that should help evaluate games, has been active in recent years, but it is still unknown how helpful they are to identify playability problems in games [41].

## 4.4 Gameplay Metrics

Game analytics have become increasingly important in the gaming industry in the last few years, as digital games are becoming larger and more complex [20, 88, 22, 33, 80]. This user-oriented approach is originally based on instrumentation methods in Human Computer Interaction (HCI) [20].

Gameplay metrics are numerical instrumentation data created by a monitoring software during the interaction of a player with the game [20, 59, 80, 81]. They offer quantitative, objective, time-stamped and highly detailed information about user behavior and user-game interaction for the entire test session [20, 59, 65, 80].

For this evaluation method no supervisor is needed. The data is automatically tracked by the system while the user is playing the game. In comparison of doing all the work by hand, as it is often done for classic evaluation methods [20] (see section 4.1.2), such a system can be deployed more timesavingly.

This makes it possible to collect numerous information from a large number of users [59, 65, 80]. Furthermore, the results may also be more accurate, errors can be reduced [37] and it allows the user to play the game in an undisturbed manner, which is a significant advantage compared to the methods presented in section 4.1.2. The game experience is not biased and the player can fully immerse themselves into the game. Game metrics data can provide information about potentially any action the player takes while playing. They can be recorded in temporal and spatial resolution and be mapped to a specific point in the game [80], which makes it possible to create a very

detailed image of the complex interactions in the game [59].

To track gameplay metrics, an infrastructure, meaning a metrics tracking system and a place to store the data (usually a database), is required. But when this equipment is available, data can be collected during testing, production as well as during the live phases of the game [20]. This makes it possible to receive data from clients who have already bought the game, as this was done in the case study of the game *Shadowrun* which is described in section 5.1.3.

Tracking data is a very complicated process [37]. In addition, their number, as well as their complexity can lead to challenges concerning their interpretation. Furthermore, the data does not include any information about the reasons why the players behaved the way they did, the emotional effects generated by playing the game or the quality of the user experience. Gameplay metrics analysis can inform what players are doing, but not why. Therefore, it is recommended to use additional user-oriented game testing methods [20, 65].

But not only gameplay metrics but also other quantitative instrumentation data, such as engine performance, project progress, sales across different countries or user interaction with the game, can be tracked as well [20, 80]. In the following section, different sources for analysis data are described.

#### 4.4.1 Data for Analytics

There are various sources of game-related information which can be tracked. [88] classifies them as follows:

##### **Performance Data**

This data is used to measure the performance of the software-based infrastructure behind a game. This is especially important for online games. Related metrics are, for example, the frame rate (which is usually measured in frames per second) at which a game runs on a specific hardware platform.

##### **Process Data**

Process data measures information about the developing process of the game. An example for process data is the task-size estimation or the average turnaround time of new content being delivered.

##### **User Data**

This data is related to the people who play the game. Depending on the perspective, they are either *customers* or *players* [88, 24].

**Customer Data:** These are for tracking metrics which are related to the revenue of the game. These metrics do not depend on the game genre and can be easily compared to other games. There are several very common metrics such as the average revenue per user (ARPU) or the daily active users (DAU) [88, 24]. There are a variety of tools which help track these metrics, for example the online tool *Game Analytics* [90].

**Player Data:** Player data focuses on in-game behavior. Information about how people interact with the game, its components and other players are tracked [88]. This can include the average of how long it takes a player to finish a level, how often a player reloads the gun during a fight or how many points are scored.

This thesis will focus on user and especially on player data and how they can be analyzed and processed to find problems in the level design.

#### 4.4.2 From Data to Metrics

To understand the differences between data and metrics it is necessary to define some naming conventions. Unfortunately, there is no standard terminology for game analytics and in literature different terminology can be found [11, 24]. In [24], Drachen, Canossa and El-Nasr categorized the collected data depending on its progress in processing as follows.

The raw data that is extracted from the game and that can be stored in the database is called *telemetry*. Telemetry is data obtained over a distance. It is measures of the attributes of objects whereas objects can be, for example game objects or players [87, 24]. An example for telemetry is the position of a character or the length of a call to the customer service. To work with the data it has to be *operationalized*, which means that the units (such as meters or milliseconds) in which the data is stored have to be defined. After this *operationalization* the data is not called *telemetry* anymore but, depending on the scientific field, most often either *variable* or *feature* [24]. This thesis uses the term *feature* for further explanations.

The data can then be transformed to interpretable measures, called game metrics. Examples for game metrics are the average completion time of a certain level, the number of tries until a user finishes a level successfully or the revenue per week. Usually metrics are calculated as a function of something [87, 24]. The typical unit is time, but can also be something else such as the build version of the game program.

It is not possible to clearly define a border between metrics and features as metrics can also be features and vice versa. But a game metric can be defined as “a quantitative measure of one or more attributes of one or more objects that operate in the context of games” [24].

### 4.4.3 Classification of Metrics

Several different classifications of metrics have been proposed. The following definitions refer solely to user data as described in 4.4.1.

One classification useful for research purposes is to separate user metrics by their applicability into *generic metrics*, which can be used for every digital game, *genre specific metrics*, which are dependent on the game genre and *game specific metrics*, which apply only for one individual game [24].

A more development-oriented approach was suggested in [88, 24]. This divides user metrics into the subgroups *customer metrics*, *community metrics* and *gameplay metrics*.

#### Customer Metrics

Customer metrics are related to the user as a customer. They measure, for example, the cost of customer acquisition or retention and are important for marketing and management. They can be used to provide insights into the download or installation rate, the in-game or out-of-game purchases or the countries in which the game was downloaded. But also information about bug reports, complaints, the interaction with the customer service or other social-interaction platforms are assigned to this category [88].

#### Community Metrics

Community metrics measure all information of the game community such as the growth of the community or any forum activities. They track the interactions between players, which can take place using different functionalities and applications. Interactions can be either in-game, using for example chat functions or post-to-*Facebook* buttons, or out-of-game, using forums or conversation applications such as *Skype* or *TeamSpeak*. Mining chat logs and forum posts can be useful to find problems or bugs in the game and is especially important in multiplayer games [88].

#### Gameplay Metrics

As already defined, gameplay metrics are data about user behavior and user-game interaction [20, 88].

They are used for the evaluation of the game design and the user experience and are therefore the most important metrics when it comes to user research, quality assurance or design questions [88].

Everything which is done by the player in the game, such as running, jumping, trading, collecting or navigating, can be tracked for evaluation. Therefore, thousands of gameplay metrics can be tracked during a single game session [88, 24].

In [88] four types of information that can be logged for each happening in a game are introduced:

- *What* is happening?
- *Where* is it happening?
- At what *time* is it happening?
- *Who* is involved?

Gameplay metrics are especially useful for informing game design. They provide the opportunity to discover if a specific game area is over- or under-used, if game features are used as intended by the game designers or if certain challenges hinder the player's progress within the entire game. Depending on the initiator and the action, gameplay metrics can be split into three categories [20, 88, 24]:

- In-game metrics consist of all information about in-game actions and player behaviors as well as interaction with game assets.
- Interface metrics include all information about the player interacting with the game interfaces and menus.
- System metrics cover the actions of the game engines and their subsystems such as artificial intelligence (AI). When an AI attacks the player or the player ascends to the next level, these events are classified as system metrics.

#### 4.4.4 Gameplay Metrics for Platform Games

It is highly dependent on the game which gameplay metrics create meaningful results and enable good insights into what is happening during the gameplay. But as games of a specific genre have similar game mechanics, it is at least possible to propose some examples of useful metrics for a game genre. As this thesis only refers to platform games, no metrics for other genres are provided, but some can be found in [24].

In platformers, it can be useful to track jumping, progression speed, items collected, damage taken and the source of damage, AI-enemy performance or power-ups/abilities used [24]. Details, such as which specific variables to track, which units to use or the frequencies of the measured metrics, have to be decided by the game developers depending on the game concept. These decisions are especially important, as described in the following section.

#### 4.4.5 Feature Selection

Feature selection is the process of choosing which data to collect. This is not an easy process and may be different for every game. Which data to track depends on the goal of the game analysis as well as on the game. A goal could be, for example, increasing the monetization or the user experience of a game [88, 80].

In the gaming industry it is most often not possible to track everything because lots of resources for tracking, storing, analyzing and possibly also bandwidth for the transformation of the data would be required [88, 24]. In games, there are large numbers of events that can be tracked. This may create enormous amounts of data which have to be analyzed and interpreted efficiently later on [37]. For their study in *Tomb Raider: Underworld*, Eidos Interactive tracked 12 features of about two million players over a time period of three months, which led to over four terabytes of data. For datasets such as this, already a simple database SELECT query can lead to several minutes of execution time [11].

It should also be noticed that tracking more data does not always mean better insights into the game, but can also add noise and lead to confusion. Different production stages also require different data. As an example, it may not be relevant and also not possible to track in-game purchases in an early developing stage. Moreover, every tracked metric has to be programmed and a person for the analysis of the data is required, as the largest dataset is meaningless if there is no one to analyze it [11, 88].

Therefore, the complexity has to be reduced and the tracked features have to be selected. This has to be done very carefully as it includes the risk of missing important information and adding bias to the dataset. To minimize this bias, the selection should be as well informed as possible and in communication with all relevant development teams including stakeholders, developers and designers. Especially the expert knowledge of the designers, who designed the game may be helpful, as they know exactly what can happen in the game and what the most relevant data to be recorded is [88, 24].

Peireira and Roque propose the use of a *Goal-Question-Metrics (GQM)* approach to support the definition of gameplay metrics for analysis of player participation. Their methodology is intended to support game designers in considering how the players take part in the game. They specified six perspectives of participation consisting of *playfulness*, *challenge*, *embodiment*, *sensemaking*, *sensoriality* and *sociability*. Based on these, they defined goals, questions and metrics for the evaluation of games [65].

Another possibility to reduce the resource requirements is sampling the data, which can of course also add bias [88, 24, 72]. There are several possibilities for sampling data, which are described in detail in [23].

#### 4.4.6 Tracking Strategies

When it is decided what to track, it is also important to define which tracking strategy to use for each part of information, in order to get the most valuable output of the evaluation. In [24] three different strategies are mentioned:

- A predefined event occurs such as the player starting the game, dying or using a weapon [24, 80].

- The information is recorded on a continual basis following a specific frequency. The position of the character can be, for example, tracked every second [24, 80].
- The tracking is started or ended by the initiative of the designer, for instance when a new patch or update is provided [24].

It is also useful to track additional data, such as the timestamp, when the data was recorded or the coordinates of the player at that particular moment. Moreover, also the originator of the metric, meaning the character which was responsible for the appearance of the event, the camera angle, the character's movement or other related information can be used to enable deeper insights into the gameplay [80].

#### 4.4.7 Analysis

After collecting all the data, the challenge is to make sense of these data by interpreting them. This makes it possible to use them for further decisions. Even for developers who know the game very well, it may be difficult to interpret the metrics they collect. Moreover, it is important to mention that a simple metric does not say anything without the context, such as for instance the previously recorded metrics [33].

### 4.5 Combining Qualitative and Quantitative

Traditional testing methods come with a large set of limitations [55]. They can be very time consuming and their output is very subjective (see section 4.1.2). Post-game interviews or surveys bear the problem that they are difficult to relate to a specific design feature. Moreover, when players are interviewed at the end of a test session, their memories are already biased and to a certain degree imprecise. Using smaller in-game surveys, which are asked in intervals, may eliminate such problems, but currently it is not sufficiently investigated to what time intervals they should be used [55].

The limit of game metrics is that they cannot provide any contextual information and they cannot measure social factors or the game experience [80]. They cannot offer insights into the players emotions, feelings and thoughts, such as if they find the game unfair, if they understand what to do, if they like the challenge, the story line or the gameplay [30, 72]. Gameplay metrics analysis cannot inform about why players behave the way they do. Moreover, metrics also miss demographic information about the users [20].

But gameplay metrics can tell, for example, that players die in one area many more times than in others [30]. They offer high-resolution, objective and quantitative data about player-game interactions [80]. Whereas traditional methods, such as usability testing or playability testing, can be used to discover, for instance, emotional feedback, if players can interact with a

game effectively and if they experience fun [20, 80].

By combining qualitative (such as surveys or questionnaires) and quantitative methods (for example metrics), a better understanding of the relationships and interactions between the player(s) and the game can be achieved. This enables detailed insights into the complex interactions driving gameplay and player experiences [55, 59] and the possibility to directly link game experience with design elements [59, 80]. To acquire information about the motivations and the reasons *why* players do what they do often a method-mix may be necessary [20, 80].

Therefore, it is very valuable to use both, collected game metrics data as well as survey data [72].



## Chapter 5

# State of the Art

In this chapter, three existing evaluation tools for video games, the possibilities they offer and some case studies in which they were already used will be presented.

### 5.1 TRUE

Over a period of several years, *Microsoft* developed a tool called *Tracking Real-Time User Experience (TRUE)* for testing video games [37, 76]. This tool combines the collection and analysis of game metrics with other human computer interaction (HCI) methods. The idea behind it was that combining existing evaluation methods, such as logging and surveys, with metrics data and rich contextual data, such as video recordings, would make the evaluation very flexible and powerful. But they created the tool not only to collect information, but also to visualize the data. This analysis tool was intended to reduce the required analysis time to enable rapid, iterative changes in the game. Figure 5.1 depicts the architecture of *TRUE*.

*TRUE* tracks user initiated events (UIEs), which are created when the user interacts with the system. For each event, the timestamp and other contextual data that belong to the UIE are also recorded. This helps to identify the cause *why* behind the *what*. When there is, for example, a car crash in a racing game, not only the crash itself but also the car which was used, the difficulty setting or the position on the route can be tracked. This helps to understand why the crash happened and therefore can help to minimize misinterpretations of the metrics data. Moreover, so-called *attitudinal data* is also collected. For example, after each race a short survey can be asked to discover if the player found the race fun or too difficult.

*TRUE* also captures a digital video of the player during the testing session. This video is automatically indexed with the tracked events which makes it possible to quickly find the video sequence correlating to an event. This functionality is useful to find out if there are any problems in the game

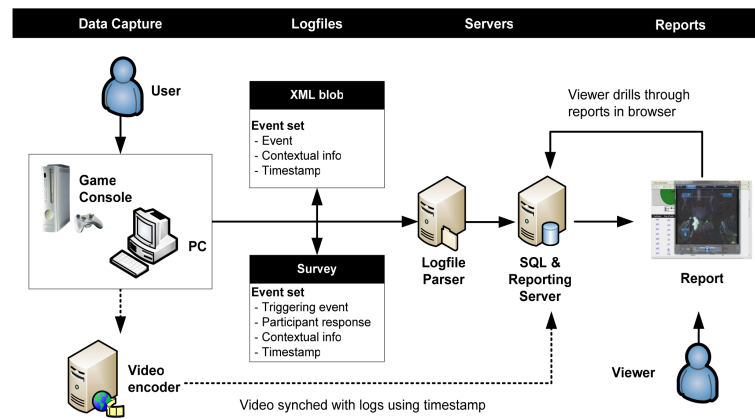


Figure 5.1: This image visualizes the architecture of *TRUE* [37].

and to verify if these particular events are actually seen as problems by the users.

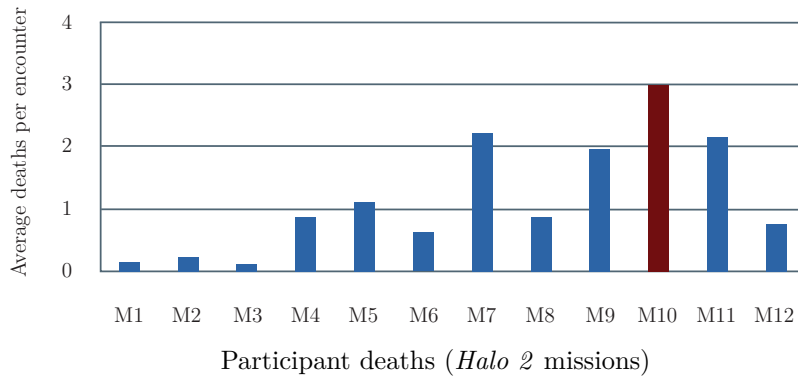
The recording of the data is just one part of the working process. The visualization of the information is equally important. To facilitate data analysis, *TRUE* can create a series of graphs and tables. It is possible to see an overview of the data and then navigate down to detailed views. Cross-links enable it to find the desired information as quickly as possible [37]. This feature was utilized heavily during the evaluation of the game *Halo 2*, which was one of the first games tested using the *TRUE* tool. In the following section, more information about how they evaluated the game and what they discovered will be given.

### 5.1.1 Case Study: Halo 2

*Halo 2* is the second game of the Halo game trilogy and a very successful<sup>1</sup> first-person shooter game for Microsoft's Xbox 360 created by Bungie Studios. A large-scale user study was completed in order to identify unintended increases in difficulty in the single player component of the game *Halo 2* [37]. The tests were done at a usability lab using 51 stations with each having a monitor, headphones, an Xbox development kit and a game controller. The 44 participants all had prior experience with other first-person shooter games. Most of them were able to finish the game within the test session.

During the gameplay, they tracked information about player deaths, such as the time when the event happened, how they were killed and by whom. They also collected attitudinal data by pausing the game every three minutes to ask questions to rate the difficulty of the game. It should be noted that

<sup>1</sup>Microsoft sold about 8,000,000 copies of *Halo 2* which means that about 33% of the userbase bought this game [85, 100].



**Figure 5.2:** This diagram illustrates the death events divided by mission in *Halo 2*. In mission M10, clearly more deaths happened than in all the other missions. The original diagram can be found in [37].

this could have affected the flow and the player experience of the game as the player’s game flow was interrupted by the questions.

Using the data, they discovered that in a particular mission, players died more often than in all the other missions of the game (see figure 5.2). With *TRUE* and its visualization tool, they were able to “drill down” into the specific UIEs of this mission and locate the area which caused the high death rate. By focusing even further, they separated the deaths by causes and used the possibility of viewing the corresponding video sequences of the deaths. In the end, the designers were able to understand how to fix the problem by making some changes concerning the spawning and behavior of the enemies. This indicates that having a good and flexible visualization tool for the tracked data can help find problems and their causes and may reduce evaluation time.

In the second test session, they tested if the changes in the game were successfully done. The death rate at the problem points had dropped immensely but they were concerned whether the game was too easy at the specific area in the game now. Using the attitudinal data, they found out that only 4% of the players reported that the game was too easy whereas 74% answered that the difficulty was “[a]bout right, I’m making good progress”.

### 5.1.2 Case Study: Halo 3

The next game in the *Halo* series was again extensively tested [21, pp. 487–488]. During this study more than 3,000 hours of gameplay of about 600 everyday gamers were analyzed [103]. In a dedicated usability lab for stress-testing of games people were monitored during the entire play-testing session using rotatable video cameras. The supervisor sat behind a one-way mirror

observing every expression and behavior of the player.

Everything which was happening in the game, such as the player's location, when and where a weapon was fired, the type of the used weapon, the use of vehicles and data about player deaths, was tracked. Every fixed timestamp screenshots were made to find out how quickly the participants got ahead in the game. For the analysis of player deaths heatmaps were created.

Additionally, video recordings and the thinking-aloud technique (see section 4.1.2) were used, as well as in-game questions which interrupted gameplay every few minutes to find out how engaged, interested or frustrated the participants were [103]. The gathered information was visualized on a map as dots in different colors at the positions where the players had answered the question [21, pp. 487–488].

This method-mix enabled them to find differences between what the game designers intended the players to do in the game and what they actually did [103]. It made it possible to detect difficult passages and provide detailed suggestions to the development team to enhance the game [21, pp. 487–488]. As an example, they discovered that in a particular level, players often run out of ammunition although a lot of it was placed in the level. They had a closer look at the problem and figured out that a significant number of people were firing at enemies when they were still too far away. Knowing this, they were able to fix the problem by coloring the reticule<sup>2</sup> when the target was in range.

### 5.1.3 Case Study: Shadowrun

*TRUE* was also used for the evaluation of the multiplayer, first-person shooter game *Shadowrun* [37]. This game allows the users to customize their characters. They can buy different weapons and choose between four different main characters. In the study longer-term trends and patterns in player behavior were analyzed. Data from ten thousand players who played the game on their personal *Xbox 360* were tracked using the *Xbox Live!* online web service to stream the live data. The entire study lasted about four months and enabled the designers to tweak game parameters and provide an updated version as a download.

One of the observed player behaviors involved the popularity of the four different character classes. It was intended by the designers that each character has different advantages and disadvantages and appeals to a specific playing style. The analysis of the metrics data tracked by *TRUE* revealed that after some time, the *Elf* character was clearly preferred by the users and that the gamers who chose this character were also more successful in the game. Having a look at the metrics data, the game designers were able

---

<sup>2</sup>Crosshairs

to tweak the attributes to accomplish that no class was clearly dominating the others.

Furthermore, the various weapons were analyzed. They were created by the designers to fulfill different purposes. To receive information about this topic, the position of the killer and the victim as well as the used weapons were tracked every time a player killed another one. This made it possible to define the relative effectiveness of the weapons depending on the distance to the murdered player. The data showed that some weapons were ineffective at nearly all distances, which was of course not intended. Using this information made it possible to tweak the parameters of the weapons to match each unique purpose.

Having a game with lots of different parameters often makes it difficult to keep track of every single variable. As seen in this study, current video games are often very complex, which can make it impossible to tweak every single value perfectly during development time. Some (player) behaviors may also just not be foreseen by the designers. Having a tool which makes it possible to supervise the game when it has already been sold and played by the consumers is therefore a great possibility to enhance the game even after launch and learn more for further projects.

## 5.2 EIDOS Metrics Suite

The company *EIDOS* created a software called *EIDOS Metrics Suite*<sup>3</sup> for tracking game metrics data from their own games. The metrics are logged as sequences of events and stored on an SQL server. For each event, multiple types of data are tracked, each with its own timestamp and contextual information [22].

The *EIDOS Metrics Suite* can be used in combination with the *Xbox Live!* web service. This tool can be used for tracking live data directly from the *Xbox* engines of the customers who have already bought the game. Therefore, the collected data is free from the bias which can be introduced when a game is tested in an external laboratory setup, as the players can play the game in their natural habitat. Consequently, large amounts of quantitative game metrics data can be saved. But, on the other hand, no qualitative information, such as data created using questionnaires, is collected as can be done with *TRUE* (see section 5.1). But the metrics suite does of course not rely solely on the *Xbox Live!* web service, but can also track data directly from the game engine, which is useful for internal testing during the production process [20, 22, 49].

---

<sup>3</sup>The *EIDOS Metrics Suite* was renamed into *Square Enix Europe Metrics Suite* after the company was bought by *Square Enix* in 2009 [97, 49].

### 5.2.1 Tomb Raider: Underworld: Clustering into Different Player Types

*Tomb Raider: Underworld* is, as all games of the Tomb Raider series, a combination of an adventure game and a 3D platformer [49]. They include lots of challenges such as solving puzzles and navigating through the world without falling down from too far above. It was also the first game which was tested using the *EIDOS Metrics Suite* [49]. For this case study, data from 1365 players who completed the game was used [22].

Six gameplay features were extracted from the logged metrics data. The first three features described the reasons of death as death provoked by an opponent (AI agent), death caused by the environment and death by falling. They also used the total number of deaths, the completion time for the whole game and the number of Help-on-Demand requests<sup>4</sup>.

To analyze this large-scale data collection, a data mining method called self-organizing maps was used. More detailed information about how this was achieved can be found in [22]. As a result, the chosen method grouped the data into four data clusters which represented player types. The first cluster corresponded to players that did not die very often and whose deaths were mainly caused by the environment. They completed the total game very quickly and did not use the Help-on-Demand (HOD) functionality very often. The authors called them *Veterans*. The second cluster represented players who died very often because of falling. It took them very long to complete the game. They did not really use the HOD feature for solving puzzles. Therefore, they were called *Solvers*. The third cluster included players who primarily died because of opponents and whose completion time was below average. The so-called *Partifists* formed the largest group. The players assigned to the last cluster, who were named *Runners*, completed the game very fast. They died very often and mainly because of the opponents and the environment. It should be noted that this study was created using *only* metrics data of players who had already finished the game. Therefore, there may be also other player types playing *Tomb Raider Underworld* which are not included as they may not finish the game (ever).

It would be interesting to ask the players about their player style and check if they would also count themselves to the cluster which was assigned to them by the algorithm. Unfortunately, this study does not evaluate the feelings the players had while playing the game.

Using such an automatic data mining approach as described in this study can help reduce the required amount of time for the analysis of tracked metrics data. In this case, the data was used for detecting different player styles but it would also be conceivable to use such methods to find, for example, problems within the level design or the whole game.

---

<sup>4</sup>The Help-on-Demand functionality in *Tomb Raider: Underworld* gives the player hints or the complete solution to a puzzle in the game.

### 5.2.2 Fragile Alliance

In the case study of the game *Fragile Alliance*, the designers at *IO Interactive* wanted to find out if the correct events take place at the right location. In particular they tracked the player deaths and analyzed their positions and frequencies [21, pp. 298–303].

*Fragile Alliance* is a team-based multiplayer game mode from the shooter game *Kane and Lynch 2*. In this game, the players have to execute a heist as mercenaries. The game is split into different rounds. At the end of a round the winner is whoever leaves with the most money. When mercenaries die they are respawned as police officers and work together with AI bots trying to kill the remaining mercenaries. Players can kill teammates and rob their money which turns them into traitors. Killing a traitor gives the murderer an instant reward; if a police officer kills the traitor he was murdered by, this reward is even bigger [20][21, pp. 298–303].

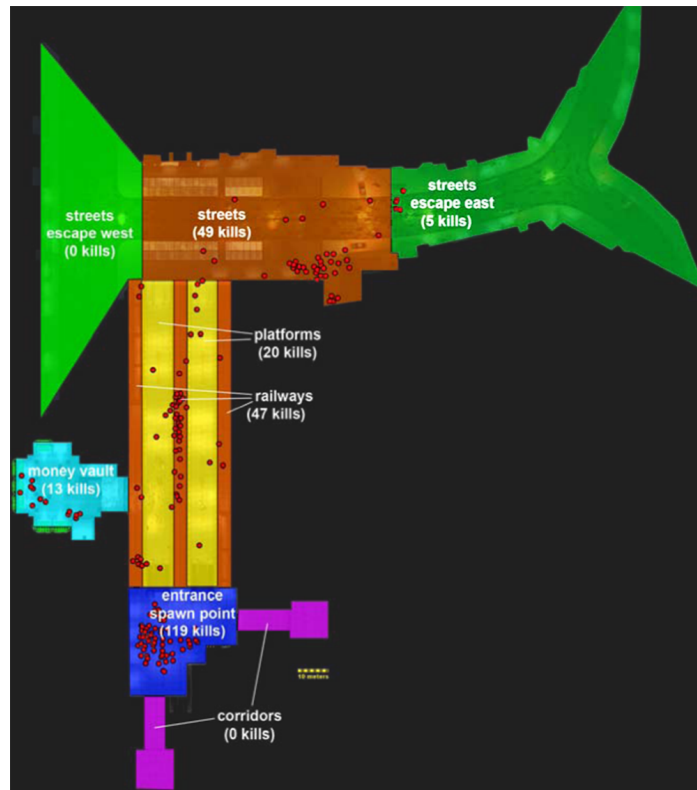
For the study, thousands of death events (roughly 38,000) were analyzed [21, pp. 298–303]. For the visualization, a Geographical Information System (GIS) called *ArcGIS* was used. There the four different zones were separated as different layers [21, pp. 298–303].

The user study was performed in order to find out if players performed the *right* actions at the *right* locations as intended by the game designers [21, p. 298]. Therefore, they analyzed the positions where players died. The level consisted of several locations (see figure 5.3): the area where the players are spawned at the beginning of the game; the vault where the money is stored; a subway station in the middle of the level; and the streets where the players exit the map. The separation into these areas allowed them a more detailed analysis. They combined spatial and temporal behavior and extracted some data which indicated that some characteristics in the game worked as expected: the AI police officer agents were, for example, spread over the entire map. Most of the ‘suicides’<sup>5</sup> occurred in the exit area where cars were driving on the street. They also found some patterns regarding which character role was killed when and where most often. Most of these results were also as expected, showing that the gameplay worked as intended [21].

The game was designed with the idea that there should be a shift in balance over time. At the beginning of the game, the mercenaries should be strong, but later on the strength of the police officers should increase. At the beginning, all players are mercenaries, but when they die they turn into officers and empower the role of the police. This was evaluated by comparing the roles of the killers during the first 45 seconds of play with those in the following 90 seconds of play (see figure 5.4). The first 53% of the kills were

---

<sup>5</sup>In this game a ‘suicide’ is considered as a death event which is not caused by any other AI character. As an example, vehicles which take too much damage can explode and kill a player.



**Figure 5.3:** The “subway” level map from *Fragile Alliance* showing the locations of player deaths in the different areas [20].

executed by mercenaries while only 38% of the murderers were either AI agents or police officers. Later, the amount of kills by mercenaries shrunk to 38% while police and AI bots killed in 59% of the cases [21, pp. 298–303].

This study demonstrates that even with tracking only one game metrics type<sup>6</sup> deep insight into how the game is played can be possible.

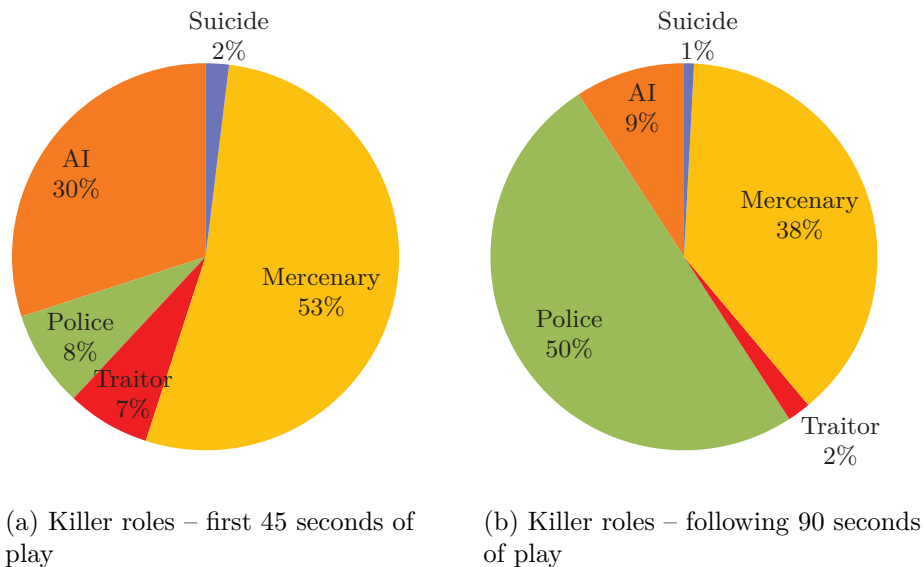
### 5.3 Volition’s Telemetry System

Lacking the resources *Microsoft* invested into their highly advanced software *TRUE* (see section 5.1), *Volition* created its own telemetry system with limited resources [48].

Tracking systems can create incredibly large amounts of data stored in a massive database. But it was necessary for them to create an efficient method for the data analysis in order to get reports quickly enough to avoid large delays in development. It was also important to create short and compact

<sup>6</sup>They only tracked information about player deaths.





**Figure 5.4:** The killer roles in *Fragile Alliance* on the “subway” level map in the (a) first 45 seconds and in the (b) following 90 seconds of play. The shift from the mercenaries towards the police is clearly visible. The original diagrams were published in [21].

reports as the designers might not have the time to read a nine-page report during development time. Hence they aimed to create detailed graphs, short video clips and short texts.

They included heatmaps in their tool as they are very useful and quick to create. Heatmaps visualizing player deaths enable insight into the difficulty of a specific area in a level and can be used to check if this is as intended by the designers. Other heatmaps they created in order to evaluate the shooter game *Red Faction Armageddon's* illustrated the progress of the players or the places where they run out of ammo. The tracking system also included the possibility to create simple tables and charts like bar charts which were used to show the usage of different kinds of weapons.

At *Volition*, they use the thinking-aloud method, interviews, surveys and observation-based tests beside their tracking suite. For the tests, they invite participants to play their game in the company. These are introduced to play as usual and answer the surveys which appear after they have completed certain milestones in the game. This may be an advantage in comparison to the questionnaires used in the evaluation of *Halo* (see sections 5.1.1 and 5.1.2) in which the users were interrupted every three minutes during the gameplay.

While the participants are playing, their in-game behavior is recorded and observed by developers and researchers using live-stream video data. This data is also stored for later evaluation as it allows researchers to see

the context in which specific problems occurred. When the observers detect a potential problem, the designers who are responsible for this part of the game can be notified immediately. By watching the video and having a look at the log data, they can get deeper insight into what is happening and what the cause for the problem may be, making it easier for them to find a solution.

This tracking tool includes roughly the same basic features as *TRUE*, but it does not offer the highly advanced possibilities, as for example automatically indexed video captures or interactive, hierarchical data visualizations including cross-link references *Microsoft* included in their tool. This shows that even with a smaller budget, it is possible to enhance common user testing methods with an automatic metrics tracking tool.

## 5.4 More Studies

The presented studies are just a few examples of game evaluations using game metrics data. There are several other documented studies about the evaluation of other games.

In [21] and [12], analyzes of the game *Kane & Lynch: Dog Days* concerning the weapon usage, the used paths and the player frustration are described. A number of different user studies are documented in [30] including tests about the player drop-off rate in the missions of *Splinter Cell*, the controlling of the camera on *Nintendo's Wii* in *Prince of Persia* and the climbing in *Assassin's Creed 2*. In [32] and [33] the game metrics of the racing game *Project Gotham Racing 4* were analyzed with the result that in four of the five evaluated features<sup>7</sup> 20% to over 70% of the available options were used in less than 1% of races [33]. Another rally video game case study is described by Guardini and Maninetti in great detail [29]. For *Tomb Raider: Underworld*, predictions when players would stop playing or in the case they would finish the game, how long this would take them, were created based on metrics data [49].

## 5.5 Conclusion

The studies show the power and flexibility of video game evaluation using game metrics either alone or in combination with other methods. Several different aspects of games can be analyzed as their playability, the usage of weapons, the used paths, the player frustration or player types to name just a few. For every part, other metrics might be useful; therefore it is important to know in advance what in particular should be tested to decide which methods to use and which data to track.

---

<sup>7</sup>In this study the game modes, the event types, the routes, the vehicles and the vehicle classes of the game were evaluated.

In some of the examples above, a combination of metrics tracking and other usability testing methods, such as surveys, were used. But none of them has tested the evaluation of the tracked data directly in the game to ask questions based on the collected data. This is the chosen approach for the evaluation tool presented in this thesis. More information about this will be given in the next chapter.

# Chapter 6

## Project

As part of the master's project, a metrics tracking system was developed and integrated into an already existing platform game. It consists of several components which make it possible to analyze the collected data immediately after it was tracked in the game and ask questions dependent on these metrics. It also supports questions which are not context-specific. Moreover, it includes basic functionality for the generation of statistics and data visualizations. This chapter describes the basic idea, the system and its components.

### 6.1 Concept

#### 6.1.1 Problem, Idea and Requirements

The original idea came up during the creation of the platformer *Elements*. The game was tested and questionnaires were used for the evaluation. This meant a lot of work and time for both the participants as well as the developers, who had to analyze the received surveys afterwards. The outcome of this study included very limited level-specific information such as positions which were perceived as being too difficult by the users or which were causing trouble in order to progress in the game. These could have helped to identify and fix potential problems within the level design.

This led to the idea of the game evaluation tool. The system and its utilization should be useful and affordable for indie game developers. The evaluation should not be very time consuming or expensive. It should require as few resources as possible but at the same time it should create valuable results which could help to alter the game and level design to improve the gameplay experience.

### 6.1.2 Solution

As already determined in section 4.1.2, traditional evaluation methods have several drawbacks which make them unsuitable for the previously described purposes.

The proposed solution was the creation of a tool which can easily be included in a game and which automatically tracks the metrics data produced by the players. This could enable detailed insight into what is happening in the game but could not define why the users react the way they do or how they feel while playing (see section 4.5). Therefore, it was decided to create a tool which combines metrics data and questions. Moreover, the system should also be able to ask questions based on the tracked data to get information corresponding to a particular situation or potential problem within the game.

To make it easy and inexpensive for game developers to track data of numerous test users, it was decided that the developed evaluation application should be usable by anyone, without the presence of a supervisor. This was achieved by creating a downloadable test tool which included all instructions necessary for the evaluation. This also had the advantage that users could test the game at home at any time and as long as they wanted and that they did not have to drive to any testing labour to do the evaluation.

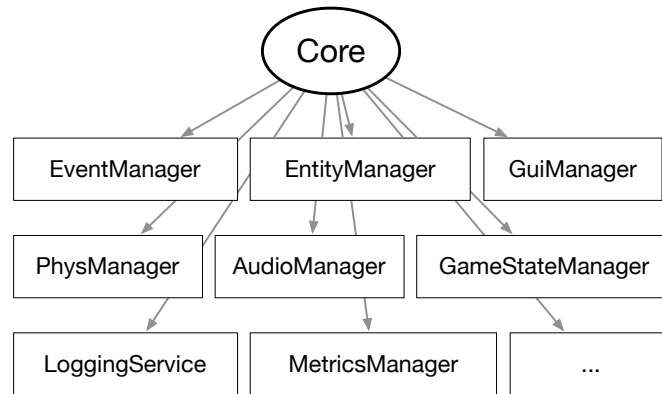
## 6.2 Architecture and Implementation

This section describes the main components of the tool and how they work together to enable a framework which fits the requirements described above.

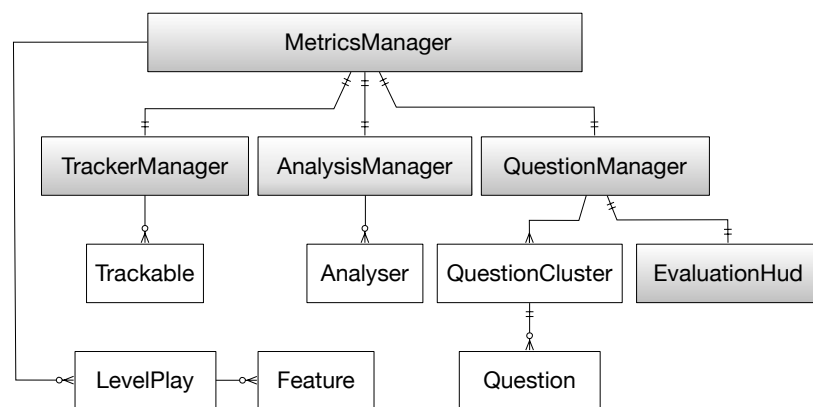
### 6.2.1 xis-engine

The developed system was implemented as a **service** for the *xis-engine* which is a self-made C++ cross-platform game engine. It was created by two students (Christoph Lipphart and the author) during their previous years of studying at the *University of Applied Sciences, Hagenberg*. The engine was mainly created for the development of 2D games and had already been used for several game projects including the platform game *Elements* (see section 6.3), which was used for testing the system.

The engine is service-based and was built to be easily extensible by creating additional, custom **services** (see figure 6.1). This proved to be very helpful as by implementing the metrics tracking system as a **service**, it was easy to add it to the core of the engine and to access it from everywhere in the game. Using a self-made game engine made it possible to easily access everything required for the metrics tracking system. In the case that something was not accessible per se, it was possible to develop the necessary interface therefore at any time during development.



**Figure 6.1:** The *xis-engine* consists of a singleton core which can have several different services such as the `MetricsManager` installed.

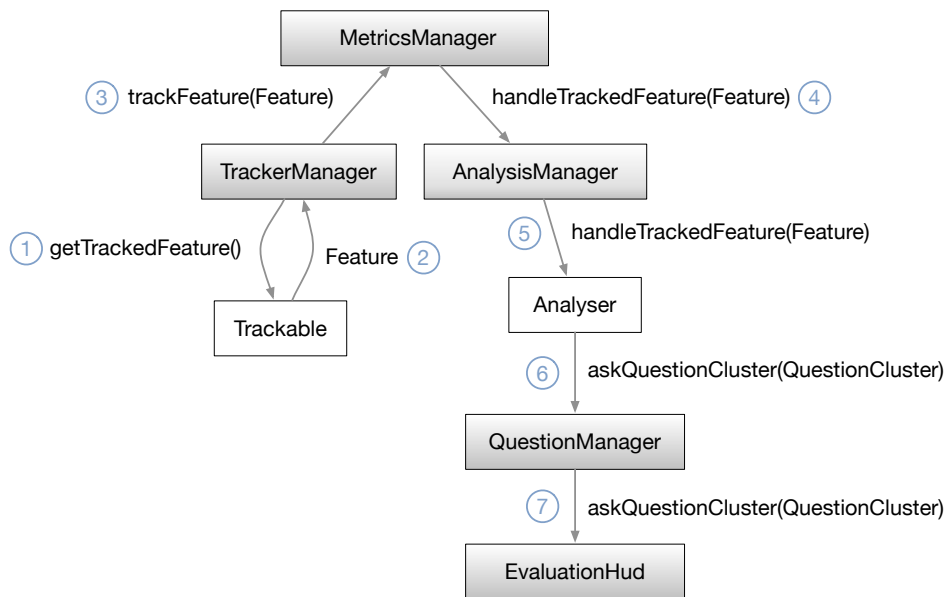


**Figure 6.2:** This diagram illustrates the entity relationships within the `MetricsManager` service. The white entities can appear more than once or never within the system while the grey components exist once at any time.

### 6.2.2 Components

The tool includes several components; each of them is responsible for a different aspect such as tracking, analysis or visualization. In this section, the most important parts will be described to give some insight into the architecture (see figure 6.2).

The basis for the tool is the functionality of metrics data tracking. This is done by the `MetricsManager` service, which is the interface for all data tracking functionalities. To understand how this process works it is necessary to understand its basic components. Figure 6.3 illustrates an example how the different parts can collaborate to ask a question.



**Figure 6.3:** This diagram illustrates a variant how a question could be asked: (1) Initially, the `TrackerManager` requests a (2) feature from one of its `trackables`. (3) This is handed over to the `MetricsManager` which stores it and (4) informs the `AnalysisManager` about the new input using the function `handleTrackedFeature`. (5) The `AnalysisManager` hands the feature over to its `analyser` (there could be also several `analysers`) which analyzes it and decides to ask questions which it packs into a `question cluster`. (6) It tells the `QuestionManager` to ask the cluster. (7) Finally, the `EvaluationHud` displays the questions on the screen and creates the necessary interface for answering them.

**Feature:** The feature is the smallest data unit which can be tracked. A feature has a type which is represented by a basic string and defines what information the feature stores. In *Elements*, there are, for example, features for position, game-over, level restarts or element changed events, but a feature can basically describe anything which can be tracked within a game. Features also store the player ID as a string value. This makes it possible to record several players in one single game at the same time and on the same device. Another basic piece of information which is saved in the feature is the timestamp when it was created. This makes it possible to list the collected information sequentially to reproduce the original happenings during the gameplay session.

It was very important that features could also hold any other information as for each event and for each game other values may be interesting to be tracked. Therefore, the feature was designed as an open data unit. This was reached by adding a variable of type `model` to it. This is a data structure

implemented in the *xis-engine* which is very similar to the *JavaScript Object Notation*<sup>1</sup>. It makes it possible to store values of different types such as numerical values, texts, vectors or even other `model` objects in it. Each of these values can be accessed using a unique string identifier, which can be chosen by the developer.

**Levelplay:** A levelplay includes a list of features. It stores the progress of one single level played by one user at one moment. A levelplay begins when the player starts the level and ends when it was finished independent of the reason *why* it was finished. In *Elements*, a level can be finished either by the player reaching the goal (in this case the level was finished successfully), a game-over event or when the user exits the level manually using the menu. Moreover, a levelplay stores the information at which timestamp it was started and the ID of the level which was played. It has also an ID which is calculated using its start time and the player ID. This makes it possible to uniquely identify each single levelplay.

**MetricsManager:** The `MetricsManager` was created as a `service` of the *xis-engine*. Therefore, it is globally available in the entire game using

```
XisEngine::GetInstalledService<MetricsManager>();
```

assuming that it was installed in the core in advance. It is the center point of the tool and offers access to other components of the system such as the `TrackerManager`, the `AnalysisManager` or the `QuestionManager` which will be described later in this chapter. It contains all levelplays and is responsible for managing, storing and reloading metrics data. This data is stored as text files in the `model` file format which was already defined above. Furthermore, it provides interfaces which make it possible to access the data. When a feature has to be tracked, this can be achieved by handing it over to the `MetricsManager` which then automatically adds it to the correct levelplay and keeps track of it.

**TrackerManager:** The `TrackerManager` makes it possible to track data at an individual time frequency. This can be helpful when, for example, the position of a character should be tracked every 300 milliseconds. Therefore, a so-called `trackable` has to be added to the `TrackerManager`. Subsequently, the manager will request a feature from the `trackable` every predefined amount of time and hand it over to the `MetricsManager`. As `trackable` is an abstract class, game developers can derive from it to make it perfectly fit their needs. If necessary, the `TrackerManager` can also handle several individual `trackable` objects at once.

---

<sup>1</sup>[www.json.org](http://www.json.org)



**AnalysisManager:** The `AnalysisManager` is dedicated to in-game analysis. It can manage several `analyser` objects. As an abstract class, `analyser` is informed every time a new feature is tracked or a new levelplay is started. This enables immediate analysis of the tracked data. The objective of this in-game analysis is that adequate questions can be asked at the exact moment when an `analyser` discovered *something interesting*<sup>2</sup>.

**QuestionManager:** The `QuestionManager` is responsible for managing and asking questions. Developers can decide for each question how to add it: either it is asked immediately, interrupting the gameplay, or it is asked later on. In the second case, the question is stored in the `QuestionManager` together with a string identifier. At the time when the question(s) should be asked, the program can tell the `QuestionManager` to start the survey which includes the questions stored with this ID.

**Question Types:** To make it possible to create very different types of questionnaires, several types of questions were implemented:

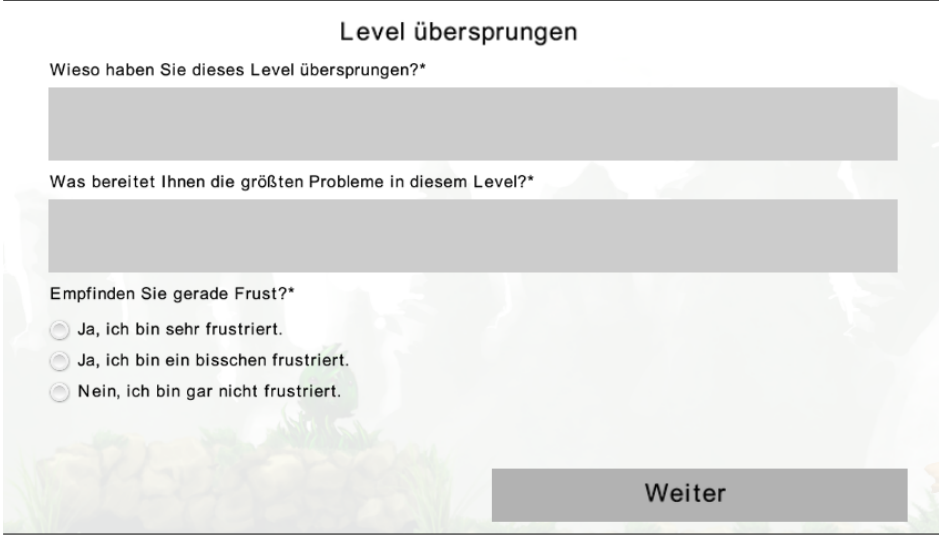
- Text questions provide a text field of variable size into which the users can write their answers (see figure 6.4). These open questions make it possible to receive qualitative feedback.
- Radio button questions (see figure 6.4) can have a various number of radio buttons. At one time only one option can be selected. When an option is already selected and the user clicks onto an other radio button, the previously checked option is automatically unchecked. Moreover, when a radio button question is optional a previously selected option can be unselected again<sup>3</sup>.
- Checkbox questions consist of a checkbox and have a name which is written right beside the checkbox. By using multiple checkbox questions at once, a multiple answer possibility can be formed (see figure 6.5).
- Info texts are another question type although they may not always represent a question. They can be used in the same context to insert information or super questions which are, for example, related to the following questions (see figure 6.5).

In section 7.1.5, the advantages of the various question types are listed and it is described how they can be used for collecting different kinds of information.

---

<sup>2</sup>The implementation of the `analyser` has to be done by the developers as only they know what they want to find out and what may produce meaningful results in their game. Some examples of what can be analyzed can be found in section 7.1.4.

<sup>3</sup>This was done to avoid getting wrong answers in the evaluation, for example when a user unintentionally ticks an option of an optional radio button question.



The screenshot shows a survey form with the following content:

- Title: **Level übersprungen**
- Question 1: **Wieso haben Sie dieses Level übersprungen?\*** (Text input field)
- Question 2: **Was bereitet Ihnen die größten Probleme in diesem Level?\*** (Text input field)
- Question 3: **Empfinden Sie gerade Frust?\*** (Radio button question)
  - Ja, ich bin sehr frustriert.
  - Ja, ich bin ein bisschen frustriert.
  - Nein, ich bin gar nicht frustriert.
- Button: **Weiter**

**Figure 6.4:** This screenshot shows a `question cluster` with two text questions and one radio button question. The system automatically added an `*` at the end of each question to indicate that they are required.

It was possible to set a question as required or optional. The system was able to emphasize required questions by adding an `*` at the end of it if this was wished by the developer (see figure 6.4).

**Question Cluster:** Questions are grouped into `question cluster` objects (see figure 6.4). One cluster can contain one or more questions. These are all shown at the same time; therefore a cluster can be compared to a page filled with questions in a survey. A cluster also has a title which can be used to quickly inform the participants about the main purpose of the questions on a page.

Clusters and the questions they contain are saved in a text file using the previously described `model` file format. Moreover, related information, such as the level ID, the levelplay ID, the player ID, a possible test session ID and the timestamp when the questions were submitted, are automatically saved.

Both, questions and clusters were created to be able to also store related information using the open data format `model`.

**EvaluationHud:** The `EvaluationHud` is responsible for the visualization of the questions. When some questions should be shown, it receives the `question cluster` to display from the `QuestionManager` and manages the

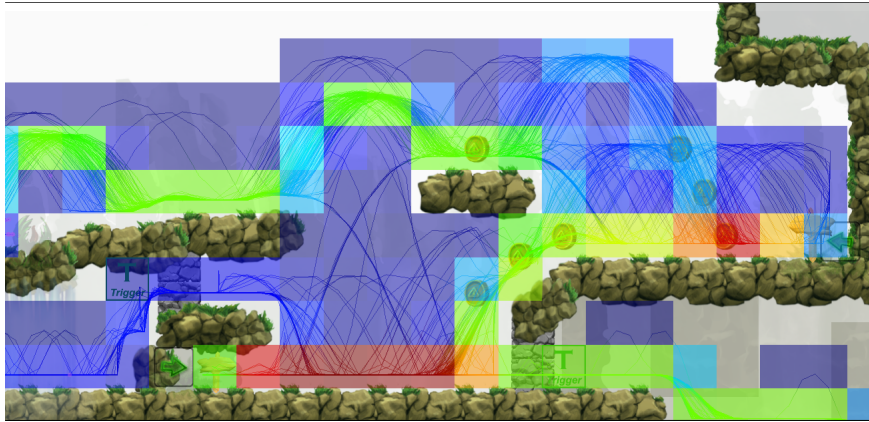
**Figure 6.5:** This question cluster includes an info text (the main question at the top), five checkbox questions and an optional text question.

**Figure 6.6:** If one or more required questions are not answered validly when the user presses the submit button, the background color of the required question(s) is colored reddish.

creation of all required GUI<sup>4</sup> components for this cluster including the submit button. When this button is pressed the `EvaluationHud` checks if all required components were answered validly. If not, the submission process is stopped and the required but not answered questions are highlighted with a reddish background color to inform the user about what went wrong (see figure 6.6).

**HeatmapManager:** The `HeatmapManager` is a very helpful tool for the visualization of the tracked metrics data. It can, for example, create a heatmap out of every position or game-over feature. Especially for the position feature, the possibility to display lines can create very meaningful results (see figure 6.7). These lines can be used to depict the paths which all players

<sup>4</sup>GUI is the acronym for graphical user interface.



**Figure 6.7:** This heatmap was created using position features. The lines visualize where the players ran.

went along. The colors for each line segment are derived from the created heatmap. The end of the lines can be marked by small purple points.

### 6.3 The Game: Elements

The tool was integrated into the already existing platform game *Elements* to make it possible to test if it can fulfill the requirements which were defined in section 6.1.

*Elements* is a 2D platform game which was created as a university project with three other students<sup>5</sup> in a previous term of studying (see figure 6.8). It was developed using the *xis-engine* (see section 6.2.1).

The goal of the game is to light the torch at the end of the level. At the time of the integration of the tracking tool, there were no checkpoints implemented for *Elements*, therefore players had to restart the level from the beginning whenever they died within a level.

The main character runs automatically. The user's controls are limited to jumping (pressing the space button) and switching the element. When the button **S** or the down arrow is pushed, the character changes to the element *stone*. By using the right arrow or **D**, the element can be changed back to *fire*. At the time of the evaluation, only these two elements were implemented but it was planned to also integrate the elements *water* and *air*. This was the main reason why there was not only *one* switching button but one button for each element.

Every element has its own properties. The fire runs faster and weighs less than the stone. This has an important impact on the jumping behavior:

<sup>5</sup>*Elements* was developed by Christoph Lipphart, Mark Mühlberger, Melanie Zeinlinger and the author in the course *Game Art and Level Design*.



**Figure 6.8:** In *Elements* waterfalls, stone crushers and killer plants imperil the player.

the fire can jump over wider distances, whereas the stone falls down more quickly. The fire is required for the end of the game where it has to light the torch but the stone is also necessary, as it can smash stone walls which can hinder the player from getting ahead in the level.

A death can be caused by different kinds of obstacles. Not every hazard is dangerous for every element; therefore one of the challenges for the players is to carefully choose the right element for every moment in the game. The obstacle *killer plant* kills both characters by eating them. The *waterfall* slacks the fire and the *stone crusher* smashes the stone (see figure 6.8). Both elements can die by falling down from the world, but the test levels which were used for testing (see chapter 7) did not include any place where this could have happened.

In the game world, *turn-around-triggers* were also placed which caused the element to change its direction. Most of them were indicated by graphical arrows.

In *Elements* it is also possible to collect coins. These are supposed to lead the players through the level [70] or motivate them to play the level again and explore it in more detail to find hidden places. Moreover, collecting coins can also just be fun [93].

For the user study, a new, invisible game object type was introduced. This was a trigger which was only used to create features to get information about how many players were at a specific location or how long it took users to get from one area to another.

## 6.4 Integration of the System into Elements

This section describes the integration of the system into *Elements* including the basic ideas, decisions, such as what to evaluate in detail, and the components that were necessary in order to create the final evaluation tool which was tested with a user study (see chapter 7).

### 6.4.1 Objective

To form the concept for the evaluation of *Elements*, it was necessary to define the goal of the game per se.

It was decided that the game should be *fun* which was of course too un-specific to be evaluable. The following passage tries to define the intended objective. Players should be challenged by levels of increasing difficulty. They should not be able to finish every level the first time they play it, but they should likewise not be so frustrated that they skip the level or stop playing. The levels should offer alternately difficult passages, which require a high level of concentration, and easier parts, where the gamers have the possibility to relax and enjoy the visual look of the game. To achieve an increasing difficulty, the complicated areas could be gradually extended or could provide ascending complexity. There should not be any position at which the main character could get stuck. Players should like the possibility of collecting coins. They should be willing to try a level several times or accept to make a detour in order to reach them. One of the main challenges of the game should be the timing. The controls themselves should not be experienced as confusing but rather as intuitive.

The objective of the user study was to *find problems in the level design*. Therefore, the next question was how to define *problems*. Several problems were collected which could be analyzed in the user study:

- The players are frustrated.
- The players are bored.
- The players are angry.
- The players are very stressed.
- The players are overstrained.
- The players do not understand what to do.
- The players do not understand the reason why they died.
- The players do not like the game.
- The players are not interested in finishing the level successfully.
- The levels are too difficult.
- The levels are too easy.

These problems are fairly common in usability research. In order to detect possible shortcomings, modern evaluation techniques such as heuristic

sets [18, 26, 39, 66] were analyzed. These proved to be a very valuable source. Additional information regarding different aspects of game design was provided by [73].

The definitions above represent the foundation for the integration of the tool and the evaluation design. They were taken into account when it came to important decisions such as what to track, where to ask questions or which questions to ask.

### 6.4.2 Architecture

When the evaluation tool was created, the core mechanics of *Elements* had already been developed. This made it necessary to include a new tool into the already existing code base, which proved to be not as easy as expected at the beginning. For example, another game state for asking questions had to be added. This state contained the GUI into which the `EvaluationHud` (see section 6.2.2) would insert all required information for the evaluation.

In the early prototype, there was no game state when the player died within a level. The current level just started all over again. This was a problem, as the system should be able to ask questions at this point within the game flow. Therefore, the complex restarting process had to be interrupted. If this possibility would have been created at an earlier stage in development, as it is recommended in [88], some problems and difficulties could have been prevented.

#### **ElemMetricsTracker**

To minimize the number of interfaces, the `service ElemMetricsTracker` was introduced. The idea was that it serves as the only interface between the metrics tracking system and the game. It is informed by the game of events relevant for the tracking and creates and adds all features. It derives from `trackable` (see section 6.2.2) and transmits a position feature every 100 milliseconds to the `TrackerManager`. The decision how often a position feature should be tracked was made based on trial and error. First the frequency was much too low, resulting in a heatmap such as shown in figure 6.9. By increasing the frequency, it was possible to create a continuous heatmap and precise levelplay lines such as shown in figure 6.7.

#### **TestSessionManager**

For the management of the test sessions, the `TestSessionManager` was developed specifically for *Elements*. It could be argued that this component should be included in the engine itself but this have may been too abstract to be useful for other games. To enable easy communication between the objects, it was, for instance, necessary to store and access some components of the code base of the game.



**Figure 6.9:** The frequency of tracking the position feature was not high enough for this heatmap to be continuous.

The `TestSessionManager` is a very important center point of the architecture. As the `ElemMetricsTracker` it is informed when a levelplay is ended and the reason (game-over, successfully finished, user input) why this had happened. It provides the pause and the levelplay-ended menu. This makes it possible to have an individual menu when a test session is active. In these menus, it is not possible to just go back to the level-selection game state. To do this, the players have to press the button to exit the test session instead (which results in a question being asked). Moreover, the menus are also modified by the `TestSessionManager`: When the last test level is being played, the button for skipping the level says *Skip Level and Finalize Test Session*<sup>6</sup>. Furthermore, this manager creates the unique player ID using the MAC address of the device and the timestamp. It is also responsible for starting and ending the test session.

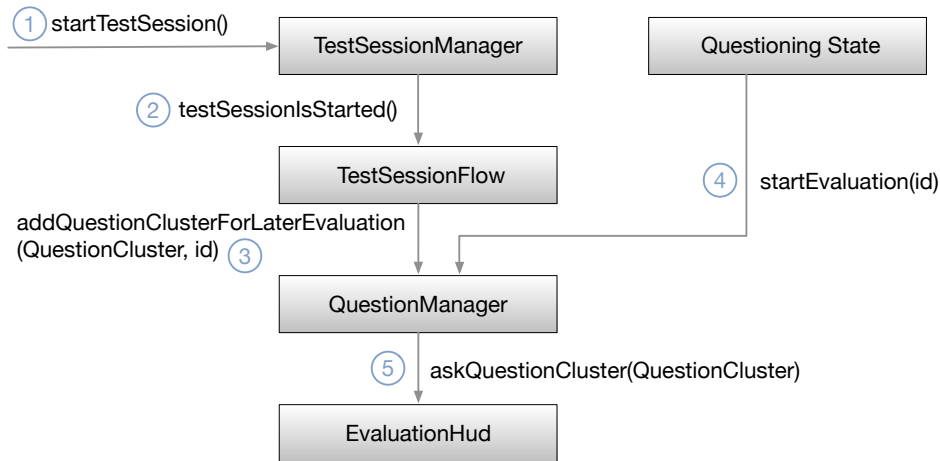
Having a control unit such as this makes it easily possible to ask questions at the intended moment. The `TestSessionManager` therefore starts the question asking sessions using the `QuestionManager`. But it does not know anything about when which questions should appear. This is the business of the `TestSessionFlow`, which is a member of the `TestSessionManager`.

### TestSessionFlow

The `TestSessionFlow` is informed by the `TestSessionManager` about the events that are relevant for the test session flow. These are when a level will start, when a level will start for the first time (in the current test session), when a level was finished successfully for the first time, when a level is skipped or when a test session is started, finalized or aborted. This makes it possible to react to these happenings by asking questions (see figure 6.10).

<sup>6</sup>The evaluation tool was created in German only. The original button name was *Level überspringen und Test abschliessen*.





**Figure 6.10:** This diagram roughly describes how a question can be asked at the beginning of the test session: (1) The game tells the `TestSessionManager` that a new test session was started. (2) This information is handed over to the `TestSessionFlow` which knows what `question clusters` to ask in this particular situation. (3) It adds them to the `QuestionManager` together with an ID. The `QuestionManager` stores them for a later evaluation. (4) At a later moment in time, the `questioning state`, which provides the necessary environment for asking questions, is opened. It orders the `QuestionManager` to start the evaluation using the ID. (5) Finally the `QuestionManager` hands the `question clusters`, which were stored in addition with this particular ID, sequentially over to the `EvaluationHud` which represents them to the user.

The `TestSessionFlow` is the component which knows when to ask which questions and which adds them to the `QuestionManager`.

Furthermore, the `TestSessionFlow` knows about the order and the IDs of the test levels. This information is requested by the `TestSessionManager`.

### Analyser

There is also one `analyser` implemented for *Elements*. It is responsible for the in-game analysis of metrics data. For this purpose it stores, for example, the history of all game-over features within one level. When a new feature is tracked, it gets informed by the `AnalysisManager` and checks if a question should be asked based on the new feature and the history. The algorithms which decide when questions should be asked are described in detail in section 7.1.4.

### ElemMetricsVisualizer

Another very important part is the `ElemMetricsVisualizer`, which is responsible for the visualization of already tracked data. It creates the menu for the data and visualization selection and contains the `HeatmapManager` and the `QuestionMapManager`. The `QuestionMapManager` is responsible for the visualization of questions which were stored with a dedicated position value. This makes it possible to visualize the positions where questions were asked.

### Statistics Creators

Furthermore, there are two classes which are responsible for the creation of statistics. The `MetricsStatisticsCreator` creates statistics out of the tracked metrics data while the `QuestionStatisticsCreator` uses the stored question data to create question-related statistics.

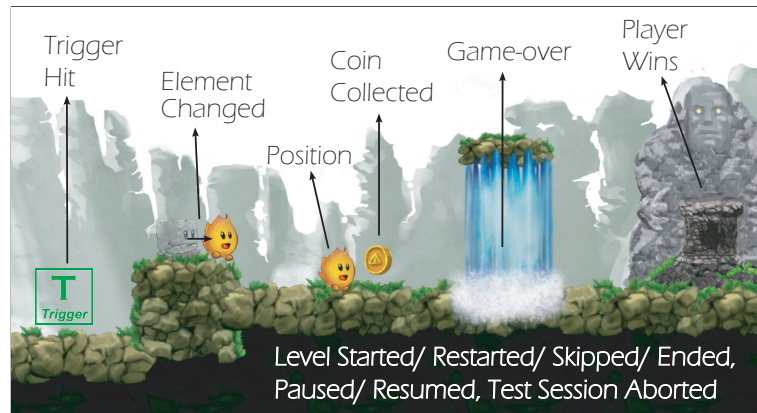
#### 6.4.3 Tracked Features

Based on the objectives (see section 6.4.1) and the game mechanics (see section 6.3) a number of features were selected for tracking (see figure 6.11):

- Level started
- Level ended: This feature also included the reason why it was ended (player died, won, exited the level using a menu button) and the number of coins which were collected in this particular levelplay.
- Level restarted
- Level skipped
- Player wins: This feature was tracked when the player reached the end of the level.
- Level paused or resumed
- Game-over: Supplementary also the reason causing the death was tracked.
- Position: Every 100 milliseconds the player's position and the current element was recorded.
- Element changed: This contained the information from which element to which other element the character had switched<sup>7</sup>.
- Trigger hit: When the player collided with an invisible trigger, the ID of the trigger and its position was stored additionally.
- Coin collected: This feature included the position of the collected coin.
- Test session aborted

---

<sup>7</sup>As there were only two elements implemented at that time, this additional information did not add any more value to the feature but it may be useful when all four elements are implemented.



**Figure 6.11:** This image visualizes the features which were tracked in the game *Elements*.

All but the level-started feature also stored the current player position as additional information.

#### 6.4.4 Asked Questions

The questions which were implemented in the evaluation tool were based upon the objective of the game and the evaluation (see section 6.4.1). They targeted potential problems concerning the player experience, the understanding of the game mechanics (see section 6.3) and the level design. The ideas therefore were inspired by several heuristics [18, 26, 39, 66].

The exact events which had to occur to result in questions being asked are listed in section 7.1.4, the entire list of questions which were asked can be found in the question list which supports this thesis<sup>8</sup>.

When a question was asked, not only the answer was stored but also some other information: For every question a unique ID was stored. For some questions it made sense to store also the position where they were asked<sup>9</sup>. Other additional information depended on the reason why the question was asked. When, for example, questions were asked because the player had collected lots of coins, it was stored how many coins were collected. For the questions which were asked when players died significantly often because of the same hazard, the obstacle causing the death was added.

<sup>8</sup>The question list is provided in German only, as the entire tool and the performed user study were performed in German.

<sup>9</sup>When a question was asked because the player had already died for several times in a certain area, the position was tracked. For demographic questions storing the position would have been of limited usefulness.

## 6.5 Conclusion

The developed tool makes it possible to track metrics data, analyze them directly in the game and ask questions at any time during gameplay. It was integrated into the already existing platform game *Elements*. This was much more work than thought at the beginning of the project. In addition to the system itself, much game-specific programming was necessary.

Moreover, tracking the data is only one part. Another very important task covers the analysis and visualization of the data. This is necessary because without meaningful statistics and diagrams, it is impossible or at least very time consuming and complicated to extract helpful information for improving the level design and the player experience of the evaluated game.

Based on the developed tool a user study was performed which is described in detail in the following chapter.

# Chapter 7

## User Study

To evaluate the developed system, it was tested by means of a user study which was performed in two steps: A preliminary study was used to find programming bugs and other major problems and then the updated program version was tested again in the main study. This chapter describes in detail the evaluation design, the problems which occurred and how they were fixed.

### 7.1 Evaluation Design

One of the main goals was that no supervisor was necessary for the evaluation. Therefore, all the required information about the tool, the game and what the testers were expected to do was provided directly in the application. Additionally, a sheet of paper<sup>1</sup> including the control information of the game was provided. This was done to help the players in case they forgot them during the test session, as there was no help button implemented in the game itself.

The objective was that the participants play three levels of the platform game *Elements* and answer the questions which were asked by the system at the beginning of the session, after a level and at the end of the test session (see figure 7.1).

#### 7.1.1 Test Levels

The three test levels, which had increasing degrees of difficulty, were especially created for the evaluation. Therefore, none of the participants knew the levels in advance, which could have influenced the results in an undesirable manner. Although it should be mentioned that some passages of already existing levels were used, partially as modified versions. But this was encountered as having no or at the most a very small impact on the

---

<sup>1</sup>In case of the evaluation by mail, which is described in detail in section 7.3.2, a digital *PDF* document was distributed instead.

study, as most of the testers had never played *Elements* before<sup>2</sup>. Moreover, the game had never been released beforehand, therefore for most of the people the *GameStage*<sup>3</sup> event on June 28<sup>th</sup>, 2013, was the only possibility at which they could have played the game earlier. This was nearly a year before the evaluation and the game was at that time at a very early stage. Moreover, they could only have tested it for a very short period of time of about a couple of minutes. Furthermore, the test levels of the evaluation had a completely new structure and what they could have seen was not very significant. Therefore, the fact that some may have played the game earlier had no or only a very slight influence on the user study.

### 7.1.2 Starting the Test Session

Each time a player pressed the button to start a new test session, a unique test session and player ID was created using the timestamp and the MAC address of the computer<sup>4</sup>. Subsequently, the players received some information including the following.

- The prototype was for testing the game *Elements*.
- They should just play the levels and answer the questions validly and they could not do anything wrong.
- In the background, data was tracked.
- Everything was saved anonymously.
- There was no possibility to go back to already answered questions.
- They could skip a level, but they should only do that if they really could not complete it.
- They could ask at anytime if anything was not clear.
- Basic instructions describing how the game was controlled.

After this information was given, some basic demographic questions were asked to find out the age of the participants, how often they played games and if they had already played the game *Elements*. Then the first level started.

### 7.1.3 Menu Options

To control the test session, the users were given some basic menu options. The pause menu contained buttons which enabled the players to either con-

---

<sup>2</sup>In the preliminary study, only three out of 14 participants declared that they had already played *Elements* in advance according to the analysis of the question data. In the main user study the percentage was with 15% (five out of 34 people) even smaller.

<sup>3</sup>More information about the *GameStage* can be found in section 7.2.1.

<sup>4</sup>In the preliminary study, only the timestamp was used for the creation of the session ID, as the function for getting the MAC address was not implemented in the game engine at that time.

tinue the game, restart the level, finish the test session or skip the level<sup>5</sup>. In the menu that appeared at the end of the levelplay (when the player died or finished the level successfully), the users could also end the test session, restart the level or move on to the next level (in case they had never finished the level successfully beforehand, this button was named *skip level* or *skip level and end test session*<sup>6</sup>, depending on the number of the level). It should be noted that, although it was possible to restart a level again, after it had been finished successfully, it was not possible to replay a previous level.

It was very important, that the users could skip a level, because for some participants the second or third level<sup>7</sup> could have been too difficult to master. This was also emphasized by the results of the studies, which are presented in section 8.1. The functionality of the premature finalization of the test session was created for people who had to stop the test, for example because they had to leave to catch their bus. This was especially important for the preliminary study at the *Ars Electronica Center* in Linz<sup>8</sup> (see section 7.2).

The target audience of the study included all people of at least 14 years of age, as in Austria, where this study was performed, this is the legal age to participate in a study without the requirement of a signature of a parent or legal guardian. The entire user study, including all questions, explanations and buttons was in German as for most participants this was most likely their native language. It was very important that they understood the questions properly and easily to receive correct answers. In this thesis some questions are mentioned. These are translated into English by the author.

#### 7.1.4 Triggering of Questions

Questions were asked either before a level was started or after the level, but not amongst playing, to avoid any disruption of the player experience during the level. Some questions were asked for each user equally in the same situation, for example the demographic questions at the beginning of the test session. Other questions depended on the player behavior which was measured using the tracked metrics data. The following list gives an overview over the moments, when questions that were independent of the metrics data appeared.

- At the beginning of the test session (demographic questions).

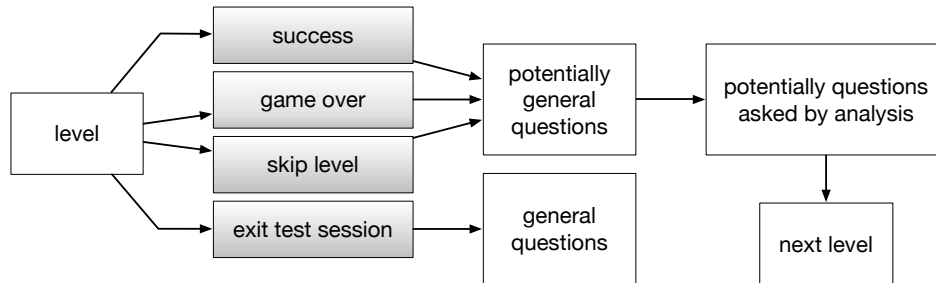
---

<sup>5</sup>The possibility to skip a level was not available in the pause menu of the preliminary user study.

<sup>6</sup>The original button names in German were *Level überspringen* or *Level überspringen und Test abschliessen*.

<sup>7</sup>It was not possible to die in the first level.

<sup>8</sup>But the possibility to abort the test session was only used two times at the preliminary study. One person aborted the test session having already tried the third level, in this case this can be equated with the skipping of the third level.



**Figure 7.1:** This flowchart illustrates the flow of questions during a test session.

- After a level was successfully finished or skipped: After the first and second level, questions about the controls of the game were asked. After each level the players had to vote how much they liked the level and how difficult it was for them.
- After a level was skipped: It was asked, for example, why they skipped the level, what their biggest problems were and if they were frustrated.
- If the test session was aborted, a text question gave them the opportunity to communicate why they had exited the test. It is important to mention that this question was optional, to make it also possible to validly exit the test, for participants who had to leave immediately.
- After the test session<sup>9</sup>.

The questions mentioned above were only asked at most once during the entire session or once for a level. When, for instance, a player would finish a level two times successfully, the corresponding questions would not appear for a second time.

### In-Game Analysis

The in-game analysis made it possible to ask questions based on the tracked metrics data. The basic attempt was to find problematic situations and positions in the game and levels. Section 6.4.1 defined how such problems could look like. As an example, the players might be frustrated, bored or overwhelmed. Based on these declarations, it was tried to derive some possibilities how these could be measured using metrics data. To do this, some general assumptions were proposed:

- When users die frequently, this could hint that they are overwhelmed because the challenges might be too difficult according to their current skills. This could lead to frustration and an increased stress level.

<sup>9</sup>These questions were not asked when the test session was aborted.



- When they die several times because of the same hazard, this could imply that they do not correctly understand the obstacle's mechanic.
- A position at which players die significantly more often compared to the overall level might be more difficult.

These propositions form the basis for the triggering of questions which were based on the metrics data:

**Died very often:** When during the entire test session for one level 10 or 30 game-over features were tracked, the players were asked questions about their current feelings. These questions could have appeared maximal four times, as they can be asked two times for one level, and there were two levels within which it was possible to die (level 2 and 3).

**Died often in the same area:** When players died five times in a level within an area with the radius of two units<sup>10</sup>, questions concerning their thoughts about this position were asked.

**Died often because of the same reason:** When users had died at least six times in a level and not less than half of the tracked game-over features were caused by the same obstacle, the players were asked if they understood why they had died. These questions were only asked once at a maximum for each obstacle during one test session.

**Collected lots of coins:** When participants collected more than 90% of all coins in a level, more questions appeared. There were two collections of coin questions: The first one was level specific, asking them if they liked the placing and the number of coins<sup>11</sup>. The second collection tried to find out how important and how much fun collecting coins was for the players. These general questions were triggered at most once per test session.

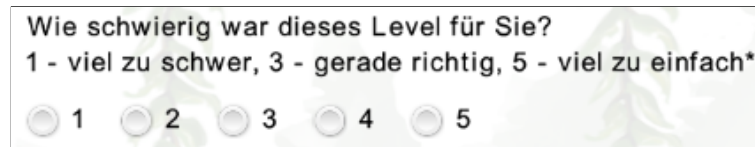
**Level was successfully finished often:** When players had already successfully finished a level three times, the system tried to identify the reason why they kept on playing the same level again and again.

The analyzed features were chosen very carefully. There were two very important parts which influenced the decisions of what to track and what to analyze: the goal of the user study, which was to find problems in the level design, and of course the game and its unique components, goals and obstacles. Examples for problems are when the players not understand what to do, got stuck or when the level was too difficult for them to complete

---

<sup>10</sup>The height of a platform is one unit.

<sup>11</sup>The question about the number of coins was included the first time in the main user study.



Wie schwierig war dieses Level für Sie?  
1 - viel zu schwer, 3 - gerade richtig, 5 - viel zu einfach\*

1  2  3  4  5

**Figure 7.2:** This radio button question asks *How difficult was this level for you?* 1 – *very difficult*, 3 – *just right*, 5 – *much too easy*.

or when playing was no fun for them. Therefore, the focus was highly on the game-over features, as these events are one of the most important and meaningful ones in this particular game.

The values for how often a feature has to occur to ask a specific question were selected depending on previous tests with the tool. While users were playing *Elements*, a supervisor had a look at the game and tried to figure out which values would create meaningful results. When questions would have been asked too often, their significance might be low and if they would have been asked too rarely it could happen that not enough data would be tracked to get helpful information for improving the game and the level design.

The mentioned questions and question topics represent of course not everything which was asked, but are only quoted to give some insight into the considerations and ideas behind what was tried to find out. More information about the exact questions can be found in the question list which is provided in addition to this thesis.

### 7.1.5 Types of Questions

For the evaluation, several types of questions were used. As described in chapter 6, the tool supports radio button questions, text questions and checkboxes. All of these types were applied, but it is also important to mention how they were used as they can be utilized in very different ways.

**Radio Button Questions:** Most questions were posed as radio button questions. This was done on purpose as they limit the number of different answer possibilities. This can shorten the answer time for the participants because they do not have to think for themselves what to answer, but they only have to select the best fitting answer. They also do not have to write the answer themselves. This makes them easier to compare, as there are not an infinite number of answer possibilities (as for example provided by a text box). Moreover, they cannot answer invalidly (assumed that the evaluation is digital and a control mechanism is used for checking).

Mostly the radio button questions provided four answer options. It would also have been possible to use five, but the middle option has less explanatory

Was bereitet Ihnen hier die größten Probleme (Mehrfachnennungen möglich)?

Wechseln zwischen Elementen

Sprunghöhe

Weite des Sprungs

Timing (ich drücke die Taste zu früh/spät...)

Ich verstehe nicht wie das gehen soll/ bin verwirrt.

Anderes ...

**Figure 7.3:** This question cluster includes several checkbox questions and an optional text question. It asks *What causes the biggest problems for you (multiple answers possible)?; changing between the elements; the height of the jump; the width of the jump; timing (I push the button too early/late...); I don't understand how this is intended to work/am confused.* The optional text field contains more space for *other* reasons.

power than the other ones, as it does not refer to any extreme, but may rather correlate to an undecided statement [8, p. 162]. But there were also questions in which five options were used, as it made sense to also provide a middle option for these. An example would be the question about the difficulty of the level, which is presented in figure 7.2.

Most of the questions had numerical answer options. The meaning of the numbers was explained in the question itself, as can be seen in the example. This was done to make them easy and quick to read. For some questions it was decided that it was more adequate to use textual options instead, for example when the participants were asked about their age. Some questions also provided the answer option *Don't know* or *I am not sure*.

**Checkbox Questions:** Checkbox questions were used to provide the possibility to check multiple options. They were, for example, adopted when the player was asked what caused the biggest problems (see figure 7.3).

**Text Questions:** Text questions represented the only possibility to get qualitative feedback and were also the only type which was sometimes optional. They were used to get some more detailed information and extend the answer possibilities. For example, when it was asked what caused the biggest problems, an optional text box enabled the users to list some more difficulties, which were not already covered by the provided checkboxes (see figure 7.3). They were also used to discover why a player had skipped a level or exited the test session prematurely.

This demonstrates that every question type can collect different types of information and therefore every single one of them can be very valuable

for a questionnaire. But it is not only important which question type is chosen but also *how* exactly it is used. There are lots of decisions which have to be made, such as the number of answer options or if an even or odd number of answer options is provided for radio button questions [8, pp. 162–163]. The formulation of the questions is also critical; including judgements can, for example, lead to biased results. Furthermore, the moment when the question is asked can also be relevant [2]. When there are lots of very personal demographic questions at the beginning, people may abort the study there, as they do not want to give away this information. But if the same questions are placed at the end of the evaluation they may not quit at this time [89, p. 176]. Therefore, it is necessary to choose and phrase the questions very carefully in order to get valid results.

## 7.2 Preliminary Study

The preliminary study was the first test of the game evaluation tool. Therefore, it was not astonishing that problems in the system were detected (see section 7.2.4). This enabled further improvements of the application for the main user study. The preliminary study took place at the *GameStage* event which turned out to be not the best environment for this particular user study as will be discussed in the course of this section.

### 7.2.1 GameStage

The *GameStage* is a public event which takes place several times a year at several locations in Linz, Upper Austria, Austria. The purpose of this free event is to communicate the variety of computer games and to connect people who are interested in games and game developers in Linz and Austria [91].

On May 16<sup>th</sup>, 2014, when the preliminary study was performed, the event took place at the *Ars Electronica Center*<sup>12</sup> (*AEC*) in Linz. In one room, a couple of games, including the described game evaluation application, were exhibited and could be tested by the visitors. The system ran on one computer with a mouse, a keyboard and an external monitor. Additionally, a sheet of paper, explaining the controls of the game, was placed on the desk. People could come and sit down to play and test the game, but it was also possible to just stand behind the player and watch him or her play the game and do the evaluation. Figure 7.4 illustrates the test situation.

### 7.2.2 Advantages of the Setting

Using a public event as the stage for the evaluation had some very significant advantages. First of all, the organizational effort for this evaluation was

---

<sup>12</sup>[www.aec.at](http://www.aec.at)



**Figure 7.4:** This image was taken from [102] and was shot at a *GameStage* event on June 28<sup>th</sup>, 2013, at the *Ars Electronica Center*. The test environment of the preliminary study was nearly identical to the one shown on this picture.

minimal. Most of the equipment (including table, chair, monitor, mouse and keyboard) was provided by the organizers, who also promoted the event. The *GameStage* is already known by people as already six previous *GameStage* events took place since the year of 2013 [104]. Therefore, it was well attended with approximately 180 visitors. Another advantage was that the participants were physically there, which made it possible to directly talk to them to receive even more feedback (see section 7.2.6).

### 7.2.3 Problems with the Setting

Although it was a good test environment, it did not prove to be the perfect one for this particular evaluation. One problem was that other people were watching while standing behind or beside the player. Therefore, they saw many things before they were able to do the evaluation themselves. They saw the levels, the mistakes the player made and were able to identify difficult areas. This way it is possible that they have already learned how to play the game before trying it on their own. Moreover, they saw the questions which appeared and also the answers the user gave. This potentially had an effect on the provided answers, as the players may not have felt comfortable writing about their problems and mistakes in a rather public environment. But it should also be mentioned that no one ever complained about that.

Moreover, there was also some time pressure, as there were more people who wanted to test the game. Therefore, the players could not just play the

game for as long as they wanted or needed to finish all three levels. This pressure was surely also intensified by the presence of the others around the player.

Furthermore, some people were not interested in evaluating the game for a longer period of time. Maybe they did not understand (or were not interested in) how the user study was intended to be. Indeed, there was an explanation at the beginning of the test session, but it was observed that some just skipped it without reading it (in detail). Some of the comments people added when being asked why they skipped the level indicate such behavior: Someone explained that he wanted to see the new level<sup>13</sup> and another person declared that it was time to explore other games<sup>14</sup>.

Subsequently, it could be inferred that some people just had other expectations when they tried the game evaluation system. This would not be astonishing at all, as the purpose of the *GameStage* is to present games and not to provide an environment for scientific user studies as already mentioned in section 7.2.1.

#### 7.2.4 Detected Problems in the Tool

As this formed the first test of the game evaluation tool, some bugs in the program were found during or following the user study. The major problems are listed in table 7.1.

#### 7.2.5 Execution

During the user study, it was discovered that there was a much too difficult passage in the second test level. As people had problems to overcome this area, it was changed during the study. Figure 7.5 shows the differences between the two versions<sup>15</sup>.

#### 7.2.6 Further Improvement Opportunities for the Tool

The test not only revealed problems, but also other improvement opportunities for the test application. These were found out by talking to the test subjects.

Some of the players mentioned that they would have liked it better if the options of the radio button questions were sorted from 1 to 4 instead of the reverse order, which was used by the tool. To make it more clear what is meant, the question in figure 7.6 will be used as an example for

---

<sup>13</sup>The original answer was *I wanted to see the new level*. This participant did not speak German. The evaluation was done by having the supervisor sitting next to him and translating the questions and answers for him.

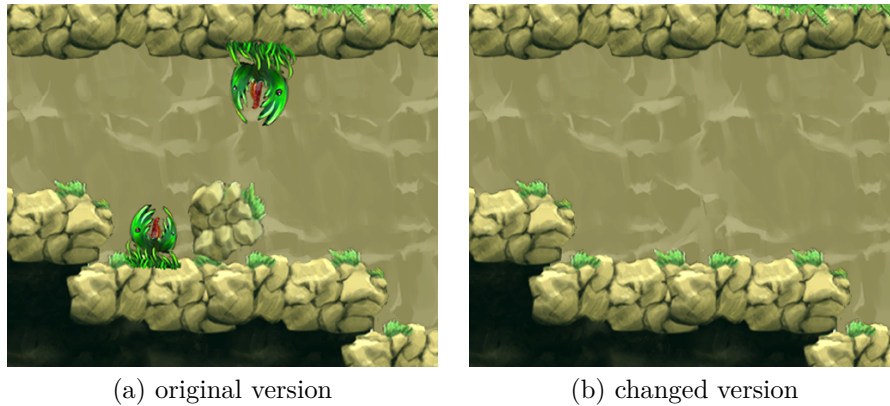
<sup>14</sup>The original German answer read *Zeit andere Spiele zu erkunden*.

<sup>15</sup>Of course it was documented which test users had played the difficult level and who had used the easier one.

**Table 7.1:** This table lists problems which were found in the preliminary user study.

| ID | Description   |
|----|---|
| P1 | The questions that should have been asked at the end of the test session, did occur at the end of the first level of the follow-up test session.  |
| P2 | The wording of some questions led to confusion on the part of some test users. There was, for example, the question <i>Are you confused?</i> This question was asked when a player died several times (see section 7.1.4) because of the same obstacle. The original purpose was to find out if the players understood why they had died, but without the context, the question was not clearly understandable to some of them. |
| P3 | There was an error in the question <i>How old are you?</i> It was a radio button question with the answer options <i>14–17, 18–24, 25–35, 35–50</i> and <i>50+</i> . The problem was that the age of 35 was listed two times.   |
| P4 | Sometimes when a user died, the program did not reset the world correctly. Therefore, the player immediately died again when the next levelplay was started. This resulted in the tracking of more than one game-over feature.  |
| P5 | In the application sometimes more than one level-ended feature was tracked. This was no big problem for the analysis and did not distort the results, but had to be fixed for the final test version.   |
| P6 | As already mentioned in chapter 6, each question had its own unique ID to simplify the creation of statistics of specific questions. It turned out that for some questions this ID was missing.   |
| P7 | There were also some cosmetic problems with the spelling of the questions such as a letter being unintentionally written as upper-case or lower-case, an $\beta$ which was depicted as $\beta$ because of a character set problem, or a missing line break. Sometimes the title of a question cluster was named <i>QUESTION</i> instead of the German word <i>Frage</i> .   |

further explanations. The original idea was that they should tick a higher number the more they liked the level. But when talking to them, most of them referred to the Austrian grading system which is used in school. There



**Figure 7.5:** During the preliminary user study, the second level was slightly changed in one very difficult area. (a) represents the original version of this region and (b) the same passage, after it was altered.

Wie gut hat Ihnen dieses Level gefallen?  
 4 - sehr gut, 1 - gar nicht\*

4    3    2    1

**Figure 7.6:** This question says *How much did you like this level?* 4 – very much, 1 – not at all.

1 is the best grade possible whereas 5 is the worst. This seemed to be more common to them than the original interpretation. It would not have influenced the meaning of the question, when the order would have been changed, but it maybe would have helped the users to quickly understand and answer the questions. Therefore, it was decided to change this for the final study.

A couple of users did not like that they had to begin a level from the very beginning on when they died. The possibility of checkpoints was often proposed as a potential solution. Some also mentioned that they had not expected the obstacle plant to kill them regardless if they were stone or fire<sup>16</sup>. For some players it was confusing that the main character sometimes changed its direction when there was no graphical indicator such as an arrow. They suggested to place such a hint at every position where the character turned around.

<sup>16</sup>There were expectations such as the fire burning the plant or the stone squishing it.



### 7.2.7 Data Cleaning

To use the collected data, it had to be cleaned to remove the errors which were listed in section 7.2.4. This section describes for some of the problems how they were fixed or which effects they had on the data.

In general, the data was analyzed and fixed by iteration over the files. A simple program analyzed each levelplay and all answered questions and changed them if necessary.

The missing question IDs (see problem P6 in table 7.1) were easily reproduced using the wording of the questions themselves to identify which question it was. This was possible as each question was only asked because of one specific reason. If the application would have asked the same question for several purposes (for example at the end of the level as well as when a player died at a specific location) it would have been more complicated, but also possible because the program also saved for each question more information such as *why* or *where* in the level it was asked.

As the questions that should have been asked at the end of the test session were not asked at the correct time, this data was not valid and could not be used for the analysis (see problem P1 in table 7.1).

The problem that in the demographic question about the age of the player, the age 35 was listed in two answer options (see problem P3 in table 7.1) could not be fixed. Therefore, this data may not be clearly interpretable, but this is counted as no big issue. Maybe there is also no problem at all, depending on if any player was exactly 35 years old, but unfortunately it cannot be said anymore.

As the analysis system was not ready developed when the preliminary user study took place, the examination of which data structure would fit best, was not finished. Therefore, it was necessary to slightly restructure and change the files afterwards.

## 7.3 Main Study

For the final study the tool was revised. The problems and bugs found in the tool during the preliminary study (see section 7.2.4) were corrected and some other minor changes were made. The system was for instance changed to make any cleaning process (see section 7.2.7) unnecessary. The main menu screen was modified: For the second study, it only provided a button for starting a session and some short information which told the participants to only do the test once and in the tool for the mail study (see section 7.3.2) it also reminded the user to send the tracked data after the test by mail to the test manager.

Ist diese Stelle unlustig?\*

sehr unlustig    eher unlustig    eher lustig    sehr lustig

**Figure 7.7:** This question asks *Is this area no fun? no fun at all, rather no fun, rather fun, very much fun.*

### 7.3.1 Changes in the Questions

Most of the changes affected the questions. Nothing was changed regarding *when* the questions were asked, but only their spelling and formulation. Of course, the cosmetic problems mentioned in section 7.2.6 were fixed and the questions which led to confusion (see section 7.2.4) were reformulated. The question *Are you confused?*<sup>17</sup> was, for instance, replaced by the formulation *Do you find it confusing that you died at this obstacle with the current element?*<sup>18</sup>. Other questions, concerning the feelings of the players, were also changed such as *Are you stressed?*<sup>19</sup> which read *Do you feel currently stressed?*<sup>20</sup> in the second evaluation.

The wish of some participants to change the order of the radio button answer options to begin with the smallest number, as described in section 7.2.6, was fulfilled.

One question was added asking the participants to rate the number of coins if they had collected more than 90% of all coins in a level (see also section 7.1.4).

For a couple of radio button questions, the answer options were reduced from five to only four. This was, for example, the case for the question for which the participants had to vote how much the statement *In every level, I can discover something new* applied to them<sup>21</sup>.

For the question whether an area was no fun, the number options were replaced by text options with the same meaning, in order to make it more clear what was meant even if the users did not read the question completely (see figure 7.7).

There were also some other minor changes. But these were rated as having (nearly) no influence on the outcome of the evaluation as their meaning was not significantly changed.

<sup>17</sup>The original German question was *Sind Sie verwirrt?*

<sup>18</sup>The original German question was *Finden Sie es verwirrend, dass Sie mit dem aktuellen Element bei diesem Hindernis gestorben sind?*

<sup>19</sup>The original German question was *Sind Sie gestresst?*

<sup>20</sup>The original German question was *Fühlen Sie sich gerade gestresst?*

<sup>21</sup>The original German statement was *In jedem Level kann ich etwas Neues entdecken.*

### 7.3.2 Setting

It was possible to participate in the study in two different ways: Either by coming to the evaluation at university on July 8<sup>th</sup>, or July 9<sup>th</sup>, 2014, or by doing the test at home and sending the data produced by the system via mail to a dedicated mail address.

The invitation was sent by mail to all students of the degree programs *Interactive Media*, *Digital Arts* and *Media Technology and Design* of the *University of Applied Sciences Hagenberg*. It was also posted on *Facebook*<sup>22</sup> visible for students of the *University of Applied Sciences Hagenberg* and the *Facebook friends* of the author<sup>23</sup>. As an additional inducement two *Amazon*<sup>24</sup> vouchers, each worth 10 €, were given away in a drawing.

#### Testing at University

For the test at university, the participants had to enter their names into a time slot in an online *Google Spreadsheet* first. It was possible that two players played the game simultaneously, as the test application ran on two computers<sup>25</sup>. In case they forgot the controls while playing, a sheet of paper providing this information lie on the desk. In contrast to the preliminary evaluation, a maximum of four people were in the room while the participants played. No one but the tester looked at the screen and it was very quiet. The time slots were chosen very generously. Therefore, everybody had more time than required for the test.

Only six people<sup>26</sup> chose this method of participation.

#### Testing by Mail

The application for the mail version of the evaluation had to be downloaded using a provided link. It was delivered as a *zip* archive file which also included a digital *PDF* document with the instructions and the game controls. The tool was provided for computers running the operation system *Microsoft Windows*<sup>27</sup> only.

In order to participate in the evaluation, the players had to start the application, create and finish the test session, *zip* the folder *evaluation*, which was created by the program, and send the file via mail to the dedicated<sup>28</sup>

---

<sup>22</sup>[www.facebook.com](http://www.facebook.com)

<sup>23</sup>Only students were invited to the user study at the *University of Applied Sciences Hagenberg*.

<sup>24</sup>[www.amazon.com](http://www.amazon.com)

<sup>25</sup>Between the two test stations a separating wall ensured that no player was able to look at the other's monitor during testing.

<sup>26</sup>In total, 34 people took part in the user study.

<sup>27</sup>[windows.microsoft.com](http://windows.microsoft.com)

<sup>28</sup>The mail address was noted in the invitation mail, the *Facebook* invitation post, the instructions and on the main menu screen of the application itself.

mail address.

An advantage of the mail evaluation was that the participants were able to play the game whenever and wherever they wanted to. There was no dedicated test supervisor, but it cannot be identified if they were supervised or disturbed by another person. This means a loss of information about the actual test situation. But it can be assumed that most of them have played the game in their own chosen environment such as their homes.

## 7.4 Lessons Learned

It proved to be very helpful to make a preliminary user study. Lots of problems concerning the tool and the question formulation, which proved to be very critical, were found this way.

One participant of the study mentioned that it *hurt* him a little bit that he had to tick the age option *25-34*. Maybe it would have been – psychologically – better to cluster the ages slightly differently, meaning that the age *25* would be the last number of an option instead of the first one (*19-25* and *26-35*).

## 7.5 Conclusion

This chapter described the evaluation design and the two studies. Using the preliminary study, it was possible to detect several problems in the tool which were subsequently fixed to maximize the quality of the data produced by the main study. This evaluation included a mail test version which allowed people to participate independent of time and location. In the following chapter the outcome of the main study will be presented and discussed to evaluate the proposed system.

# Chapter 8

## Evaluation

This chapter deals with the results of the main user study which was described in detail in the previous chapter. The first part analyzes the results and represents how they can be used to improve the game. Subsequently, some further ideas for the system are proposed. Finally, it is discussed if this evaluation method was able to fit the requirements which were specified in section 6.1.1.

### 8.1 Results of the User Study

The created evaluation tool (see chapter 6) combines metrics tracking with questionnaires. The tracked data was used to generate different kinds of outputs such as statistics or heatmaps. It would go beyond the scope of this thesis to list all results<sup>1</sup>; therefore only some of them will be discussed to provide some insight into the possibilities this system offers.

**Note concerning the diagrams:** The majority of the diagrams in this chapter was created using the information of the answered questions of the main user study. Frequently the translated<sup>2</sup> question will be used as heading. Many times in the original user study, a scale from 1 to 4 or from 1 to 5 was used as answer options for radio button questions. To avoid any disarray and confusion with the result numbers, it was decided to use alphabetical descriptions (from *A* to *D* or from *A* to *E*) in this chapter instead.

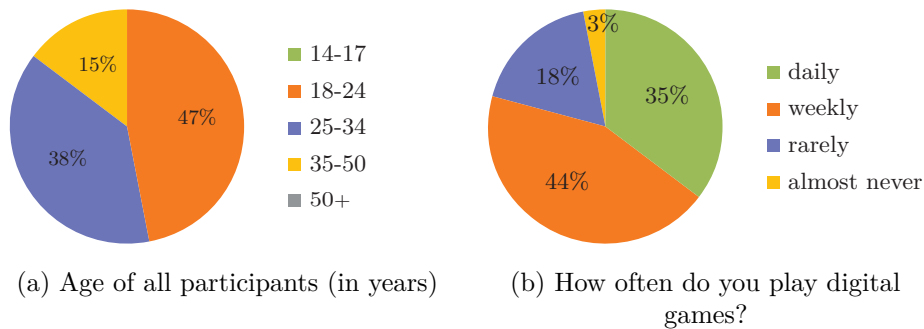
#### 8.1.1 General

In total, 34 people participated in the user study. Most of them (28 people) chose to evaluate by mail (see section 7.3.2).

---

<sup>1</sup>As an example, in total nearly 60 different questions were used for this evaluation.

<sup>2</sup>The user studies were performed in German. The translations were made by the author.



**Figure 8.1:** The demographic questions revealed that the participants were relatively young and that most of them played digital games frequently.

About 85% of the participants (29 people) had never played the game in advance. This was desirable as they therefore had less knowledge about the game and had never had the possibility to learn the game's controls in advance which could have had an effect on the results.

The players were mainly young adults. More than 75% percent played digital games frequently (see figure 8.1). This indicates that the study had reached the target group of the game (which consists of gamers). If the participants had consisted of people who never play games, the usefulness of the results for further development and improvements of the game would have been questionable.

### 8.1.2 Test Session Aborted and Level Skipped

None of the 34 participants exited the test session prematurely. This was very pleasing, as in the preliminary user study (see section 7.2) two out of 14 aborted the evaluation. Furthermore, also the percentage of players who skipped levels decreased<sup>3</sup> (see figure 8.5). It can be suggested that this was caused by the different setting (see section 7.3.2). But this time also one participant skipped a level (level 2) because he or she just did not want to play the entire game but only test it<sup>4</sup>.

One person skipped the first level in which it was not possible to die. This player tried five times to play the level but then was very frustrated and gave up stating that the character always got stuck. There was also a second person who played this level two times until succeeding. Three others played it again for several times, one of them even five times. The reasons this user mentioned for this were the wish of collecting all coins and because

<sup>3</sup>In the preliminary study no one skipped the first level, four participants jumped over the second level and nine players never finished level 3 successfully.

<sup>4</sup>This was the answer given for the open text question which asked why the level was skipped.

it was fun.

Three players skipped the second level noting that they were a little bit frustrated. Two of them tried it only three times (see figure 8.2). Both of them also jumped over the third level. Interestingly, the third participant tried the level 22 times before unintentionally pressing the skip button<sup>5</sup>. These results suggest that the second level would have been feasible for all participants. But it also has to be taken into account that most of them played games frequently (see figure 8.1 (b)). Therefore, it can be assumed that in general they already had some practice concerning video games (although the type of games they usually play is not known).

The third test level was skipped by nine people. Six of them tried the level at most five times (see figure 8.2) until they finished the evaluation. All of them also played the second level for a maximum of six times which is relatively rare when concerning that the average number of times a player had to play this level until finishing it successfully was about seven times. Most of the people who skipped level 3 (six participants) were a little bit frustrated. Only one person was very frustrated and complained that the level was too difficult to solve it without any checkpoints<sup>6</sup>. Two players were not frustrated at all, one of them had only tried it for three times and the other one had unintentionally hit the button<sup>7</sup>.

When it came to voting the difficulty of this level, only one of them said that it was much too difficult and no one liked the level *not at all*.

This analysis suggests that the levels were not too difficult for the majority of the study participants. It was aimed to find out what their biggest problems were. In the open text questions, several issues were mentioned frequently:

- They complained that there were no checkpoints.
- They had problems with changing between the elements.
- They claimed that they did not have enough time to react. Reasons mentioned for this were that the speed was too fast and that they saw the obstacles very late.
- Some people also claimed that they had no more time<sup>8</sup>.

This feedback provides some valuable information about how the game could be improved.

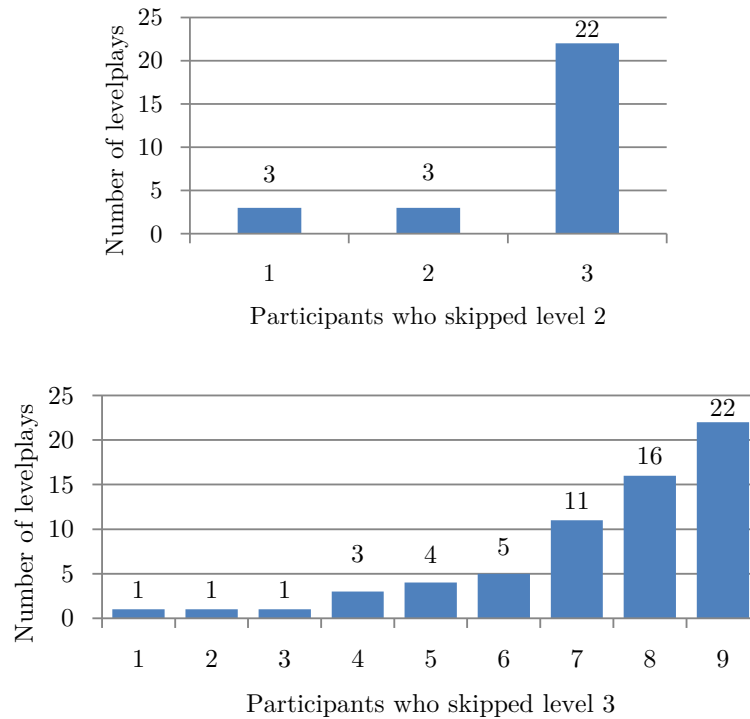
---

<sup>5</sup>This was the answer given for the open text question which asked why the level was skipped.

<sup>6</sup>This participant tried this level 11 times.

<sup>7</sup>This was the reason mentioned in an open text question.

<sup>8</sup>The time how much they had invested varied in this case between approximately 5 (shortest session time) and 19 minutes.



**Figure 8.2:** This diagrams visualize how often the players who skipped the second or the third level played these levels. Half of the participants who skipped level 2 and three had tried it only for a maximum of three times.

### 8.1.3 Difficulty

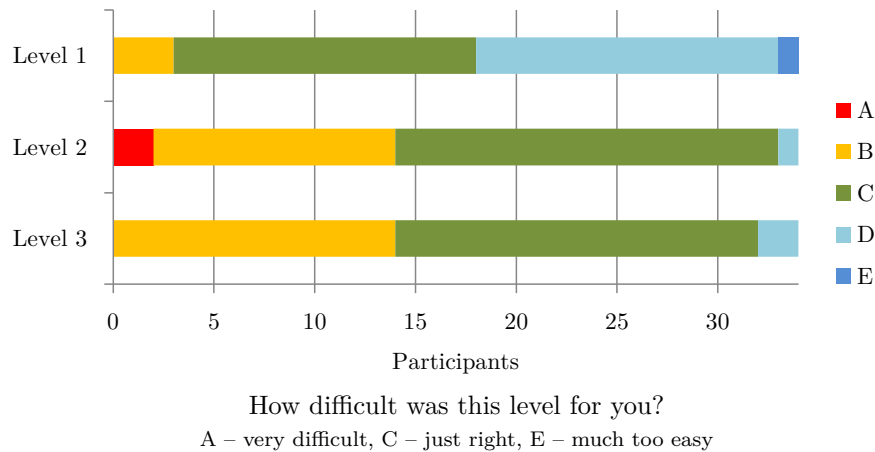
After analyzing the problems which occurred for participants who skipped a level, the focus will be on the overall difficulty. First, it will be analyzed if the difficulty was actually increasing between the different levels as this was intended by the user study. Afterwards, the biggest problems are analyzed and subsequently the outcome of the *difficult area questions* will be described.

The difficulties concerning the controls are described in a separate section (see section 8.1.6). Furthermore, also the feedback users provided offers valuable clues concerning this topic (see section 8.1.10).

#### Level Difficulty

This section deals with the overall difficulty between the three test levels. The analysis of the diagram about the difficulty of the various levels clearly states that the first level was not very difficult (see figure 8.3). But it is not clearly evident if the third level was the most difficult one (which was





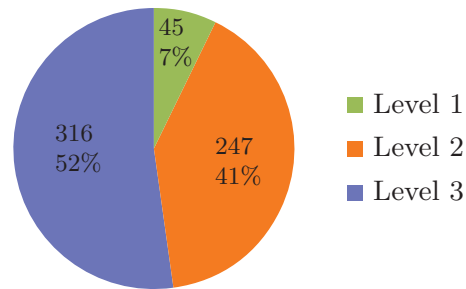
**Figure 8.3:** The first level, in which it was not possible to die, was experienced as being the easiest while this chart raises the question if the third level was more difficult than the second one.

intended). It seems as if the second and the third level were approximately equally difficult. But it has to be mentioned that the chart was created using the *subjective opinions* of the users who played them, always in the same order. Therefore, they saw the second level first and learned to survive the different obstacles there. They expanded their skills in playing this game and maybe this was one reason why they thought that the third level was not so difficult.

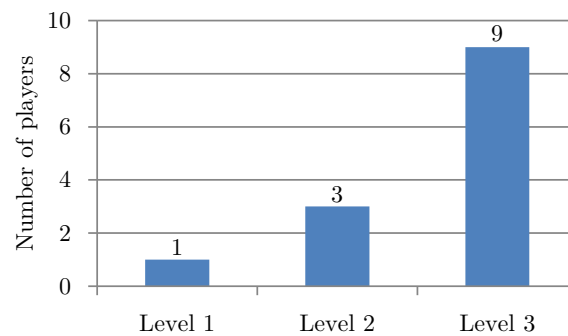
But on average they needed more tries until they finished the third level than they needed for the second<sup>9</sup>. Furthermore, most of the levelplays are from the third level (see figure 8.4) and the ratio between succeeding levelplays and levelplays which resulted in game-over events is also slightly higher for the second level<sup>10</sup>. Moreover, the third level was also skipped by approximately every fourth person which is three times as often as the second level was skipped (see figure 8.5). But it has to be noted that the reasons for skipping may not (always) have been the difficulty (see section 8.1.2). Nonetheless these values indicate that the third level was indeed more difficult than the second one.

<sup>9</sup>On average the players had to play the second level about seven times and the third one ten times until they finished it successfully.

<sup>10</sup>In level 2 the probability of winning versus dying was 13.28% (209 levelplays resulting in game-over and 32 successfully finished levelplays) while for the third level this rate was only 8.41% (283 game-over and 26 winning levelplays). The number of manual restarts (most likely caused by getting stuck) was thereby not taken into account.



**Figure 8.4:** This chart shows the number of levelplays separated by level.



**Figure 8.5:** This diagram visualizes the number of players who skipped the levels.

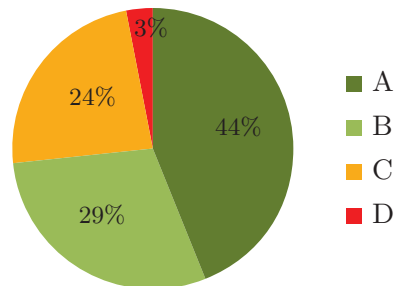
## Problems

Almost more important than the difficulty value is what caused the levels to be that hard. About a quarter of the participants answered that the game could not be played pleasantly and with uninterrupted flow (see figure 8.6). Some people noted in the open text questions that the character sometimes got stuck; perhaps this caused this high value.

More than a third claimed that they had problems with changing between the elements (see figure 8.13). This issue was also mentioned by players who skipped a level (see section 8.1.2) and as additional feedback some people noted that they would suggest to only use one button for changing the element (see section 8.1.10).

## Difficult Area Questions

When the user studies were developed, it was suspected that the overall level difficulty value could be increased by a few very complicated areas. Therefore, questions referring to the particular location were asked when a



The game can be played pleasantly and with uninterrupted flow.

**Figure 8.6:** Only about three quarters of the participants experienced the game as being a pleasant and flow-like experience.

person died several times in the same area (see section 7.1.4).

In total, these questions appeared 12 times in the whole user study. It was expected that they would have been raised more often but this also has some explanatory power. Perhaps the difficulty of the levels was rather balanced, meaning that the challenge of the game was not caused by one particular part but having to finish the entire level at once without dying at anytime as there were no checkpoints.

Due to this small number, it is not possible to draw significant conclusions. But nonetheless, some information may be useful when being combined with different data.

The questions were asked at six different obstacles. All of them were placed relatively early in the levels. This correlates with the number of times players were at these locations<sup>11</sup>. At four positions, only one person was asked, at the other two in each case four players answered these questions. It is not astonishing that these two areas were the locations at which most players died in the second and in the third level.

Next, some statistics of the most difficult location (see figure 8.7) of the third level will be discussed. Although the four players had died there already five times, they found this place rather fun. No one was of the opinion that this position should be removed or that it would ruin the fun of the game.

The statistics created using all position questions deliver similar results (see figure 8.8). The areas were not experienced as being very difficult or more difficult than the rest of the level. Most of the players did not even wish for an alternative path. In general, the locations were perhaps frustrating but not a complete fun killer. Again it should be noted that the number of data and therefore their explanatory power is very limited.

<sup>11</sup>Every levelplay starts at the beginning and ends sooner or later on the path to the goal. This issue also occurred for heatmaps (see 8.1.8).



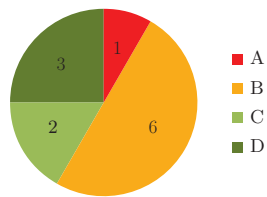
**Figure 8.7:** This image shows a part of the game-over position heatmap of level 3. The locations at which questions were asked because participants had already died several times in this area (see section 7.1.4) are highlighted with red, square outlines. The number in the middle of the squares describes how often such a question was asked there.

#### 8.1.4 Level Popularity

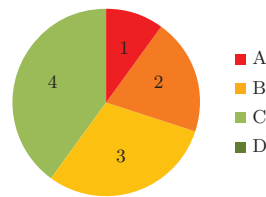
After discussing the difficulties, another very important aspect will be analyzed: to what extent the players actually liked the levels.

Figure 8.9 reveals that nearly all participants liked all the levels. But there were also two players who claimed that they did not like a level at all. This raised the question if this was the same person. Therefore, the `QuestionStatisticsCreator` (see section 6.4.2) was extended with the possibility to save supplementary information, such as the player ID, in addition to the question statistics. This revealed that these were two different users. Their tracked data was analyzed to gain more information about who they are and why they did not like the levels at all. In the following section, the results of the analysis are presented. In order to increase the readability, fictitious names are assigned to the player IDs.

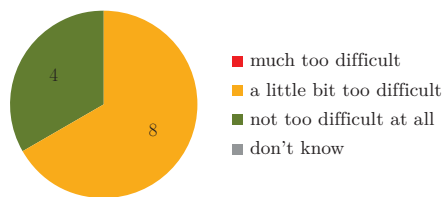
The participant who did not like the first level at all will be called Emily. She is between 25 and 34 years old, plays video games weekly and had never played this game in advance. One question which came up was if the level



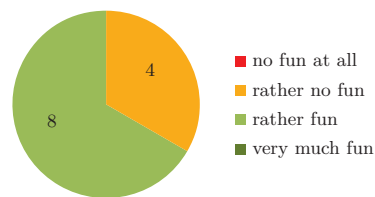
Is this area frustrating?  
 A – very frustrating,  
 D – not frustrating at all



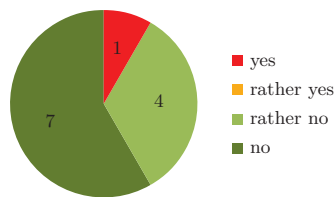
Is this area too cruel?  
 A – much too cruel,  
 D – not cruel at all



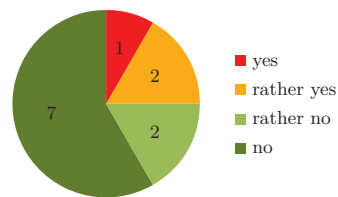
Is this area too difficult?



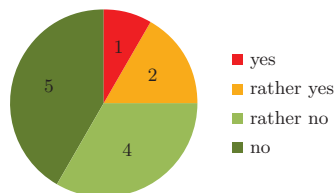
Is this area no fun?



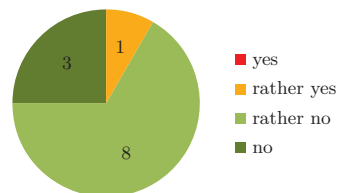
This area should be removed.



This area destroys the fun.



It should be possible to avoid this area (alternative route).

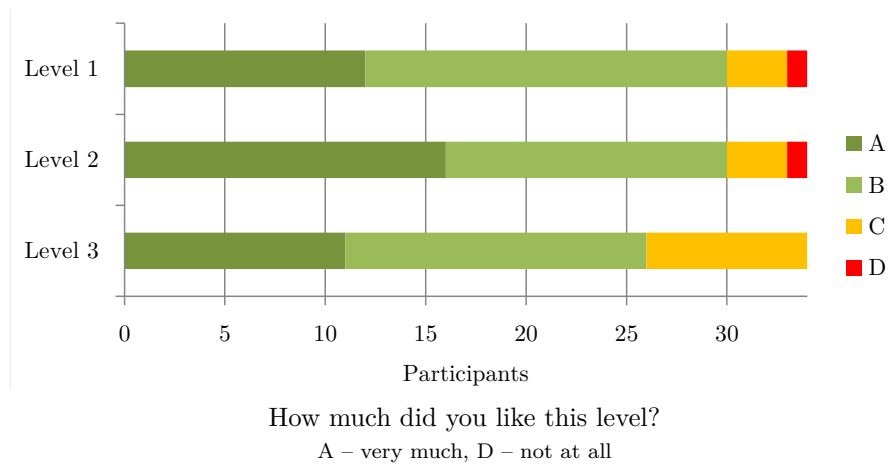


This area is much more difficult than the rest of this level.

**Figure 8.8:** These statistics were created using the answers of all 12 times when the questions concerning a *difficult position* (see section 7.1.4) were asked. It should be mentioned that these questions referred to different locations. Furthermore, the number of data is not enough to derive clear statements but they can be used to fortify other assumptions.

was too difficult for her, but this seemed not to be the problem: the difficulty of the first level was just right (according to her answer). Furthermore, also the other levels were not very difficult for her<sup>12</sup>. She was also able to finish

<sup>12</sup>The second level was rather difficult for her and the third one again just right.



**Figure 8.9:** In general the participants liked all three levels.

all the levels even faster than the average<sup>13</sup>. The next idea was that the controls may be to blame. The results from this analysis are ambivalent; On the one hand, she answered that they are not complicated, work rather well and are rather good to recognize, but on the other hand she claimed that they are not intuitive at all and rather confusing. But she also said that she coped rather badly with the switching between the different elements. Therefore, this may be one issue. Moreover, she also found the levels not very interesting. She answered that they become boring after some time, that she would like some more variety and that she *could not discover something new in every level*<sup>14</sup>. In a comment about the first level, Emily also complained about the fact that she could not go to the right and left by herself. It also took her a little bit longer than the average player to finish this level<sup>15</sup>.

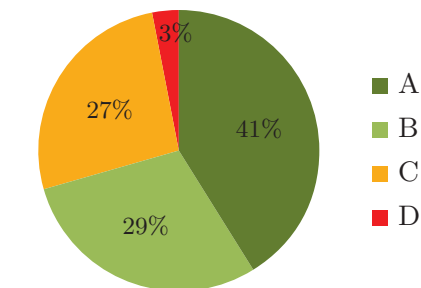
There was one passage in the first level at which several participants ran from left to right a couple of times until they were able to jump at the right time to cross the gaping which separated them from the next part of the main path through the level. When they did not jump at the correct time, they had to move relatively long until they again had the opportunity to try the jump. If they had had the possibility to turn around by themselves, this punishment would have been smaller.

Perhaps running often from left to right was the reason why it took Emily longer to finish this level and therefore she was probably angry about

<sup>13</sup>She finished the first level at the first time, needed only five tries for the second level and three tries for the third level to finish it successfully while the average player would have to play the second level seven times and the third level ten times until succeeding.

<sup>14</sup>This was a question at which the participants had to select one option between 1 (yes) and 4 (no).

<sup>15</sup>She needed about 2.2 minutes for her only levelplay in the first level while the average length of a level 1 levelplay was 1.85 min (median: 1.72, min = 1.04 min, max = 3.5 min).



How much do you like the dark caves?  
A – very much, D – not at all

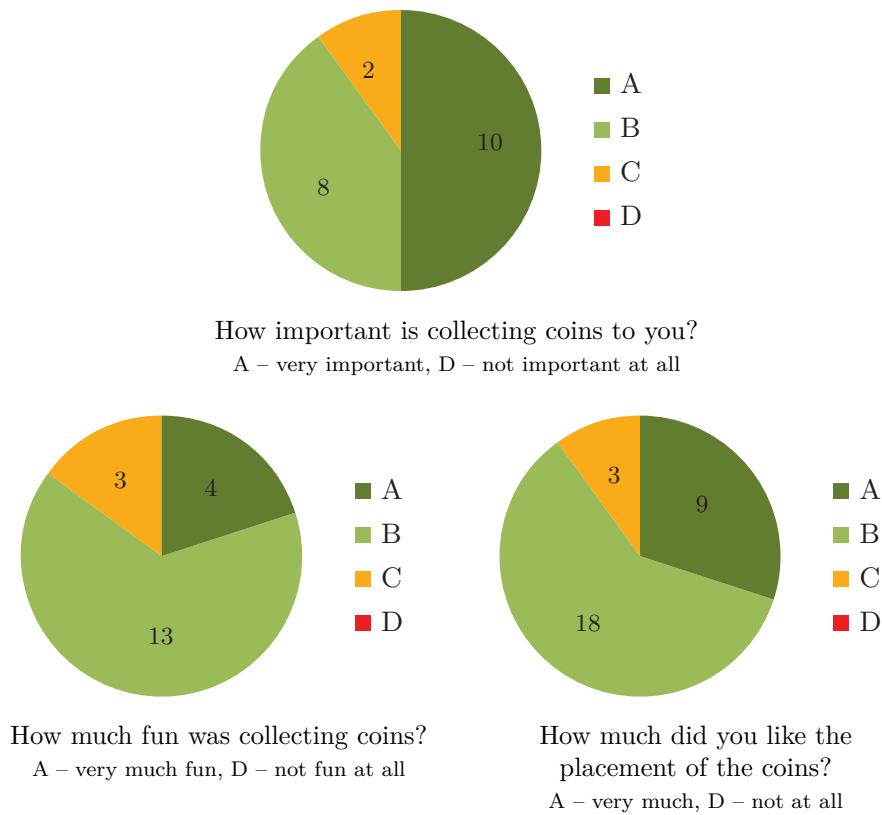
**Figure 8.10:** Not everybody liked the caves, one person even did not like them at all.

the limited control and did not like the level. But she also did not really like the other levels as well.

Interestingly, the profile of the player who did not like level 2 at all holds some similarities to Emily's. From now on this person will be referred to as Anakin. He plays digital games daily but had never played *Elements*. He was also able to finish all three levels. As Emily, he did only play the levels until he finished them successfully once and then went on to the next one. The first level was much too easy for him (he was the only person who claimed this), the second level was a little bit difficult and the third level was just right. He was also the only one who did not like the dark caves at all<sup>16</sup> (see figure 8.10). As Emily, Anakin is of the opinion that the levels become boring after playing them for some time. He answered that he can rather not discover something new in every new level and that he would like more variety. He was content with the controls; the only bad thing he thinks about them is that they are rather complicated. Anakin answered that he does not entirely understand what he has to do and why he dies. He also complained about this in the question about what he would change regarding the second level. He found it not logical that the fire is eaten by the plant and he did not expect the stone to fall through the stone walls. Moreover, he noticed that the waterfalls should stand out more and that checkpoints could improve the game. When he was asked what caused the biggest trouble to him in level 3, after he died for several times at the same position, he answered that this was the changing between the elements.

This suggests that the difficulty was not their major problem but that they did not like the levels because they found them rather boring and were not satisfied with the overall game concept such as the controls and the obstacles.

<sup>16</sup>Emily did rather not like the dark caves.



**Figure 8.11:** These statistics are created using the answers of all 30 times the coin questions were asked.

### 8.1.5 Coins

People who collected more than 90% of all coins in a level were asked several coin-specific questions (see section 7.1.4). In total, these questions were asked 30 times for 20 different participants (see figure 8.11). For 18 of them, collecting was important. No one said that it was no fun at all and also the number of coins was appropriate: two third of them stated that the number of coins was just right and no one claimed that there were far too many or far too few coins.

In the general feedback question (see section 8.1.10), some participants also claimed that they wanted to collect all coins but that this was difficult for them.

It would have been interesting to know what the other participants thought about collecting coins, if it was important to them or if they did not care at all, as maybe some of them also tried to collect all coins but were just not able to collect the high amount of 90% of all coins in a level.



### 8.1.6 Controls

Several asked questions referred to the controls of the game (see figure 8.12). The statistics generated out of this data indicate that only about 75% of the participants did not have any serious problems with controlling the character.

This issue becomes even more serious when having a look at the comments concerning why people skipped levels (see section 8.1.2) and their general feedback (see section 8.1.10). Frequently it was mentioned that they had problems with switching between the different elements. Figure 8.13 summarizes the significance of this problem.

### 8.1.7 Understanding

This section aims to discover in how far the players understood the game. In this context, the different questions referring to this topic are discussed.

When people died often<sup>17</sup> because of the same obstacle, some questions were asked to discover if they understood why they had died. These questions were asked 15 times for the obstacle waterfall, four times for the plant and only once for the stone crusher. This relations may be caused by the *predominance of fire* which is discussed in detail in section 8.1.9. Everyone who was asked these questions because of frequently dying by reason of a waterfall or a stone crusher answered that they understood why they died, that they found it logical and that they were not confused because of this. The four people who were asked these questions when being eaten by a killer plant claimed that they understood why they died but only two of them found it completely logical<sup>18</sup>. This indicates that most people understood after five times dying because of the same obstacle what they had done wrong. But the idea would have been that they understood it right away. Unfortunately, this was not covered by a question. But a look at the open text questions concerning level 2 revealed that they would have liked descriptions at the first occurrence of the obstacles to understand them.

At the end of the test session, the users were asked if they understood what to do and why they died (see figure 8.14). The results show that all but one understood why they died and that no one ever was completely confused about what to do. But as these questions were asked at the end of the session this just means that after some trial and error they found it out on their own and that it is clear at the end.

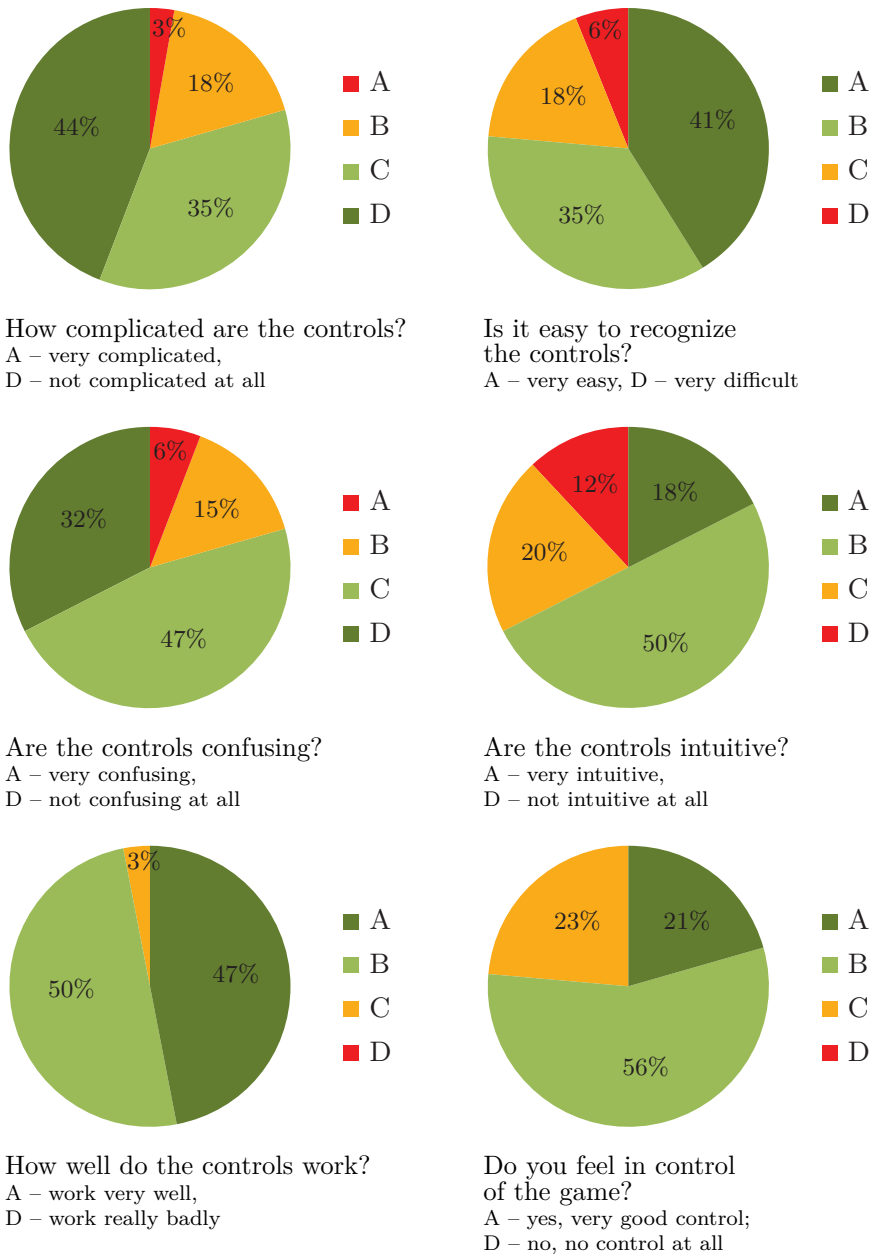
Out of the 12 players who died frequently at a certain position (see section 7.1.4), only one checked the checkbox which said *I don't know how this is intended to work/ am confused*.<sup>19</sup>

---

<sup>17</sup>What *often* means in this context is explained in section 7.1.4.

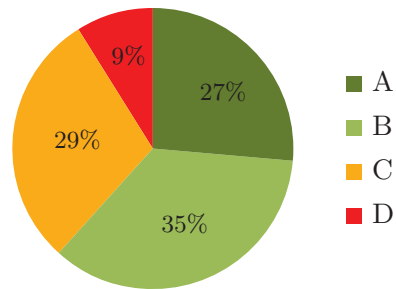
<sup>18</sup>One person said that it was *rather logical* and one that it was *rather not logical*.

<sup>19</sup>The original German question said *Ich verstehe nicht wie das gehen soll/ bin verwirrt*.



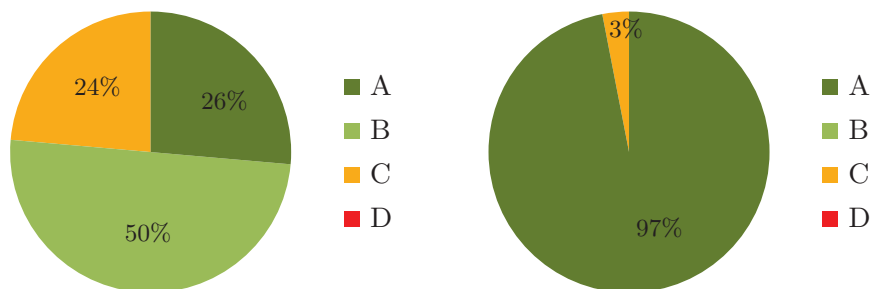
**Figure 8.12:** These statistics offer valuable information about how the participants experienced the controls.

In summary, it seems that at the beginning it was not clear to the participants how to survive the different obstacles, but at the end they understood what to do. In contrast to the waterfall and the stone crusher, the abilities of the plant were felt as being illogical by several players.



How well do you cope with the changing between the two elements?  
A – very well, D – very badly

**Figure 8.13:** Switching between the different elements seemed to be a difficult task for nearly 40% of the players.



I always understand what I have to do.  
A – yes, D – no

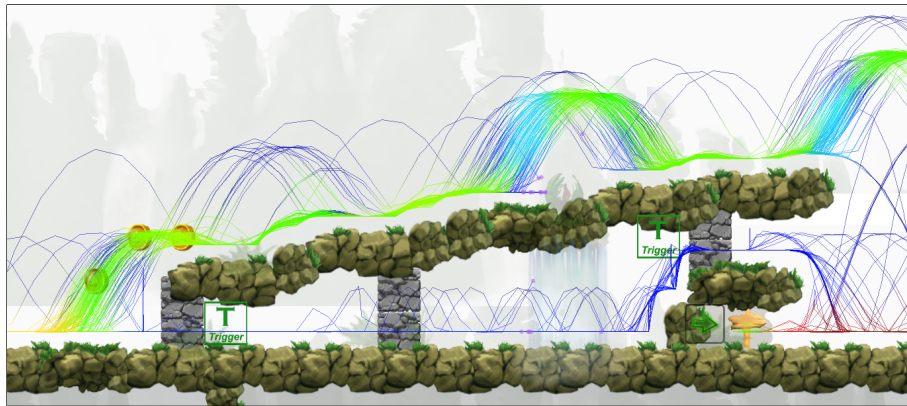
When I die, I understand why.  
A – yes, D – no

**Figure 8.14:** These statistics offer valuable clues about if the participants understood the game.

### 8.1.8 Heatmaps and Lines

Using the `HeatmapManager` (see section 6.2.2) and the tracked metrics data, two types of heatmaps were generated: position heatmaps and game-over position heatmaps.

Unfortunately, with the flow of the level the colors of the heatmaps became more and more bluish. This was because the color calculation was done absolutely using the typical heatmap color code: The smallest value was mapped to blue and the highest value was mapped to red. As all levelplays start at the opening of the level and then end sooner or later on the path to the goal, necessarily more features got tracked at the beginning of the level and fewer at the end, resulting in the described color progression.



**Figure 8.15:** This heatmap shows a part of the first level with two possible paths. Thanks to the heatmap color code it is clearly visible that most players chose to use the upper path.

### Position Heatmaps

The position heatmap was generated using the tracked position features of all participants. It was decided to only draw the lines to achieve clearer results<sup>20</sup>.

The calculated levelplay lines offered valuable information about where most people went and which parts of the level were not visited frequently. It was possible to discover that no participant had found the secret place which was hidden in a level<sup>21</sup>. Furthermore, they illustrated for forks which way was used by the majority of the players (see figure 8.15) and enabled valuable information about where the character got stuck (see figure 8.16).

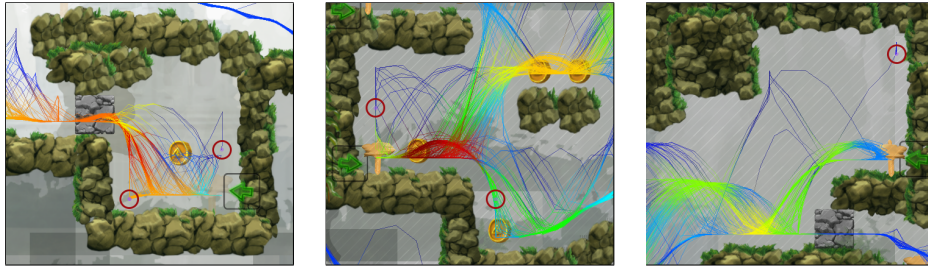
Moreover, there were several positions where players could fall down to a previously visited area in the level. The levelplay lines visualized in which areas this happened.

### Game-over Position Heatmaps

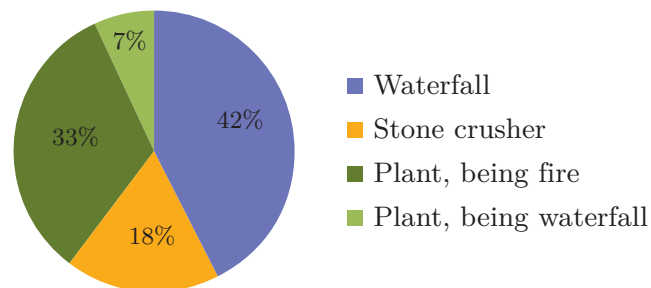
In addition to the position heatmaps, game-over position heatmaps were also generated using the positions of the obstacles at which players died. Thus, it was possible to discover the positions where most players died. Unfortunately, most positions were bluish, which is caused by the problem mentioned above but further also indicates that except for a few positions, players died all over the map approximately uniformly distributed. But this hypothesis would have to be verified by a more detailed analysis. Figure 8.7 shows a detail of the game-over position heatmap of level 3.

<sup>20</sup>An example image of a heatmap in which both, lines and rectangles are drawn can be found in chapter 6 in figure 6.7.

<sup>21</sup>In the preliminary user study, one participant discovered this area.



**Figure 8.16:** These heatmap details represent positions in the first level where participants got caught. The red circles mark the points at which players got stuck.



**Figure 8.17:** This diagram shows the game-reasons. Most often a game-over was caused by a plant or a waterfall. In 75% of the death events the element was fire.

### 8.1.9 Predominance of Fire

For every position feature it was tracked if the character was either fire or stone at this particular moment. Using this information it was possible to calculate the percentage of how often fire was used (about 74% of the time) and how often stone was used (about 26% of the time)<sup>22</sup>. Correlating, players also died more often while being fire (see figure 8.17).

It can only be suggested why people chose to use the fire that often. One reason may be the level elements: Waterfalls require the use of the stone as well as stone walls<sup>23</sup>. To survive stone crushers, the element fire was needed. Plants killed both characters but jumping over them was maybe easier using the fire as this element can jump much higher than the stone. Furthermore, high or wide jumps and the torch which had to be enflamed at the end of

<sup>22</sup>It should be noted that this values refer to the gameplay duration and not to the covered distance which may be different because of the higher running speed of the fire and because it was not taken into account if the character was currently moving or standing.

<sup>23</sup>At some positions (e.g. in level 1) it was also possible to avoid stone walls by jumping over them. Therefore it was not necessary to use the stone there.

the level also required the use of fire.

By counting the number of different obstacles, it was found out that their number was approximately equal (summing up all three levels). But the explanatory power of this is low as there were alternative ways which made it possible to avoid some hazards. Moreover, a detailed analysis of the balance of the level would also have to consider the position of the obstacles (if they are at the beginning or at the end of the level) and their surroundings (for example if there are three waterfalls behind one another).

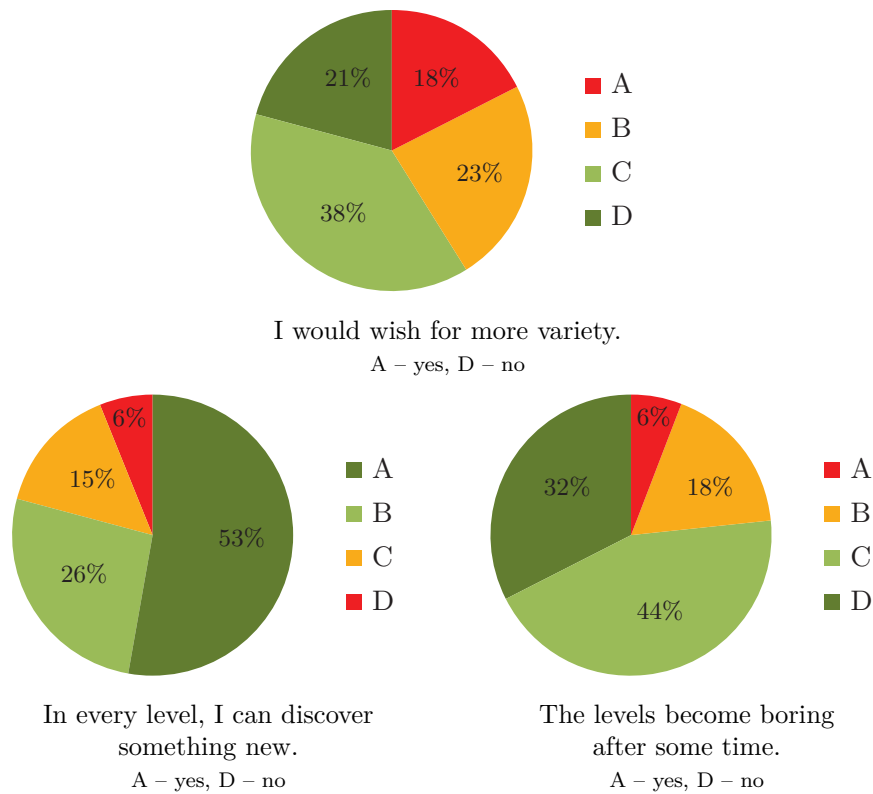
Without a proper analysis it can only be noted that the fire was used most of the time but it cannot be identified why.

### 8.1.10 Feedback

At the end of each level, the participants were asked if there was anything they would change in the level. The answers were mainly not level specific but concerned general issues. Therefore the following list combines these results with the answers gained from the question which appeared at the end of the test session which was dedicated to feedback of any kind.

- The desire for *checkpoints* was mentioned frequently.
- Some complained about not having the ability to *turn around* by themselves or having not enough turn-around possibilities. Especially linked with the attempt of collecting all coins this caused difficulties.
- Coin collectors also complained about too many different ways which would make it *difficult to collect all coins*. Some did not see the possibility of collecting all, although it would have been possible.
- Some mentioned that there were several positions where the character *got stuck*.
- It was noted that at their first appearance a *tutorial or explanations* of the obstacles would have been helpful.
- They criticized that it was not logical that the *plant* ate the fire instead of getting burned.
- Having only *one button for changing the element* would have been appreciated.
- Some users mentioned that the *level design* of the third level was unclear.
- Several participants noted that *sound* could improve the game.

Furthermore, some general questions also allowed valuable insight into the wishes of the players (see figure 8.18). Nearly half of them would have liked more variety and nearly a quarter thought that the levels became boring after some time. But it has to be taken into account that the participants played only three levels and some of them tried them exceedingly often.



**Figure 8.18:** These diagrams suggest that the levels may not stay permanently interesting.

### 8.1.11 Summary

Summing up the already described data, it can be suggested that one major challenge was to finish the levels without dying even once. Checkpoints could solve this problem. They could decrease the number of times the players have to try the levels until they finish them successfully. This could lead to the effect that players do not think anymore that the levels become boring after some time. Furthermore, the control system is still not perfect yet and several participants had problems with changing the elements. There are several possibilities which could be taken into account to enhance this: For example, using only one button for switching between the elements and adding a button for changing the direction. This could also help people who wanted to collect all coins. A different idea would be to add more turn-around positions. It seems as if the hazards require proper introductions to instantly make it clear to anybody how to survive them. Moreover, it was possible to exactly identify positions where the character got stuck, which provides the chance to fix these issues.

This information could be used to create an improved version of the game and then test it again to find out if the changes really improved the game and if other bugs appear maybe caused by these changes. Such an iterative process is also suggested by [71, pp. 11–13][27].

During the analysis, several more ideas for the tool and further evaluations emerged. These are presented in the next section.

## 8.2 Further Ideas

This section describes some further development possibilities for the evaluation system. These are grouped into two parts: ideas which would affect the pre-evaluation analysis and concepts which refer to the tracking process.

### 8.2.1 Analysis

The enormous amount of tracked data would also allow further analysis possibilities. It would, for example, be possible to create position heatmaps for the element fire and for the element stone separately. This would figuratively illustrate where the players used which element. The main user study discovered that most often the element of fire was used (see section 8.1.9). This would suggest that there may be areas in the levels in which no person ever used the stone. Knowing these locations could simplify the balancing process which could aim to make the elements approximately equally important.

Another idea would be to calculate the *relative mortality rate* for each obstacle in every level separately as the number of times a player died because of this hazard divided by the number of times a player tried to pass this position (either dying or safely). In contrast to the game-over heatmaps (see section 8.1.8), this approach could also identify difficult areas at the end of a level.

### 8.2.2 Tracking

In the performed user study, it was not tracked when a person pressed a certain key to control the character. Only the outcome, such as the character changing its element, was recorded. If this information would have been tracked it would have been possible to determine at which positions players tried to jump. Visualizing this information could reveal where players usually tried to jump and if, for example, they fell down from a platform because they pressed the space button too late.

Furthermore, adding video capturing could be used to gain more information about the feelings of the users. Mapping the facial expressions or other player reaction, such as swearing, to certain areas in the levels could provide valuable feedback concerning how much they liked or disliked these particular parts.



### 8.3 Summary

A couple of very different kinds of information were tracked. Metrics data allowed insights into what had happened during the gameplay, such as where players went or the positions where they had died. It was possible to exactly specify areas in which the character could get stuck, which represent major problems of the level design as the entire level has to be restarted in this case. Furthermore, quantitative questions offered valuable information about how people felt or what they thought under certain circumstances. They addressed particular situations and areas to gain deeper insight into how they were experienced by the users. Other questions focused on general game elements such as the controls or dealt with demographics. This broad range of information and its ability to be combined enables detailed analysis which can be done in many different ways such as, for example, using player analysis (see section 8.1.4) or heatmaps (see section 8.1.8) to address various topics.

Problems concerning the game itself, its controls and the level design such as, for instance, stuck points were found. Based on this data it was possible to propose some suggestions for improving the game (see section 8.1.11). Using the evaluation tool, it was easily possible to reach players and to get a lot of data without having to invite them. Most of the data was received comfortably via email. A disadvantage was the amount of time which was required for the integration of the tool into the game. The necessary effort therefore is highly dependent on the particular game, its architecture and ability to be extended, for example, by a game state for questioning. But once it is included, it can be used for any further tests during development and the number of participants does not influence the amount of time required for the study. The only thing which then still has to be taken into account is the time needed for the analysis of the data, which is highly dependent on the data, the intended results and the preciseness of the analysis.

With respect to all these facts, it is argued that the tool was able to match the defined requirements.

## Chapter 9

# Conclusion

The goal of this thesis was to find problems in the level design of platform games. Therefore, a metrics tracking system was developed and integrated into the self-made platform game *Elements*. It was extended with the possibility to ask questions which could be triggered by an in-game evaluation of the metrics data. The user study revealed that this system was able to automatically collect a significant amount of different kinds of data which allowed insights into the happenings and the feelings and thoughts of the players during gameplay. This way, it was possible to find problems concerning the level design and the overall game concept.

A very difficult task was to find adequate questions and to decide when to ask them. It was found out that the wording of these questions was of particular importance. Therefore the preliminary user study formed a fundamental part of this work and was essential for the quality assurance of the collected data. This indicates that an iterative development cycle may not only be useful for games and their quality assurance, but also for their evaluation methods and instruments.

In addition to the development of the system itself, the integration into the game proved to be very time consuming. Furthermore, also the required amount of time for the creation of appropriate analysis software should not be underestimated when the development of such a program is targeted.

The created tool demanded from the participants to send the tracked data via mail to the test supervisor. A more advanced system could automatically send the data via an internet connection to a dedicated server which would probably be more comfortable for both parties.

The objective of this thesis was merely to find *problems* in the game, but for developers other aspects, such as what the players highly appreciated, might be interesting as well. On the basis of the performed user study it can be supposed that the developed system would also be able to serve other purposes. It could be the mission of another study to test its full potential.

# Appendix A

## Content of the CD-ROM

**Format:** CD-ROM, Single Layer, ISO9660-Format

### A.1 PDF-File

**Pfad:** /

[Bugl\\_Angelika\\_2014.pdf](#) Master's thesis

### A.2 Study Results

All information in the *study results* folder refers to the main user study. The MAC address of the participants was replaced by 'mac-address-' followed by an increasing number wherever it appeared in the tracked data or in any statistic file.

**Pfad:** /study results

- [gameover heatmaps](#) . . . Game-over heatmaps; there is no game-over heatmap for the first level, as it was not possible to die in this level.
- [position heatmaps](#) . . . Position heatmaps
- [questions](#) . . . . . This folder contains the German question list of all original questions which were asked at the final user study at the *University of Applied Sciences Hagenberg* and the asked questions of the mail and the fh evaluation in the `model` file format.
- [raw data](#) . . . . . Tracked metrics and question data
- [statistics](#) . . . . . Statistics as `model` files

### A.3 Miscellaneous

**Pfad:** /

- [application](#) . . . . . The *Elements* evaluation prototype which was used for the mail evaluation.
- [images](#) . . . . . Various used images
- [literature](#) . . . . . Copies of the online sources

# References

## Literature

- [1] Erik Andersen et al. “Gameplay Analysis through State Projection”. In: *Proceedings of the Fifth International Conference on the Foundations of Digital Games*. (Monterey, CA, USA). FDG 2010. New York, NY, USA: ACM, June 2010, pp. 1–8 (cit. on pp. 11, 17).
- [2] Salley Barnes. *Questionnaire Design and Construction*. Tech. rep. Institute for Learning and Research Technology ILRT, Oct. 2001, pp. 1–4 (cit. on pp. 22, 67).
- [3] Richard Bartle. “Virtual Worlds: Why People Play”. In: *Massively Multiplayer Game Development 2* (2005) (cit. on pp. 12, 14, 15).
- [4] Harvey Russell Bernard. *Research Methods in Anthropology: Qualitative and Quantitative Approaches*. 4th ed. AltaMira Press, 2006 (cit. on p. 20).
- [5] Regina Bernhaupt. “User Experience Evaluation in Entertainment and Games”. In: *Human-Computer Interaction–INTERACT 2011*. (Lisbon, Portugal). Ed. by Pedro Campos et al. Vol. 6949. Lecture Notes in Computer Science. Berlin Heidelberg, Germany: Springer, Sept. 2011, pp. 716–717 (cit. on p. 11).
- [6] Regina Bernhaupt, Manfred Eckschlager, and Manfred Tscheligi. “Methods for Evaluating Games: How to Measure Usability and User Experience in Games?” In: *Proceedings of the International Conference on Advances in Computer Entertainment Technology*. (Salzburg, Austria). ACE 2007. New York, NY, USA: ACM, June 2007, pp. 309–310 (cit. on pp. 11, 12).
- [7] Frank Biocca, Chad Harms, and Jenn Gregg. “The Networked Minds Measure of Social Presence: Pilot Test of the Factor Structure and Concurrent Validity”. In: *Presence 2001 4th Annual International Workshop on Presence, Philadelphia, PA*. (Philadelphia, Pennsylvania, USA). PRESENCE 2001. May 2001 (cit. on p. 23).

- [8] Norman Bradburn, Seymour Sudman, and Brian Wansink. *Asking Questions: The Definitive Guide to Questionnaire Design – For Market Research, Political Polls, and Social and Health Questionnaires*. Jossey-Bass, 2004 (cit. on pp. 66, 67).
- [9] Jeanne H. Brockmyer et al. “The Development of the Game Engagement Questionnaire: A Measure of Engagement in Video Game-Playing”. In: *Journal of Experimental Social Psychology* 45.4 (July 2009), pp. 624–634 (cit. on p. 23).
- [10] Emily Brown and Paul Cairns. “A Grounded Investigation of Game Immersion”. In: *CHI 2004 Extended Abstracts on Human Factors in Computing Systems*. (Vienna, Austria). CHI EA 2004. New York, NY, USA: ACM, 2004, pp. 1297–1300 (cit. on pp. 12, 14, 15).
- [11] Alessandro Canossa. “Meaning in Gameplay: Filtering Variables, Defining Metrics, Extracting Features and Creating Models for Gameplay Analysis”. In: *Game Analytics: Maximizing the Value of Player Data*. Ed. by Magy Seif El-Nasr, Anders Drachen, and Alessandro Canossa. London: Springer-Verlag, 2013. Chap. 13, pp. 255–283 (cit. on pp. 26, 29).
- [12] Alessandro Canossa, Anders Drachen, and Janus Rau Møller Sørensen. “Arrrgghh!!! – Blending Quantitative and Qualitative Methods to Detect Player Frustration”. In: *Proceedings of the 6th International Conference on Foundations of Digital Games*. (Bordeaux, France). FDG 2011. New York, NY, USA: ACM, 2011, pp. 61–68 (cit. on p. 41).
- [13] David A. Clearwater. “What Defines Video Game Genre? Thinking about Genre Study after the Great Divide”. In: *Loading... The Journal of the Canadian Game Studies Association* 5.8 (2011), pp. 29–49 (cit. on p. 17).
- [14] Kate Compton and Michael Materas. “Procedural Level Design for Platform Games”. In: *Proceedings of the Second Artificial Intelligence and Interactive Digital Entertainment Conference*. (Marina del Rey, California, USA). Ed. by John Laird and Jonathan Schaeffer. AIIDE 2006. The AAAI Press, June 2006, pp. 109–111 (cit. on pp. 8, 9).
- [15] Ben Cowley et al. “Toward an Understanding of Flow in Video Games”. In: *Computer Entertainment* 6.2 (July 2008), 20:1–20:27 (cit. on p. 13).
- [16] Mihaly Csikszentmihalyi. “A Theoretical Model for Enjoyment”. In: *Beyond Boredom and Anxiety*. Jossey-Bass, 1975 (cit. on p. 13).
- [17] Mihaly Csikszentmihalyi. *Flow: The Classic Work on how to Achieve Happiness*. Rider, 2002 (cit. on p. 13).

- [18] Heather Desurvire, Martin Caplan, and Jozsef A. Toth. “Using Heuristics to Evaluate the Playability of Games”. In: *CHI 2004 Extended Abstracts on Human Factors in Computing Systems*. (Vienna, Austria). CHI EA 2004. New York, NY, USA: ACM, Apr. 2004, pp. 1509–1512 (cit. on pp. 16, 22–24, 54, 58).
- [19] Yellowlees Douglas and Andrew Hargadon. “The Pleasure Principle: Immersion, Engagement, Flow”. In: *Proceedings of the Eleventh ACM on Hypertext and Hypermedia*. (San Antonio, Texas, USA). Hypertext 2000. New York, NY, USA: ACM, 2000, pp. 153–160 (cit. on pp. 11, 12).
- [20] Anders Drachen and Alessandro Canossa. “Towards Gameplay Analysis via Gameplay Metrics”. In: *Proceedings of the 13th International MindTrek Conference: Everyday Life in the Ubiquitous Era*. (Tampere, Finland). MindTrek 2009. New York, NY, USA: ACM, 2009, pp. 202–209 (cit. on pp. 10, 22, 24, 25, 27, 28, 30, 31, 36, 38, 39).
- [21] Anders Drachen, Alessandro Canossa, and Janus Rau Møller Sørensen. “Gameplay Metrics in Game User Research: Examples from the Trenches”. In: *Game Analytics: Maximizing the Value of Player Data*. Ed. by Magy Seif El-Nasr, Anders Drachen, and Alessandro Canossa. London: Springer-Verlag, 2013. Chap. 14, pp. 285–319 (cit. on pp. 34, 35, 38–41).
- [22] Anders Drachen, Alessandro Canossa, and Georgios N. Yannakakis. “Player Modeling Using Self-Organization in Tomb Raider: Underworld”. In: *CIG2009 2009 IEEE Symposium on Computational Intelligence and Games*. (Milano, Italy). CIG 2009. IEEE, Sept. 2009, pp. 1–8 (cit. on pp. 24, 36, 37).
- [23] Anders Drachen, André Gagné, and Magy Seif El-Nasr. “Sampling for Game User Research”. In: *Game Analytics: Maximizing the Value of Player Data*. Ed. by Magy Seif El-Nasr, Anders Drachen, and Alessandro Canossa. London: Springer-Verlag, 2013. Chap. 9, pp. 143–167 (cit. on p. 29).
- [24] Anders Drachen, Magy Seif El-Nasr, and Alessandro Canossa. “Game Analytics – The Basics”. In: *Game Analytics: Maximizing the Value of Player Data*. Ed. by Magy Seif El-Nasr, Anders Drachen, and Alessandro Canossa. London: Springer-Verlag, 2013. Chap. 2, pp. 13–40 (cit. on pp. 25–30).
- [25] Laura Ermi and Frans Mäyrä. “Fundamental Components of the Gameplay Experience: Analysing Immersion”. In: *Changing Views: Worlds in Play*. (Vancouver, Canada). DiGRA 2005. Vancouver, Canada: University of Vancouver, June 2005, pp. 15–27 (cit. on pp. 11–14).

- [26] Melissa A. Federoff. “Heuristics and Usability Guidelines for the Creation and Evaluation of Fun in Video Games”. MA thesis. Indiana University, Bloomington, 2002 (cit. on pp. 16, 24, 54, 58).
- [27] Remigius Fierley and Stephan Engl. “User Experience Methoden und Games: Erkenntnisse aus der Praxis”. German. In: *Interaktive Kulturen: Workshop-Band: Proceedings der Workshops der Mensch & Computer 2010 - 10. Fachübergreifende Konferenz für Interaktive und Kooperative Medien*. (Duisburg, Germany). Ed. by Ulrik Schroeder. DeLFI 2010. Berlin, Germany: Logos Verlag, Sept. 2010 (cit. on pp. 10, 19–22, 95).
- [28] Alex Galuzin. *Ultimate Level Design Guide*. 2011. URL: [www . WorldOfLevelDesign.com](http://www.WorldOfLevelDesign.com) (cit. on p. 18).
- [29] Pietro Guardini and Paolo Maninetti. “Better Game Experience Through Game Metrics: A Rally Videogame Case Study”. In: *Game Analytics: Maximizing the Value of Player Data*. Ed. by Magy Seif El-Nasr, Anders Drachen, and Alessandro Canossa. London: Springer-Verlag, 2013. Chap. 16, pp. 325–361 (cit. on p. 41).
- [30] Eric Hazan. “Contextualizing Data”. In: *Game Analytics: Maximizing the Value of Player Data*. Ed. by Magy Seif El-Nasr, Anders Drachen, and Alessandro Canossa. London: Springer-Verlag, 2013. Chap. 21, pp. 477–496 (cit. on pp. 30, 41).
- [31] Andreas Holzinger. “Usability Engineering Methods for Software Developers”. In: *Communications of the ACM* 48.1 (Jan. 2005), pp. 71–74 (cit. on pp. 21, 22, 24).
- [32] Kenneth Hullett et al. “Data Analytics for Game Development (NIER Track)”. In: *Proceedings of the 33rd International Conference on Software Engineering*. (Honolulu, HI, USA). ICSE 2011. New York, NY, USA: ACM, May 2011, pp. 940–943 (cit. on p. 41).
- [33] Kenneth Hullett et al. “Empirical Analysis of User Data in Game Software Development”. In: *Proceedings of the ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*. (Lund, Sweden). ESEM 2012. New York, NY, USA: ACM, Sept. 2012, pp. 89–98 (cit. on pp. 10, 19, 24, 30, 41).
- [34] Wijnand IJsselsteijn, Yvonne de Kort, and Karolien Poels. “Game Experience Questionnaire”. URL: <http://www.gamexplab.nl/index.php?page=contact> (cit. on p. 23).
- [35] Wijnand IJsselsteijn et al. “Characterising and Measuring User Experiences in Digital Games”. In: *Proceedings of the International Conference on Advances in Computer Entertainment Technology*. (Salzburg, Austria). Vol. 2. ACE 2007. New York, NY, USA: ACM, June 2007 (cit. on pp. 11–14, 21).



- [36] Wijnand IJsselsteijn et al. “Measuring the Experience of Digital Game Enjoyment”. In: *Proceedings of Measuring Behavior 2008: 6th International Conference on Methods and Techniques in Behavioral Research*. (Maastricht, The Netherlands). Vol. 6. Noldus Information Technology, Aug. 2008, pp. 88–89 (cit. on p. 23).
- [37] Jun H. Kim et al. “Tracking Real-Time User Experience (TRUE): A Comprehensive Instrumentation Solution for Complex Systems”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. (Florence, Italy). CHI 2008. New York, NY, USA: ACM, Apr. 2008, pp. 443–452 (cit. on pp. 24, 25, 29, 32–35).
- [38] Matias J. Kivikangas. “Psychophysiology of Flow Experience: An Explorative Study”. MA thesis. Department of Psychology, University of Helsinki, Feb. 2006 (cit. on p. 13).
- [39] Christina Koeffel et al. “Using Heuristics to Evaluate the Overall User Experience of Video Games and Advanced Interaction Games”. In: *Evaluating User Experience in Games*. Ed. by Regina Bernhaupt. Human-Computer Interaction Series 2010. London: Springer-Verlag, 2010. Chap. 13, pp. 233–256 (cit. on pp. 16, 54, 58).
- [40] Elina M.I. Koivisto and Hannu Korhonen. *Mobile Game Playability Heuristics*. Tech. rep. Nokia Corporation, 2006 (cit. on p. 16).
- [41] Hannu Korhonen. “The Explanatory Power of Playability Heuristics”. In: *Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology*. (Lisbon, Portugal). ACE 2011. New York, NY, USA: ACM, 2011, 40:1–40:8 (cit. on p. 24).
- [42] Hannu Korhonen and Elina M. I. Koivisto. “Playability Heuristics for Mobile Games”. In: *Proceedings of the 8th Conference on Human-computer Interaction with Mobile Devices and Services*. (Helsinki, Finland). MobileHCI 2006. New York, NY, USA: ACM, Sept. 2006, pp. 9–16 (cit. on pp. 16, 24).
- [43] Yvonne A. W. de Kort, Wijnand A. IJsselsteijn, and Karolien Poels. “Digital Games as Social Presence Technology: Development of the Social Presence in Gaming Questionnaire (SPGQ)”. In: *Proceedings of PRESENCE 2007: The 10th International Workshop on Presence*. (Barcelona, Spain). PRESENCE 2007. Oct. 2007 (cit. on p. 23).
- [44] René Ksuz. “Discount-Methoden zur Evaluierung der Gaming Experience”. German. Hagenberg, Austria: University of Applied Sciences Upper Austria, School for Informatics, Communications and Media, Hagenberg Campus, Media Technology and Design, Mar. 2011 (cit. on p. 22).

- [45] Sauli Laitinen. “Usability and Playability Expert Evaluation”. In: *Game Usability: Advancing the Player Experience*. Ed. by Katherine Isbister and Noah Schaffer. Morgan Kaufmann, 2008. Chap. 7, pp. 91–111 (cit. on pp. 10, 11, 16).
- [46] Luis Levy and Jeannie Novak. *Game Development Essentials: Game QA & Testing*. Clifton Park, NY, USA: Delmar, Cengage Learning, 2010 (cit. on pp. 10, 11).
- [47] Petra Lietz. “Research into Questionnaire Design – A Summary of the Literature”. In: *International Journal of Market Research* 52.2 (2010), pp. 249–273 (cit. on p. 22).
- [48] Jordan Lynn. “Combining Back-End Telemetry Data with Established User Testing Protocols: A Love Story”. In: *Game Analytics: Maximizing the Value of Player Data*. Ed. by Magy Seif El-Nasr, Anders Drachen, and Alessandro Canossa. London: Springer-Verlag, 2013. Chap. 22, pp. 497–514 (cit. on p. 39).
- [49] Tobias Mahlmann et al. “Predicting Player Behavior in Tomb Raider: Underworld”. In: *2010 IEEE Symposium on Computational Intelligence and Games (CIG)*. (Dublin). IEEE 2010. IEEE, Aug. 2010, pp. 178–185 (cit. on pp. 36, 37, 41).
- [50] Raphaël Marczak et al. “Feedback-based Gameplay Metrics: Measuring Player Experience via Automatic Visual Analysis”. In: *Proceedings of The 8th Australasian Conference on Interactive Entertainment: Playing the System*. (Auckland, New Zealand). IE 2012. ACM, 2012, 6:1–6:10 (cit. on p. 11).
- [51] Sonja Marjanovic, Stephen Hanney, and Steven Wooding. *A Historical Reflection on Research Evaluation Studies, their Recurrent Themes and Challenges*. Tech. rep. RAND Corporation, 2009 (cit. on p. 20).
- [52] Alison McMahan. “Immersion, Engagement and Presence: A Method for Analyzing 3-D Video Games”. In: *The Video Game Theory Reader*. Ed. by Mark J. P. Wolf and Bernard Perron. New York, NY, USA: Routledge, 2003. Chap. 3, pp. 67–86 (cit. on pp. 12, 14).
- [53] Chris Melissinos and Patrick O’Rourke. *The Art of Video Games: From Pac-Man to Mass Effect*. New York, NY, USA: Welcome Books, 2012 (cit. on p. 5).
- [54] Janet Horowitz Murray. *Hamlet on the Holodeck: The Future of Narrative in Cyberspace*. New York, NY, USA: The Free Press, 1997 (cit. on p. 14).

- [55] E. Lennart Nacke et al. “Bringing Digital Games to User Research and User Experience”. In: *EI-2010 Entertainment Interfaces 2010 : Proceedings of the Entertainment Interfaces Track 2010 at Interaktive Kulturen 2010*. (Duisburg, Germany). Ed. by Jörg Niesenhaus, Matthias Rauterberg, and Maic Masuch. Vol. 634. Ceur Workshop Proceedings. Ceur, Sept. 2010 (cit. on pp. 10, 19, 20, 30, 31).
- [56] Lennart Nacke. “From Playability to a Hierarchical Game Usability Model”. In: *Proceedings of the 2009 Conference on Future Play on @ GDC Canada*. (Vancouver, British Columbia, Canada). Future Play 2009. New York, NY, USA: ACM, 2009, pp. 11–12 (cit. on p. 12).
- [57] Lennart Nacke and Anders Drachen. “Towards a Framework of Player Experience Research”. In: *2nd International Workshop on Evaluating Player Experience in Games (EPEX 2011): At 6th International Conference on the Foundations of Digital Games (FDG)*. (Bordeaux, France). Vol. 11. EPEX 2011. ACM, 2011 (cit. on pp. 19, 20).
- [58] Lennart Nacke, Anders Drachen, and Stefan Göbel. “Methods for Evaluating Gameplay Experience in a Serious Gaming Context”. In: *International Journal of Computer Science in Sport* 9.2 (2010) (cit. on pp. 10, 11).
- [59] Lennart E. Nacke et al. “Playability and Player Experience Research”. In: *Breaking New Ground: Innovation in Games, Play, Practice and Theory*. (Brunel, West London, UK). Ed. by Barry Atkins and Helen Kennedy. DiGRA 2009. London, UK, Sept. 2009 (cit. on pp. 10–12, 24, 25, 31).
- [60] Lennart Nacke and Craig Lindley. “Boredom, Immersion, Flow – A Pilot Study Investigating Player Experience”. In: *Proceedings of the IADIS Gaming 2008: Design for Engaging Experience and Social Interaction*. (Amsterdam, The Netherlands). Ed. by Eleonore T. Thij. IADIS Gaming 2008. IADIS. IADIS Press, July 2008, pp. 103–107 (cit. on pp. 10, 11).
- [61] Lennart Nacke and Craig A. Lindley. “Flow and Immersion in First-Person Shooters: Measuring the Player’s Gameplay Experience”. In: *Proceedings of the 2008 Conference on Future Play: Research, Play, Share*. (Toronto, Ontario, Canada). Future Play 2008. New York, NY, USA: ACM, 2008, pp. 81–88 (cit. on pp. 11, 12).
- [62] Nakamura, Jeanne and Csikszentmihalyi, Mihaly. “The Concept of Flow”. In: *Handbook of Positive Psychology*. Ed. by C. R. Snyder and Shane J. Lopez. New York, NY, USA: Oxford University Press, 2002. Chap. 7, pp. 89–105 (cit. on pp. 13, 14).

- [63] Jakob Nielsen. “Heuristic Evaluation”. In: *Usability Inspection Methods*. Ed. by Jakob Nielsen and Robert L. Mack. John Wiley & Sons, 1994. Chap. 2, pp. 25–62 (cit. on p. 23).
- [64] Jeannie Novak. *Game Development Essentials*. 3rd ed. Clifton Park, NY, USA: Delmar, Cengage Learning, 2012. Chap. 1 (cit. on p. 5).
- [65] Luís Lucas Pereira and Lincínio Roque. “Defining Gameplay Metrics from a Participation-centered Perspective”. In: *Fun and Games 2012 Workshop on Player Experience in Videogames, Toulouse, France, September 2012*. (Toulouse, France). FNG 2012. Sept. 2012 (cit. on pp. 10–12, 24, 25, 29).
- [66] David Pinelle, Nelson Wong, and Tadeusz Stach. “Heuristic Evaluation for Games: Usability Principles for Video Game Design”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. (Florence, Italy). CHI 2008. New York, NY, USA: ACM, Apr. 2008, pp. 1453–1462 (cit. on pp. 10, 16, 19, 24, 54, 58).
- [67] David Pinelle, Nelson Wong, and Tadeusz Stach. “Using Genres to Customize Usability Evaluations of Video Games”. In: *Proceedings of the 2008 Conference on Future Play: Research, Play, Share*. (Toronto, Ontario, Canada). FuturePlay 2008. New York, NY, USA: ACM, Nov. 2008, pp. 129–136 (cit. on p. 10).
- [68] Karolien Poels, Wijnand IJsselsteijn, and Yvonne de Kort. *Development of the Kids Game Experience Questionnaire*. Poster presented at the Meaningful Play Conference, East Lansing, USA, abstract in proceedings. 2008. URL: [http://www.gamexplab.nl/includes/pages/publications/posters/Poels\\_2008\\_MeaningfullPlay\\_poster.pdf](http://www.gamexplab.nl/includes/pages/publications/posters/Poels_2008_MeaningfullPlay_poster.pdf) (cit. on p. 23).
- [69] David Poulson, Martin Ashby, and Simon Richardson. “Direct Observation”. In: *USERfit: A Practical Handbook on User Centred Design for Assistive Technology*. HUSAT Research Institute for the European Commission, 1996, pp. 33–35 (cit. on p. 20).
- [70] Scott Rogers. *Level Up!: The Guide to Great Video Game Design*. Wiley Publishing, 2010 (cit. on pp. 3–8, 17, 18, 52).
- [71] Katie Salen and Eric Zimmerman. *Rules of Play - Game Design Fundamentals*. Massachusetts London, England: The MIT Press Cambridge, 2004 (cit. on pp. 10, 11, 95).
- [72] Sreelata Santhosh and Mark Vaden. “Telemetry and Analytics Best Practices and Lessons Learned”. In: *Game Analytics: Maximizing the Value of Player Data*. Ed. by Magy Seif El-Nasr, Anders Drachen, and Alessandro Canossa. London: Springer-Verlag, 2013. Chap. 6, pp. 85–109 (cit. on pp. 29–31).

- [73] Jesse Schell. *The Art of Game Design: A Book of Lenses*. San Francisco, CA, USA: Morgan Kaufmann, 2008 (cit. on pp. 8, 54).
- [74] Mark A. Schmuckler. “What Is Ecological Validity? A Dimensional Analysis”. In: *Infancy* 2.4 (Oct. 2001) (cit. on p. 20).
- [75] Henrik Schoenau-Fog. “The Player Engagement Process -- An Exploration of Continuation Desire in Digital Games”. In: *Think Design Play: The Fifth International Conference of the Digital Research Association (DIGRA)*. (Hilversum, The Netherlands). DiGRA 2011. Hilversum, the Netherlands: DiGRA/Utrecht School of the Arts, Sept. 2011 (cit. on p. 12).
- [76] Eric Schuh et al. “TRUE Instrumentation: Tracking Real-Time User Experience in Games”. In: *Game Usability: Advancing the Player Experience*. Ed. by Katherine Isbister and Noah Schaffer. Morgan Kaufmann, 2008. Chap. 15, pp. 237–265 (cit. on p. 32).
- [77] Gillian Smith, Mee Cha, and Jim Whitehead. “A Framework for Analysis of 2D Platformer Levels”. In: *Proceedings of the 2008 ACM SIGGRAPH Symposium on Video Games*. (Los Angeles, California, USA). Sandbox 2008. New York, NY, USA: ACM, 2008, pp. 75–80 (cit. on pp. 4, 6–8).
- [78] Gillian Smith et al. “Rhythm-Based Level Generation for 2D Platformers”. In: *Proceedings of the 4th International Conference on Foundations of Digital Games*. (Orlando, Florida, USA). FDG 2009. New York, NY, USA: ACM, Apr. 2009, pp. 175–182 (cit. on p. 8).
- [79] Steve Swink. *Game Feel: A Designer’s Guide to Virtual Sensation*. Burlington, MA, USA: Morgan Kaufmann, 2009 (cit. on p. 7).
- [80] Anders Tyachsen. “Crafting User Experience via Game Metrics Analysis”. In: *Proceedings of the Workshop Research Goals and Strategies for Studying User Experience and Emotion at the 5th Nordic Conference on Human-computer interaction: building bridges*. (Lund, Sweden). NordiCHI 2008. Workshop Paper. Oct. 2008 (cit. on pp. 11, 20, 24, 25, 28–31).
- [81] Anders Tyachsen and Alessandro Canossa. “Defining Personas in Games Using Metrics”. In: *Proceedings of the 2008 Conference on Future Play: Research, Play, Share*. (Toronto, Ontario, Canada). Future Play 2008. ACM, 2008, pp. 73–80 (cit. on p. 24).
- [82] Mark J. P. Wolf. “Video Game Genres”. In: *The Video Game Explosion: A History from Pong to Playstation and Beyond*. Ed. by Mark J. P. Wolf. Greenbook Press, 2008. Chap. 38, pp. 259–276 (cit. on pp. 3–5).

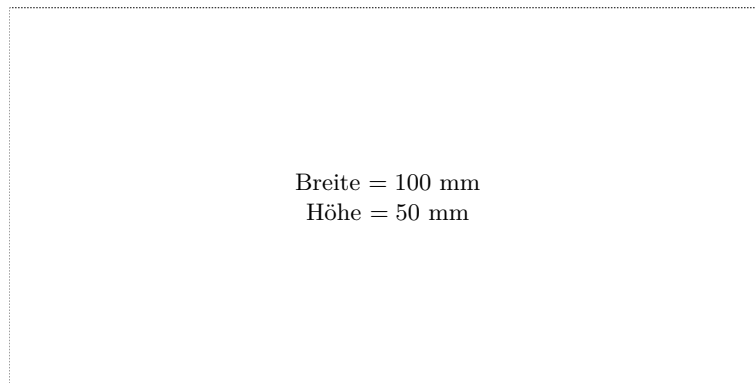
## Online sources

- [83] Sean Baron. *Cognitive Flow: The Psychology of Great Game Design*. [Online; accessed 2014-26-08]. Mar. 2014. URL: [http://www.gamasutra.com/view/feature/166972/cognitive\\_flow\\_the\\_psychology\\_of\\_.php?print=1](http://www.gamasutra.com/view/feature/166972/cognitive_flow_the_psychology_of_.php?print=1) (cit. on p. 13).
- [84] Daniel Boutros. *A Detailed Cross-Examination of Yesterday and Today's Best-Selling Platform Games*. [Online; accessed 2014-08-28]. 2006. URL: [http://www.gamasutra.com/view/feature/1851/a\\_detailed\\_crossexamination\\_of\\_.php?print=1](http://www.gamasutra.com/view/feature/1851/a_detailed_crossexamination_of_.php?print=1) (cit. on p. 8).
- [85] Statistic Brain. *Xbox Statistics*. [Online; accessed 2014-04-11]. 2013. URL: <http://www.statisticbrain.com/xbox-statistics/> (cit. on p. 33).
- [86] Phillip Derosa. *Tracking Player Feedback To Improve Game Design*. [Online; accessed 2014-08-20]. Aug. 2007. URL: [http://www.gamasutra.com/view/feature/1546/tracking\\_player\\_feedback\\_to\\_.php?print=1](http://www.gamasutra.com/view/feature/1546/tracking_player_feedback_to_.php?print=1) (cit. on p. 11).
- [87] Anders Drachen. *What are Game Metrics?* [Online; accessed 2014-01-08]. July 2012. URL: <http://blog.gameanalytics.com/blog/what-are-game-metrics.html> (cit. on p. 26).
- [88] Anders Drachen, Alessandro Canossa, and Magy Seif El-Nasr. *Intro to User Analytics*. [Online; accessed 2014-03-04]. May 2013. URL: [http://www.gamasutra.com/view/feature/193241/intro\\_to\\_user\\_analytics.php?print=1](http://www.gamasutra.com/view/feature/193241/intro_to_user_analytics.php?print=1) (cit. on pp. 10, 19, 24–29, 54).
- [89] Robert B. Frary. *A Brief Guide to Questionnaire Development*. [Online; accessed 2014-09-01]. URL: <http://www.ericae.net/ft/tamu/vpiques3.htm> (cit. on p. 67).
- [90] GameAnalytics. *GameAnalytics*. [Online; accessed 2014-01-08]. 2013. URL: <http://www.gameanalytics.com/> (cit. on p. 26).
- [91] *GameStage*. [Online; accessed 2014-07-23]. 2014. URL: <http://gamestage.radiatedpixel.com/about/> (cit. on p. 67).
- [92] Palle Steve Hoffstein. *Platformer Level Design*. [Online; accessed 2014-03-06]. URL: <http://phoffstein.wordpress.com/essays/platformer-level-design/> (cit. on pp. 7–9, 17, 18).
- [93] Diorgo Jonkers. *11 Tips for Making a Fun Platformer*. [Online; accessed 2014-03-07]. Jan. 2011. URL: <http://devmag.org.za/2011/01/18/11-tips-for-making-a-fun-platformer/> (cit. on pp. 3, 7, 8, 18, 52).
- [94] Diorgo Jonkers. *13 More Tips for Making a Fun Platformer*. [Online; accessed 2014-03-07]. 2012. URL: <http://devmag.org.za/2012/07/19/13-more-tips-for-making-a-fun-platformer/> (cit. on pp. 8, 17, 18).
- [95] *Jump 'n' Run*. [Online; accessed 2014-03-07]. URL: [http://de.wikipedia.org/wiki/Jump\\_'n'\\_Run](http://de.wikipedia.org/wiki/Jump_'n'_Run) (cit. on pp. 3, 4).

- [96] Michael Klappenbach. *What is a Platformer?* [Online; accessed 2014-03-07]. URL: [http://compactiongames.about.com/od/gameindex/a/platformer\\_def.htm](http://compactiongames.about.com/od/gameindex/a/platformer_def.htm) (cit. on p. 3).
- [97] Square Enix Holdings CO. LTD. *Square Enix Holdings: History*. [Online; accessed 2014-03-28]. URL: <http://www.hd.square-enix.com/eng/company/history.html> (cit. on p. 36).
- [98] Jamie Madigan. *The Psychology of Immersion in Video Games*. [Online; accessed 2014-08-20]. July 2010. URL: <http://www.psychologyofgames.com/2010/07/the-psychology-of-immersion-in-video-games/> (cit. on p. 12).
- [99] Jakob Nielsen. *How to Conduct a Heuristic Evaluation*. [Online; accessed 2014-06-10]. Jan. 1995. URL: <http://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/> (cit. on p. 23).
- [100] Seb Parker. *Halo: A Sales History*. [Online; accessed 2014-04-11]. 2011. URL: <http://www.vgchartz.com/article/87043/halo-a-sales-history/> (cit. on p. 33).
- [101] Alper Sarıkaya. *Manu Ganu*. 2013. URL: <https://play.google.com/store/apps/details?id=com.Alper.Manuganu&hl=de> (visited on 03/07/2014) (cit. on p. 5).
- [102] *Spielst du noch, oder baust du schon?* [Online; accessed 2014-07-24]. 2013. URL: <http://gamestage.radiatedpixel.com/spielst-du-noch-oder-baust-du-schon/> (cit. on p. 68).
- [103] Clive Thompson. *Halo 3: How Microsoft Labs Invented a New Science of Play*. [Online; accessed 2014-03-07]. Aug. 2007. URL: [http://archive.wired.com/gaming/virtualworlds/magazine/15-09/ff\\_halo?currentPage=all](http://archive.wired.com/gaming/virtualworlds/magazine/15-09/ff_halo?currentPage=all) (cit. on pp. 34, 35).
- [104] *Veranstaltungsarchiv*. [Online; accessed 2014-07-24]. 2014. URL: <http://gamestage.radiatedpixel.com/veranstaltungsarchiv/> (cit. on p. 68).

# Messbox zur Druckkontrolle

— Druckgröße kontrollieren! —



— Diese Seite nach dem Druck entfernen! —