Heuristics for evaluating the usability of video game tutorials

Tuomas Yli-Kovero

Master's Thesis

University of Turku

Department of Future Technologies

2017

Abstract

UNIVERSITY OF TURKU

Department of Future Technologies

Contents

Chapter 1

Introduction

It seems pretty common that when we think about software development, the thoughts usually wander towards different kinds of agile principles, object-oriented programming, functional programming, design patterns and so on. Whatever the case may be, the role of usability seems to lie somewhere higher on the hierarchy of the creation process. That is to say, usability is not necessarily involved in the process from the ground up, and this can have a detrimental effect on the performance of the software, no matter how powerful the underlying solutions might be; it is still usually people using it to the best of their ability, trying to take advantage of the underlying implementation. Because information is communicated to the user, the way things are presented matters, and people have become more usability conscious, partly due to the rise of smartphones and other touch devices [Chen et al., 2015]. Usability is important also for a type of software called video games, and the type of usability evaluation methods that could or should be used in game development is a debated subject [Bernhaupt et al., 2007]. Even though games are mostly seen as entertainment, bad games are hardly entertaining. The potential frustration of the player can be countered with properly guiding the player into the game, and making sure they learn the necessary properties and mechanics of the game to acquire sufficient mastery of the system in order to experience it in

a meaningful way. This seems important also because it has been found that many players give up on a game during the very first hours [Bauckhage et al., 2012]. One hypothesis then is that in order to make the player stick with the game we have to make sure the usability factor is not ignored during these first hours. Both the understanding of gameplay and narrative are important factors in player retention [Cheung et al., 2014]. A common theme for many games is that they introduce a tutorial at the beginning of the game which aims—or at least should aim—to teach the basic mechanics and necessary interface elements, anything that is fundamental to the basic gameplay.

The aim of this thesis is to provide a heuristic framework for performing an expert review on video game tutorial usability, based on selected video game tutorials and existing research on game usability heuristics, as opposed to a usability evaluation with a group of test users. A hypothesis for whether video game tutorials are usable does not feel intuitive as such, i.e. it could go either way, and of course depends on the game in question (some tutorials probably are usable and some are not). As mentioned before, based on some research the first few hours anyone spends with a game can be critical for player retention, so the time bracket for the potential of tutorials as something that will encourage the player to keep going seems significant, as a tutorial is usually the first thing the player encounters in a game: "The first time a player sits down with a game is critical for their engagement. Games are a voluntary activity and easy to abandon. If the game cannot hold player attention, it will not matter how much fun the game is later on if the player quits early." [Cheung et al., 2014].

Chapter 2

Theoretical concepts

This chapter will go over the most important concepts in the thesis before we apply them in practice in the later chapters. We look at what is usability and how it relates to HCI (human–computer interaction), i.e. software and video games, and what are heuristics in the context of usability. We also provide an overivew of tutorials. From an HCI perspective, we can say that we are performing ontological games research, meaning we are mostly thinking about game design and interfaces among other things. The ontology of games is also said to consist of rules, aesthetic, fiction and game design patterns. [Carter et al., 2014]. Delimiting where a tutorial falls along these ontological concepts is not unambiguous. We can also think about a tutorial as a bridge between the ontology of a game and the player.

2.1 Usability

Usability is a word that can come up often in conversation. It sounds familiar and should be rather easily explained. However, it is not necessarily that simple in this contex. Is something usable? If we can use something, does that mean that the artifact is usable? How can we measure this? Is there good usability and bad usability? Of course we can just say something is usable, but that does not necessarily tell us

anything more than that there exists something we can interact with. To further complicate the issue, usability does not even necessarily mean the end product or the user interface the user will be interacting with; we can also apply usability guidelines to the actual software development processes [Carvajal et al., 2013]. Usability also relates closely to design. We can talk about Norman doors—doors that are so badly designed and unusable that we can't figure out how to open them—derived from Donald Norman's classic book *The Design of Everyday Things* [Norman, 2013]. In this sense usability is just not something that exists, but is required. We can not take it away or separate it from the under-the-hood functionality, or it becomes pointless since we are unable to utilize it. Let us say we have a microwave oven that works perfectly, but we remove all the buttons and displays from the front panel. It is still capable of cooking things, but it is really hard to enable that functionality since we have really poor options for interaction. Perhaps we could screw the whole thing open and try to apply some MacGyverisms and ad hoc solutions in order to produce a warm meal, but that would most likely feel highly unusable. The next step could be that we add one button to the front panel that turns the oven on and off. We would also need a way to open the oven's door. After that all kinds of additional things come to mind that we in a way take for granted. At some point we will start to approach the other end of the usability spectrum: we have too many things and also things that are irrelevant. Things that only come in the way of the core functionality we want to enable or convey. Some might feel that the best kind of microwave would be the one that has only one button to turn it on and off, many might want to be able to change the power and add a timer and other typical things we might have in microwaves. The core issue, however, is that without adding usability to the object, its existence becomes in a sense pointless. Taking this idea of usability and bringing it to many different areas in life—from doors and microwaves to video games and many things in between—we can start to appreciate the value it gives us in the tools and entertainment we come into contact

with in a weekly and even a daily basis. This also brings forth the idea, that no matter what kind of great tools and things we are able to create, they will make no difference unless we think about their usability. Through usability we will strive to increase and maximize the potential of anything we have decided to make.

One classic view on usability comes from Jakob Nielsen in the form of a usability definition and a list of 10 usability heuristics for user interace design. We will look at heuristics more closely in the following chapters. Nielsen defines usability as five quality components [Nielsen, 2012]:

- Learnability: How easy is it for users to accomplish basic tasks the first time they encounter the design?
- Efficiency: Once users have learned the design, how quickly can they perform tasks?
- Memorability: When users return to the design after a period of not using it, how easily can they reestablish proficiency?
- Errors: How many errors do users make, how severe are these errors, and how easily can they recover from the errors?
- Satisfaction: How pleasant is it to use the design?

Nielsen also mentions a sixth attribute for usability he calls *utility*. It is an important attribute to discern, since utility is at the core of *why* something is made in the first place. We can imagine something that takes into account all the principles of good design and usability, is beautiful in every way and a pleasure to use, but does not really do any or most of the things we need it to or designed it to do. Through this marriage of usability and utility, Nielsen comes to define whether something is actually *useful* [Nielsen, 2012]:

- Definition of Utility = whether it provides the features you need.

- Definition of Usability = how easy and pleasant these features are to use.
- Definition of Useful = usability + utility.

We also have to keep in mind that here, usability can refer to any type of design and design process, not only something related to software and games. In the following sections, usability is dissected in a more specific context, i.e. how it has been defined in relation to software and video games.

2.1.1 Usability in software

Standards have been created to help with designing usable interfaces for software. Two central standard bodies involved in developing standards for HCI (Human–computer interaction) and usability are International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC). The word *usability* has been summarized in a standard as follows:

> *Usability*: the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use – ISO 9241-11: GUIDANCE ON USABILITY (1998) [ISO, 1998]

Later, we will attempt to draw a line from the mentioned user satisfaction to player satisfaction in games. Standards for HCI and usability are generally categorized as follows [Bevan, 2006]:

1. The use of the product (effectiveness, efficiency and satisfaction in a particular context of use).
2. The user interface and interaction.
3. The process used to develop the product.

4. The capability of an organization to apply user-centered design.

This structure shows us the way in which usability is generated into the use of the product (1) starting from the capability of an organization (4).

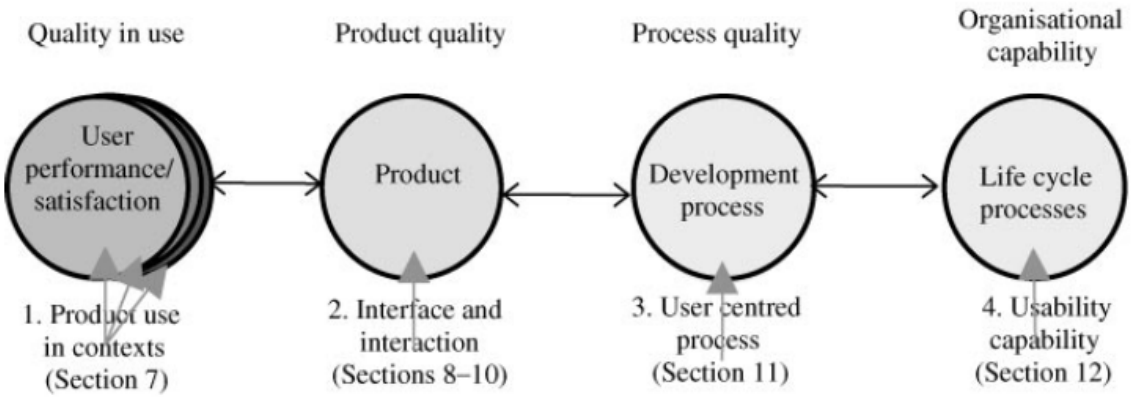

Figure 2.1: The logical relationships of standards related to usability [Bevan, 2006]

It is further exemplified in Figure 2.1, illustrating the logical relationships: "the objective is for the product to be effective, efficient and satisfying when used in the intended contexts. A prerequisite for this is an appropriate interface and interaction. This requires a user-centred design process, which to be achieved consistently requires an organizational capability to support user-centred design." [Bevan, 2006] When we look at ISO 9241-11, there is a promising toolkit for how to take usability into account, specifically considering user performance and satisfaction, but also the context of the system. They claimed that it is also possible to derive factors affecting the quality of the entire system from these variables of user performance and satisfaction. [Bevan, 2006] Interestingly, this can lead us to think about the quality of video game tutorials, and how they might function as this mediating component between the user (player) and the actual game they are attempting to learn – or perhaps even more so, are being taught. There have been further definitions of usability as a software engineering standard by ISO and IEC, namely ISO/IEC 9126-1, which defined usability as a set of attributes that bear on the effort needed for use,

and on the individual assessment of such use, by a stated or implied set of users. [Bevan, 2006] This has been later replaced with the ISO/IEC 9126-1 which has a new definition [Bevan, 2006]:

> *Usability*: the capability of the software product to be understood, learned, used and attractive to the user, when used under specified conditions.

This new standard also brought attention to the important aspect of utility, similarly as to how Nielsen has defined it, that in the context of usability there is not necessarily much use to define usability per se, but rather realize how it always operates in a given context and aims to maximize the utility of the underlying capabilities, hence making usability valuable only if the requirements made by the specified conditions are fulfilled. The ISO/IEC 9126-1 also uses another term, *quality in use*, attempting to combine the ideas of usability in general but also the importance of context [Bevan, 2006]:

> *Quality in use*: the capability of the software product to enable specified users to achieve specified goals with effectiveness, productivity, safety and satisfaction in specified contexts of use.

These definitions are bringing us closer to how we can think about usability in the context of video games. Interestingly, with virtual reality (VR) technology currently becoming more widely available, even the safety aspect, originally perhaps geared more towards industry applications, can be a valuable thing to take into account, and is present in many existing usability models [Dubey et al., 2012]. Another interesting take on usability are the so called dark patterns. They abuse basic human psychology and perception to actually make the user do the opposite thing they think they are doing, to the author's benefit [Brignull, 2013]. Thus usability becomes an even more important part of product design when we realize we do not want to have the user make the opposite action by accident.

2.1.2 Usability in video games

It has been said that all software applications are tools that help users accomplish certain tasks, but is this true for games as well? How should we look at games in relation to other software and specifically heuristics designed to be used with the so called productivity software. In both cases we need information about who are the users of the software are and what they need to accomplish. [Ferré et al., 2001] Is there a parallel between efficient spreadsheets and efficient fun (since the claim could be made that games mainly aim to accomplish the goal of the user having fun while using the software)? Game developers making games for other game developers and video game enthusiasts at large does not necessarily bring forth the question of usability very early in the process, although it has been said that usability should be taken into consideration even before prototyping software [Holzinger, 2005]. It is when broader audiences are sought, and perhaps people who have never played games before, when the question becomes more important. [Isbister and Schaffer, 2008] This notion is also important because it has been observed that the type of user has a big impact on the level of perceived usability, not just how the usability would be rated in general [Cavallin et al., 2007]. An experienced player might be able to see where a tutorial or game is going and be fine with it, even if it would be rated as lacking of required characteristics for at least novice users. They might even be happier with the lack of instruction in a familiar setting, which might just feel like it is getting in the way. This is one reason the ability to turn tutorials off is discussed later.

In the context of usability, the term user satisfaction comes up often. A host of problems can be found in the usability of different video game genres [Pinelle et al., 2008b]. The forementioned *effectiveness* and *efficiency* can also be related to games, but being first and foremost a medium for entertainment and storytelling, rather than a tool for solving problems—i.e. some professional productivity software—it might be important to think about how to achieve player satisfaction first. This is important

in part because games are first and foremost a form of entertainment and are acquired voluntarily. A game that is not fun will most likely not be very succesful. [Federoff, 2002] In this sense, we could think about effectiveness and efficiency as something that can help us deliver satisfaction. That is to say, we create a tutorial that is effective and efficient in teaching the player how to play the game, which brings us to create player satisfaction. There is also an important difference between video games and other types of software: we could say that generally, video games almost always come with an integrated tutorial, whereas most other types of software do not. Why is this? Could it be that games are usually meant to be played in certain predefined ways, and the rest, more or less productive types of software can be used for different tasks depending on the environment? Spreadsheet software, web browsers and other common tools rarely teach you how to use them when first opened. It can be that they have become so commonplace that if guidance is required, it is brought from an outside entity, such as a consultant coming into a company to give training in a specific software. Another thing is that these types of software are generally meant to be used indefinitely, whereas games are completed; the story comes to a close, a character is leveled to the highest possible level, all functionality (character skills etc.) is finally unlocked, every puzzle solved and so on. This means that a tutorial can be created for a set of predefined outcomes and mechanics. Surely there can be a tutorial on how to enter numbers in a spreadsheet, but the possibilities of using those numbers to your advantage are so numerous that creating a tutorial within the software that would show you everything is not necessarily a justified way to spend resources on.

What, then, is the significance of usability in games? A set of predefined outcomes and/or mechanics is not necessarily a huge platform to build on, which means that creating meaningful and useful tutorials should be easier. This is not to say that all games are simple and easy. It is possible to create a system of numerous possibilities where the player operates with simple mechanics. Since there are so

many genres of games, this can often be a case by case problem. Even within a genre, games often like to reinvent the wheel and introduce some type of new gimmicks previously unfamiliar to the genre, to provide a sense of freshness and novelty. It is through these mechanics and the interfaces we use to interact with them we start to create the basis for a need for a tutorial: which mechanics to present and the best way to teach them to a new player.

2.2 Heuristics in usability

"Heuristic evaluation – a technique where experts inspect software and determine where the application violates predetermined policies for good usability – is an effective technique for evaluating productivity software." [Livingston et al., 2010]

So how would we then evaluate a tutorial in a given video game? There are a few possibilities to do this, of which popular are expert review and user or usability testing. Since the aim of this thesis is to provide an expert review solution, we will not be looking at usability testing with a group of users. Rather, we need to build a set of heuristics that is based on some existing video game usability heuristics, and also go through a list of games with our knowledge of usability principles and define a set of heuristics that can be better used with evaluating the usability of video game tutorials. But what exactly is a heuristic? The Merriam-Webster dictionary defines the word *heuristic* as follows [Merriam-Webster, 2017a]:

- Involving or serving as an aid to learning, discovery, or problem-solving by experimental and especially trial-and-error methods
- Heuristic techniques
- A heuristic assumption; also : of or relating to exploratory problem-solving techniques that utilize self-educating techniques (such as the evaluation of

feedback) to improve performance

- A heuristic computer program

This is still a very broad definition and does not necessarily tell us much here. One way is to think of a heuristic as a shortcut or a guideline-based method [Isbister and Schaffer, 2008]. Heuristics in e.g. psychology stands for shortcuts people use to tackle complex problems with incomplete information [Kahneman et al., 1982]. In the context of usability we can think of heuristics as a design guideline that we can use as tools for evaluation, which traditionally relates to user interfaces. The goal here is to make the interface easy to learn, use and master, opposing the usual game design goal of "easy to learn, difficult to master". [Desurvire et al., 2004] It's not necessarily a good idea to make the interface difficult to learn, even if gameplay-wise this can often be a good choice (meaning that as the mastering of a game can be difficult, it should not necessarily be unnecessarily difficult to control). Desurvire et al. further state that we can not only think about games through the interface: we must evaluate other factors as well, such as game play, story and mechanics [Desurvire et al., 2004]. In the third chapter we will be looking at these heuristics in more detail. Nielsen and Molich [1990] have defined four ways to evaluate a user interface:

- Formally by some analysis technique
- Automatically by a computerized procedure
- Empirically by experiments with test users
- Heuristically by simply looking at the interface and passing judgement according to ones own opinion

Now, we do not want to simply look at some games and shout out some opinions based on how we feel like. It would be more useful to base it on some existing

heuristics about usability, but usability heuristics for video game tutorials are rare. Therefore, we can find some existing heuristic lists for usability evaluation in general, and combine these lists to better suit the evaluation of video game tutorials. In their paper *Heuristic evaluation of user interfaces* Nielsen and Molich also provide a subset of principles to be used for the evaluation in question: [Nielsen and Molich, 1990]

- Simple and natural dialogue
- Speak the user's language
- Minimize user memory load
- Be consistent
- Provide feedback
- Provide clearly marked exits
- Provide shortcuts
- Good error messages
- Prevent errors

At this point these principles serve more as an example to describe what kind of heuristics can be used in a heuristic evaluation. We will have a large pool of different heuristics which we will combine to use in our expert review of tutorials. Some bullet points here might feel self-evident and some might not feel right for the context of video game tutorials, so we must make decisions on whether to include any given heuristics for our analysis in question. Furthermore, such lists are highly valuable for an evaluation, since we don't have to rely on our intuition alone, but have a scientific basis for evaluating different artifacts. Nielsen and Molich have also defined other advantages for using heuristic evaluation: [Nielsen and Molich, 1990]

- It is cheap
- It is intuitive and it is easy to motivate people to do it
- It does not require advanced planning
- It can be used early in the development process

However, it is not only always positives regarding heuristic evaluation, especially when performed by a single person. It has been concluded that at least in some cases heuristic evaluation would be best done with three to five evaluators separately, and it can also be difficult to come up with solutions to the usability problems found when using a heuristic approach. [Nielsen and Molich, 1990] On the other hand, when we are going to look at video game tutorials, we will have a combination of a number of different heuristics to use, and also games from different genres, to which some forms of heuristics might be better suitable than others. The studies by Nielsen and Molich were done in part on static screen dumps and old voice response systems using telephone buttons. Those systems are not exactly similar to modern video games, so we can not yet say with certainty how a heuristic approach might work for video game tutorials specifically, even if it has not always worked well for other types of systems and interfaces. This could become even more evident when we compare different video game genres and the types of tutorials therein. In that regard, applying a combination of different heuristics to different types of contexts (i.e. video game genres) we can hope to expect many different types of outcomes for the heuristic approach.

2.3 The role of the tutorial in a video game

The word tutorial can be interpreted as the transferring of knowledge or practical information [Merriam-Webster, 2017b]. It doesn't necessarily then mean that there

would have to be a separate tutorial section in the game, but rather the tutorial can be happening occasionally many hours into the game, as new mechanics or strategies etc. are introduced. Another way to think about this is to have the tutorial be explicit or implicit, i.e. there is either a separately designed optional tutorial level (explicit) or the tutorial is part of the actual levels of the game (implicit) [Adams, 2013]. We could even not include a tutorial at all, and use a passive method of teaching the game's mechanics through level design [Moss, 2017]. This would mean that that when we talk about a game's tutorial, we should think about it as the way the game teaches you about how it is played, not whether there is a separate tutorial section or not. In any case, tutorials can have a big effect on player retention and how players learn video games [Andersen et al., 2012]. In the chosen games section of this thesis we can see that most of these games do not in fact have a separate, or explicit, tutorial. It is in this case more common to teach the player as he or she goes along.

Chapter 3

Methods for expert review

Previously we have talked about what heuristics are and how they relate to usability testing. The way we will approach tutorials here is not by using a group of test users, even though there are fast methods developed for that specific purpose [Kjeldskov et al., 2004]. In other instances, it is recommended to use all possible methods to evaluate usability [Schaffer, 2007]. There are many different games to go through, and user-centered usability testing revolves closely around the context of use [Tarkkanen et al., 2013]. In the case of user-centered usability testing the context of use would be more closely related to testing and improving the usability of a single game. Our goal is to find new heuristics and combine them with existing ones, so an expert review approach on a number of different games serves this purpose better. In this chapter the focus is on finding a set of applicable heuristics for evaluating video game tutorials, and constructing a set of heuristics from that pool to use in our expert review heuristics for tutorials. We also have to form a selection of games that will be the target of our evaluation.

3.1 Choosing heuristics

There are a number of papers and studies on the use of different heuristics in video game research and testing. Our problem here is that they are mostly related to the general game experience and how the game plays from "start to finish" in a sense. Even in a study with a large number of heuristics there might only be a single tutorial related heuristic, one that says it should exist [Almeida et al., 2010]. We have a rather specific part of a game — the tutorial — that we want to evaluate, and not all heuristics are applicable or specific enough to be used with the part in question. For example, what is generally called the Nielsen's 10 heuristics are as follows [Nielsen, 1993]:

1. Visibility of system status
2. Match between system and the real world
3. User control and freedom
4. Consistency and standards
5. Error prevention
6. Recognition rather than recall
7. Flexibility and efficiency of use
8. Aesthetic and minimalist design
9. Help users recognize, diagnose and recover from errors
10. Help and documentation

This is a very broad set of heuristics that can be applied to user interface design in a rather straightforward manner. Since we will be looking at video games and

their tutorials, it can be useful to look for a more specific set of heuristics. After a literature review on the topic there are a number of sources we will be using to combine our heuristics. [Desurvire et al., 2004], [Desurvire and Wiberg, 2009], [Federoff, 2002], [Pinelle et al., 2008a] The important thing here is to remember, that not all of these heuristics are applicable to tutorials, so we must dissect them a little bit, all the while keeping in mind the different types of video game genres they could be applied to. What follow are tutorial-specific compiled lists from separate heuristic guidelines for video game usability. Heuristic Evaluation for Playability (HEP) is a heuristic method for analyzing the usability and playability of games [Desurvire et al., 2004]. It contains 43 items but not all of them can be directly applied to tutorials. Here is a list of heuristics, separated from the whole HEP, that is closely associated with tutorials and their design. The items are in the same relative order.

1. Provide clear goals, present overriding goal early as well as short-term goals throughout play.

2. There is an interesting and absorbing tutorial that mimics game play.

3. The first player action is painfully obvious and should result in immediate positive feedback.

4. Pace the game to apply pressure but not frustrate the player. Vary the difficulty level so that the player has greater challenge as they develop mastery. Easy to learn, hard to master.

5. Mechanics/controller actions have consistently mapped and learnable responses.

6. Shorten the learning curve by following the trends set by the gaming industry to meet users expectations.

7. Controls should be intuitive, and mapped in a natural way; they should be customizable and default to industry stan- dard settings.

8. Player should be given controls that are basic enough to learn quickly yet expandable for advanced options.

9. Provide immediate feedback for user actions.

10. The player can easily turn the game off and on, and be able to save games in different states.

11. The player experiences the user interface as consistent (in control, color, typography, and dialog design) but the game play is varied.

12. Players should be given context sensitive help while playing so that they do not get stuck or have to rely on a manual.

13. Sounds from the game provide meaningful feedback or stir a particular emotion.

14. Players do not need to use a manual to play game.

15. Get the player involved quickly and easily with tutorials and/or progressive or adjustable difficulty levels.

The original HEP contains 43 heuristics in total in four categories: Game Play, Game Story, Mechanics and Usability. Tutorial-specific heuristics could be found in all categories except Game Story. HEP was revisited in 2009 and was named Game Usability Heuristics (PLAY) for Evaluating and Designing Better Games: The Next Iteration [Desurvire and Wiberg, 2009]. The categories in it have been reduced to three and are as follows: Game Play, Coolness/Entertainment/Humor/Emotional Immersion and Usability & Game Mechanics. There are subcategories added to each three, which makes the heuristics a little more accurately categorized. We

can see that even though there is still a separate category for usability, there are other heuristics that can be applied to tutorials as well, that can make the tutorial experience better but are not necessarily usability-related heuristics. In other words, a broader view not related only to usability can help make tutorials better. Below is a list of heuristics compiled from PLAY that we can use to evaluate tutorials.

1. Easy to learn, hard to master.
2. The game goals are clear. The game provides clear goals, presents overriding goals early as well as short term goals throughout game play.
3. The skills needed to attain goals are taught early enough to play or use later, or right before the new skill is needed.
4. The first ten minutes of play and player actions are painfully obvious and should result in immediate and positive feedback for all types of players.
5. Players feel in control.
6. Player does not need to read the manual or documentation to play.
7. Player does not need to access the tutorial in order to play.
8. Game controls are consistent within the game and follow standard conventions.
9. Status score Indicators are seamless, obvious, available and do not interfere with game play.
10. Controls are intuitive, and mapped in a natural way; they are customizable and default to industry standard settings.
11. Consistency shortens the learning curve by following the trends set by the gaming industry to meet users expectations. If no industry standard exists, perform usability/playability research to ascertain the best mapping for the majority of intended players.

12. Game provides feedback and reacts in a consistent, immediate, challenging and exciting way to the players actions.

13. Provide appropriate audio/visual/visceral feedback (music, sound effects, controller vibration).

14. Player is given controls that are basic enough to learn quickly, yet expandable for advanced options for advanced players.

15. Player interruption is supported, so that players can easily turn the game on and off and be able to save the games in different states.

16. Upon turning on the game, the player has enough information to begin play.

17. Players should be given context sensitive help while playing so that they are not stuck and need to rely on a manual for help.

18. All levels of players are able to play and get involved quickly and easily with tutorials, and/or progressive or adjustable difficulty levels.

It is apparent that some parts have remained the same or have been reworded, and also new heuristics have been introduced. The heuristic "there is an interesting and absorbing tutorial that mimics game play" from the original HEP has been left out, but some games do not necessarily offer any kind of help to the player, so it is a good heuristic to include when researching games.

The heuristics compiled by Federoff [2002] are very similar to the previous ones. In the following list we will leave out the ones that are similar to the ones already presented in HEP and PLAY.

1. Get the player involved quickly and easily.

2. The game should give hints, but not too many.

As we can see, what is left is a lot shorter list of heuristics. Once again, in the following list the same operation has been performed to the heuristics compiled by Pinelle et al. [Pinelle et al., 2008a]

1. Allow users to skip non-playable and frequently repeated content.

2. Provide instructions, training, and help.

This list is not very long either. The basic principles for tutorials are somewhat scarce. When we are looking at a number of games in the following chapters, we will try to find elements that could be helpful to add in these heuristics, and also see which of the heuristics defined here are already applied or applicable to these tutorials.

It has been said that the majority of the forementioned Nielsen's heuristics are mostly helpful when analyzing user interfaces [Federoff, 2002]. In the game-related heuristics it seems that the emphasis is often on game play and general usability, but tutorials are not addressed very specifically, very often only in the form that there should be one. It is also interesting to note that even though the tutorial can be in its own category when heuristics are defined, it can still have many elements from different heuristics categories. It is not only that there is a tutorial element in the game, but also that the tutorial has elements from a wide range of elements in the gameplay heuristics. This means that tutorials can be evaluated e.g. from a gameplay perspective, not only from a tutorial or a usability perspective. The game goals being clear and that the skills needed to attain those goals are taught in an orderly fashion belong in a sense to gameplay heuristics, but those can be easily applied to tutorial settings as well. However, some of these heuristics are still too general and vague for considering tutorials. The final combined list of the necessary heuristics presented here is availabe in Appendix A.

3.2 Defining tutorials

As defined earlier, in general the word *tutorial* refers to transferring of knowledge. It can be used in different contexts. Here we are mostly concerned with computers and how video games are explained within. In games, it is possible that there is an entirely separate section that teaches the fundamentals of the game to the player. Another—if not perhaps the more common—design choice is to embed the tutorial in the regular gameplay. This can mean that the player is learning different ways to play hours into the game, and not only during some initial short introduction to the controls and mechanics etc. It can also help with the storytelling and drama when it's not clear from the very start what kind of things to expect from the game. The player might be wondering if they are ever going to get a gun for example. Then, after a couple of hours of playing and sneaking in a scary environment, they find a pistol and and experience relief. If the gun mechanics and controls would be introduced early on, it might not create tension in an optimal way.

3.3 Basis for selecting applicable video games

There needs to be some basic fundamentals for how we choose the video games we want to evaluate here in order to have a somewhat meaningful selection in relation to the results we arrive to. Based on an earlier study on video game usability testing, we can lay out the following defining criteria and go on to choose applicable games [Febretti and Garzotto, 2009]:

1. To be well known, professionally-developed, succesful titles published in the last ten years (which can be a potential indicator of long term engagement).
2. To be refereed by specialized web sites for game quality assessment.
3. To have at least one significant usability problem that clearly emerges at some point of the gameplay.

The time frame of the last ten years does not feel like something that should be very strict. There are great examples of games that are eleven years old, for example. Here we will just make the point of not playing only the latest games, but rather good or known games in general. We also can not confirm point three beforehand. It can be very likely that there are at least some kind of usability problems in every game we are going to look at, and e.g. some might not have a tutorial at all. The key point still is not to find errors. We want to look at games and their tutorials, see how they are made, what they feel like, what elements they have in common with previously defined heuristics and what new things can we learn from the games to apply to a new set of heuristics more closely related to video game tutorials.

Chapter 4

Chosen games and their tutorials

In this chapter, we will go through a list of video games and their tutorials. The tutorials are not identical in each, meaning that one game might have a dedicated tutorial section apart from the main game, where another game might have an integrated tutorial with the actual gameplay and gradually guiding the player in the game from there. Generally there are not usually separate tutorials, but rather the information is embedded in the beginning of the actual game. Perhaps this is to make sure that anyone playing the game gets the information they need, rather than having to find it from a tutorial separate from the main game, but it also depends on the game type. In a fighting game like Guilty Gear Xrd Revelator, the separate tutorial can be justified because there is actually only one game mode, which is two characters fighting. Adding a tutorial to a regular match would be difficult because it would possibly have to be constantly paused and would not feel like the actual gameplay anyway. On the other hand, a more complex tactical shooter like SWAT 4 has a separate tutorial section that shows you all the important mechanics in the game where you need to command your team and have a lot of controls and options for engagement, which also reflects the realistic setting in a more immersive way.

4.1 Initial impressions and takeaways

The goal here is to establish a general view on these games and how their tutorials work, if there is one. Observations will be made about the user interface, mechanics, presentation of information, the general flow and feel of the tutorial and anything else that comes to mind. As games are often very different from one another, no general style of approach will be defined at this point. Rather, we want to see how the tutorials operate and if there is anything we could think about as a heuristic method that could be applied to tutorial design in general. These will be laid out in the takeaways section of each game. All of the games are PC versions found on the Steam platform unless otherwise noted, the secondary platform being Android. Name of the developer and release year are also listed.

4.1.1 Dark Souls 2 (FromSoftware, 2014)

Genre: Action / RPG

Separate tutorial: No

The player can read stone tablets he comes across in the first area of the game, which show the controls separately, one or two basic controls in each tablet. However, they are not presented as something that the player will without a doubt come across. The tablets are scattered in the caves around the area, and a careless player could just run through the area without consulting them. When a tablet is in front of the player character, a prompt comes up that says to push a button to interact, and when the player does text is displayed about how to perform the action written on the tablet. The tablets contain a lot of basic controls and introduces enemies the player can try the controls out on. Which button to press to attack is introduced first, then how to lock a target, how to dash and so on. Other controls presented include how to perform critical hits, rolling, switching weapons, backstepping, how to use an item, how to move the camera around the character, dual-wielding and

a dashing jump for distance. Enemies and chasms to jump over come about as the player finds more tablets in the caves. Parrying, guard breaking and how to perform a plunging attack by dropping on an enemy are introduced last. There are basically no basic controls left untold, it is just up to the player to find them as it is in the core of the Dark Souls experience that there is no hand-holding. The player is expected to find out things on his own most of the time, often so by dying and trying again. When death is a game mechanic, it also feels like a design choice to leave the responsibility to the player to explore the first areas of the game as well as they can.

Takeaways: The tutorial reflects the intended gameplay experience. It is not supposed to be forced on the player, even in the early stage, since the philosophy of the game is to be unforgiving and requires research on the player's part in general.

4.1.2 Ori and the Blind Forest (Moon Studios, 2015)

Genre: Platformer

Separate tutorial: No

As the player moves for the first time, the game displays information to hold the A button on the gamepad longer to jump higher. Then to hold down and press A to 'jump down through platforms'. The game also let's the player know how collectibles work as they are found and how to hold the B button to save your game (saving is an in-game mechanic, not a separated menu functionality). Ori keeps introducing new mechanics many hours into the game, and tells you what the controls are in-game, as you unlock new abilities. There are areas that are not cleary yet accessible until you unlock new abilities (i.e. a double jump) to go there. The game will always prompt you to use the controls accordingly to get there the first time. It's never paused and the tutorial is always integrated to the gameplay. Ori is a certain type of game where the versatile gameplay is not revealed all at once, but through acquired

abilities, escalating the possibilities towards the end of the game and the tutorial follows that same progress as abilities are unlocked.

Takeaways: The tutorial reflects the pacing of the gameplay. Teach mechanics in the proper context when they are first needed, not all at once and before they can be even used in the game.

4.1.3 Heroes of the Storm (Blizzard Entertainment, 2015)

Genre: MOBA

Separate tutorial: Yes

Heroes of the Storm has two separate tutorials you can choose to play: one concerning character development (called "tutorial") and basic gameplay, other (called "battleground training") concerning the differences of arenas, that have their own special mechanics. Before the tutorial you have to choose which role to play: assassin, warrior or support. At the start of the tutorials, you are given a basic overview of the lanes each map has and your objective: to destroy the main structure at the enemy base. You are also told that at least one hero should be present on each line during play. The remainder of the tutorial has the following progress, each bullet point representing what information is given to the player::

- The overview of the character you chose, i.e. what kind of abilities she has.
- A "Tip Panel" on the top left of your screen which has the basic controls, and are told to familiarize yourself with them.
- A short cinematic about how you gain experience (xp) when killing enemy minions and are shown what some basic info on the screen means.
- information on how your abilities work and are thrown into battle to try them out.

- As you kill enemies and proceed on the map, you encounter the first turrets, and are told what their mechanics are, which feels very useful (too).
- The secondary objective(s) each battleground has.
- You're told to get to the secondary objective, and also how to ride a mount to get there faster.
- You're told how the secondary objective here, a watch tower, increases your teams visibility around it.
- You're told to defeat a camp of giants to make them fight on your side for a while down the nearest lane.
- You're told to go and destroy the enemy base, after which the tutorial ends.

In battleground training, you are able choose from the same three hero types (assassin, warrior and support). When the training starts, you get more information about keyboard shortcuts, such as how pressing tab shows you the score and status card. There's also a question mark present on the screen at all times from which you can review all the tutorial tips. Then you get told how to regain mana from fountains or how to teleport back to base to refill your stats. The tips often pause the game and wait until you've read them. With the tribute battleground mechanic, you're told very specifically what to do and how it affects the other team if you succeed. This is further enhanced by using specific markers to show locations on the minimap relevant to the tutorial. When you active the tribute and weaken the enemy team, you are told to attack the enemy and the tutorial ends when their main building is destroyed.

Takeaways: Have separate tutorials for different aspects of the gameplay if necessary (e.g. character development and arena-specific mechanics).

4.1.4 Psychonauts (Double Fine Productions, 2005)

Genre: 3D Platformer / Action-adventure

Separate tutorial: No

When you start a new game, it eventually throws you into a sequence where you are asked to perform controls, starting with a command to move the right stick towards the right. You are then told about the object you're looking—a collectible—and what it does in the game. After that, you are asked to look up, get told about another collectible there and the sequence ends. After this the game starts. When you first open the menu, you get a prompt how to move in the menu. When you go to a tab in the menu for the first time, you get a prompt that tells you what the menu is about, and then click a button the close the prompt. When you pick up things from the ground for the first time, you get a prompt that tells you what they are and then have to click a button to close the prompt. You can run around in the first area collecting stuff, but the next area is blocked until you complete an obstacle course that teaches you the rest of the controls. This obstacle course is made in the spirit of a real world obstacle course, such as the training area in SWAT 4, where actual psychonaut candidates are, or would be, trained. First you need to jump over things, then double jump over bigger distances. The basic training obstacle course is not an easy one and really tests the players abilities with jumping, climbing, punching and collecting things. During the course constant information is given to the player on how to perform these actions and what new collectibles mean. After the basic training course is completed, the player has been shown everything about the basic controls and gameplay mechanics and they gain access to rest of the areas in the game.

Takeaways: Teach basic movement controls interactively, perhaps even before the actual game starts, at least if the controls remain relativle similar for the rest of the game. Have the first area of the game be a tutorial section that needs to be

completed before the rest of the game becomes available.

4.1.5 Quake Live (id Software, 2010)

Genre: FPS

Separate tutorial: Yes

Quake Live has a training center where you can optionally get information on how the game is played. There are both videos that show you what you should be doing, and training where you get to try out the different controls yourself. There are three different training sections, called *crash course*, *accelerate* and *elevate*. The crash course shows you the basic controls on how to move and shoot, and is very thorough. You are told how long the tutorial will take and that you can exit at any time by pressing F3. Then you enter an arena with a non-player character (NPC) who gives you detailed information about ammo pickups, weapons, health the map in question and the general playstyle that is considered good (spawn, gear up, fight, restock). Then, a training match with the NPC commences and you fight her in order to complete the first part of the tutorial. The second and third part of the tutorial section teach you how to strafe jump and rocket jump, respectively. The training center apprently wants to make sure you are aware of these mechanics, that are not evident in a general first person shooter (FPS), but doesn't force you to complete the training sections before you can start to play against other people.

Takeaways: Think about the possibility of combining video demonstrations with the tutorial. Use a training match with an NPC character for familiarity before playing against actual people. Have a clear option to exit the tutorial.

4.1.6 Rocket League (Psyonix, 2015)

Genre: Sports / Soccer / Racing

Separate tutorial: Yes

Rocket League has a training option in the main menu, from which you can access the game's tutorial, which includes two sections: basic and advanced. The basic tutorials tells you the basic controls one after the other and requires you to complete different things before it proceeds to the next one. These include tasks like pushing the ball to the goal, jumping at the ball and scoring, making a powerslife turn etc. The tutorial map is a smaller version of the regular game map. All the possible controls are not accessible during the tutorial, only the required ones at any given time. The same principles apply to the advanced tutorial. You are given more advanced mechanics, but need to proceed with them in a similar way with limited controls to make sure you do things in the required way.

Takeaways: When teaching controls, consider only allowing the currently required controls to be available and pace the tutorial accordingly.

4.1.7 Guild Gear Xrd Revelator (Arc Systems Works, 2016)

Genre: Fighting

Separate tutorial: Yes

Guilty Gear Xrd Revelator is a 2D fighting game that has an extensive tutorial[1] teaching you many facets of the game. Starting with the movement controls, it shows you how to move, jump and dash. This is done by making the player pop balloons with his movements, and obstacles are presented that have to be jumped over and dashed through quickly enough. All of this makes sure that the correct controls are used in order to proceed in the tutorial. After movement training, the tutorial moves on to how to attack in different ways. The available buttons for attacking are presented on the screen the whole time, and different targets are given for the player to hit with specific attacks, with a section dedicated to each attack type (i.e. "attack all of these targets with the specified attack type"). A timer is also introduced later.

[1] https://www.youtube.com/watch?v=0oWBwcYr1LM

The specified attacks have to be made quickly enough in order to proceed with the tutorial. The target to hit is also specified with a symbol and letter specifying the attack to be used, which makes the object easy to understand at all times. The tutorial also teaches different types of basic combo attacks (a sequence of single basic attacks), and gives you targets to be beaten with given combos. This gives the game a very tactile feel from the start, giving practical applications to the controls instead of just giving you the controls and not showing how they can be applied and combined. This can feel like the tutorial is a little minigame in itself, apart from the actual intended gameplay. The third mechanic the tutorials introduces is blocking, i.e. how to block your opponents attacks. It also tells for what type of attacks each block type is effective (for standing, crouching and aerial blocks). NPC's attack the player, and can only be harmed with a proper counter attack after a succesful block. A mechanic specific to the Guilty Gear series, a *roman cancel*, is also taught in a similar fashion. After all this there's a recap of what has been learned (movement, attacks, blocking and roman cancels), and you need to apply all of the techniques to beat a number of NPCs combined with an obstacle course forcing the player to jump and dash in order to complete the section. Finally, there is one last fight against a more powerful character, and you get tips for how to proceed with the fight. This means e.g. using a specific type of mid range combo to build up the meter for using roman cancels for additional combo power. There's also a separate menu in the game from which you can access match-up tutorials. These show you how to best deal with specific characters and their unique abilities.

"Was that a lot to remember? Good, because a lot of servants are about to attack you. Defeat them all."

Takeaways: Use a detailed obstacle course to make sure all the required controls are used. Go beyond the basic controls and teach possible combinations and applications of the controls that will allow the player to start thinking about his own

style and variations.

4.1.8 Quake Champions (beta), (id Software, 2017)

Genre: FPS

Separate tutorial: Yes

When you first start the game, the first thing you see is an annoucement about where training videos are found in the menu. At the moment of writing this they are called beta tutorials, and the tutorials are video only, so the player needs to try to make use of the information in a separate setting. The first information given in the introduction video is not about controls, but rather about the spawning mechanic, weapons and types of power-ups. The first video of the three is a general introduction with all these concepts. The second video is about health and armor, and the third one about power-ups. All collectibles and their functionality is presented. It is assumed that the player has the required knowledge about the elementary controls. Information about game dynamics is also given, such as how it's important to start gathering power-ups and weapons when the match starts. Emphazising what to do, rather than how to do something, can be a good idea: "spawn, gear up, fight, restock".

Takeaways: Consider using a separate video that can be available online to demonstrate the basic principles of the game.

4.1.9 Dead Cells (early access), (Motion Twin, 2017)

Genre: 2D Platformer / Rogue-lite

Separate tutorial: No

Dead cells is a 2D platformer that throws the player straight into the game, and gives context-sensitive tips about how to move the character. Basic (Xbox) gamepad

knowledge is assumed on controlling the character with the left analog stick, but a text appears over the character to press 'A' on the controller to jump. Then X for main weapon, to break through a door. Then how to double jump with consecutive A-button presses. Progress is not possible if the instructions are not followed. Then the player gets a secondary weapon and the instructions to press 'Y' to use it to break through another door. The text doesn't disappear until the button is pressed. Then you get the promp to press 'B' to roll.

Takeaways: The tutorial does not have to be any longer than necessary, i.e. a very short tutorial is acceptable. Locking further progress without completing required actions is still an option, however, no matter how simple they are.

4.1.10 The Banner Saga (Stoic, 2014)

Genre: Turn-based strategy / RPG

Separate tutorial: No

The game jumps right into a combat tutorial after the first cutscenes. You are told to drag around the screen to see your surroundings, but it's not immediately evident how to track, i.e. by taking the mouse to the edge of the screen or by clicking and dragging. There's time to find out because the tutorial won't proceed until you click a button to continue. Next the initiative chart is explained, which shows the order in which the different characters will act. This also effectively means that The Banner Saga is a turn-based game. You're again asked to click to continue. All the important mechanics are explained and shown one after the other. Different actions and abilities are color-coded the further enhance their meaning. At the start of the next battle, you get the possibility to choose where to place your characters before the battle starts, but this is not mentioned right away, but in a later battle. There's question mark in the corner which you can click to get tips around the screen about what each item on the screen means. As new game modes or screens are introduced,

there is a text pop-up telling what the screen is and how to generally manage it. After the start of the game there is also a training mode accessible, where you can have training battles with your characers.

Takeaways: Use visual cues to help locate required actions in the user interface. Keywords and concepts can be highlighted or colorcoded in the tutorial text. Have a help menu accessible from the main interface.

4.1.11 Child of Light (Ubisoft Montreal, 2014)

Genre: 2D Platformer / Turn-based combat

Separate tutorial: No

Child of Light is a 2D platformer where you control an additional flying companion character on the screen with the right stick of the controller. The game has a checkbox in the options menu to turn off tutorials, but there's not really a separate tutorial. Basic controller experience is assumed, and during the first 90 minutes of gameplay, there are only a few new things presented. These include things like a pop-up that says how to move the companion character with the right stick, and how to press A to fly. There is a certain level of complexity in the controls because the player has to guide two characters on the screen simultaneously. In addition to the movement

Takeaways: Have an option to turn off in-game tutorials.

4.1.12 SWAT 4 (Irrational Games, 2005)

Genre: Tactical shooter

Separate tutorial: Yes

SWAT 4 has a separate training mode, called 'training', you can select from the

main menu. It is a shooting range that aims to simulate more realistic police training. It shows the player how to operate his firearms, interact with the environment, and most importantly how to command your squad of operatives. Different possibilities for this are quite extensive and are walk through in different parts in the training mode. There is a narrator who follows your progress and talks you through everything, while displaying necessary controls on the screen. Starting from the more simple firearm techniques on to the more complicated ways to command your squad and sniper, it builds in complexity at a steady but manageable rate.

Takeaways: In a more realistics setting consider copying a real world training scenario if there is one.

4.1.13 This War of Mine (11 bit studios, 2014)

Genre: 2D / Point & Click

Separate tutorial: No

There is no tutorial here, but the absence of it feels justified given the context and theme of the game. There are not really even any hints about the items or the mechanics of the game, but rather the player has to figure it all out as the game unfolds. The oppressive feeling of war ties into this uncertainty and what is or is not important adds to the general atmosphere in a somewhat justifiable way. Lack of knowledge as a dynamic, then, becomes an understandable design decision that can enhance the experience.

Takeaways: Lack of knowledge can be used as a game dynamic, i.e. not giving out information even if it was possible to do so.

4.1.14 Hitman (IO Interactive, 2016)

Genre: Action / Stealth

Separate tutorial: Yes

The game starts automatically with a tutorial mission, that walks the player through a simple mission that teaches the necessary mechanics and controls in order for the player to get a good grasp on available options. There are many different ways to complete a mission, but the controls required can be applied to different things in the same way. It feels like an introductory mission like this that essentially holds the player's hand is a good way to approach a game like this, that has more things to do than just moving and shooting. Necessary controls are presented on the screen as well as in narration, and sometimes in a proper context the game pauses and tell you to press a button to active and action, e.g. to subdue a person. In a simplified form, the player essentially gets told to go here, press this button, then go there and press this button and so on, so it becomes clear what kind of actions are available.

Takeaways: Go through a whole single mission applying other heuristic principles to show how the game is played step by step.

4.1.15 Binary Domain (Sega, 2012)

Genre: Third-person shooter

Separate tutorial: No

There is not a separate tutorial section. The tutorial is embedded in the regular gameplay. In the start of the game the player is asked if they want to view the basic controls of shooting and moving. Movement of the player is limited during this section, as individual controls are viewed one after another. Information of which buttons to press is presented on the screen and in narration. Context-sensitive

information comes up for the remainder of the game whenever the player has the option to take cover, pick up items etc. This is presented on the screen with an icon of which button to play and a visual representation of the action.

Takeaways: Use context-sensitive information throughout the game to show when certain actions are possible, e.g. being able to take cover behind an obstacle. Ask the player whether they want to view basic controls or not.

4.1.16 Minecraft: Pocket Edition (Android) (Mojang, 2016)

Genre: Sandbox / Survial

Separate tutorial: No

The lack of a tutorial or guidance makes this a difficult game to approach. Although basic movement controls are on the screen constantly, emulating gamepad controls, it is difficult to find out how to do other things, such as craft equipment. Even so much that it is necessary to search for information online on how to approach the game.

Takeaways: It is important to consider showing basic information about the game, no matter how simple it may feel to the developer or designer.

4.1.17 Monument Valley (Android) (Ustwo Games, 2014)

Genre: Puzzle

Separate tutorial: No

The first puzzle of the game introduces the controls to the player. There are only two of them, but they are presented clearly in text on the screen. The player can either move the character by tapping the screen, or change the form of the play area by tapping and dragging a designated area. There may not be much to

do mechanics-wise, but the information still feels very helpful and necessary, and doesn't take a lot of time to absorb.

Takeaways: Even simple mechanics should be presented in a clear and noticeable manner.

4.1.18 New Star Soccer (Android) (New Star Games, 2012)

Genre: Sports

Separate tutorial: No

New Star Soccer essentially consists of a few minigames that combine into a football manager game, but the most playing is done in actual soccer matches. There is a separate practice mode that tells you how to kick the ball to first time you play it, and also during regular gameplay there are pop-ups when actions are tried for the first time.

Takeaways: Include a practise mode for basic mechanics if they are difficult enough to master. Context-sensitive help can also come up after, not before, the player takes actions, so they can discover things at their own pace.

4.1.19 Insurgency (New World Interactive, 2014)

Genre: Tactical shooter

Separate tutorial: Yes

The tutorial of Insurgency follows a similar path as the tutorial of SWAT 4. There is an optional training facility that the player can go to, and he or she then is guided through a shooting range and an obstacle course with an instructor talking to the player about what to do next and how. Visual cues on controls are also presented and actions are sequenced so that the tutorial will not proceed until the player does the required action.

Takeaways: Even in a realistic setting, using visual cues in the environment can be justified in a tutorial to enhance the speed of learning.

4.1.20 Hammerwatch (Crackshell, 2013)

Genre: Hack and slash / Dungeon crawler

Separate tutorial: No

There is not a tutorial as such, but rather buttons with question marks littered around the map. When the player activates these buttons, tips are presented in text on the screen. However, they refer to the more high-level mechanics such as what happens when the player goes to another floor or how vendors work. No help on e.g. the controls is presented; it seems to be assumed that players are familiar with the basic control scheme of a game such as this one.

Takeaways: The tutorial can also be about what the character can or should do, not how.

Chapter 5

Analyzing expert review results

How to combine our experience with the games in the previous chapter with the heuristics we looked at earlier? Since there are games from various genres it is justified to have a somewhat abstract take on how they are or should be presented. It also depends a lot on the context, since principles used in a good fighting game tutorial do not necessary apply to a game of another genre that well. This means that when thinking about how to build a good tutorial, you only need some tools from the toolbox, but a versatile toolbox is still a good thing to have. If there is no hammer, it does not mean that nails are not a good thing. This is also partly the case with Minecraft: Pocket Edition. If there was a tutorial, the game might turn out to be easy and fun to play, but the lack of one demonstrates that good and usable things are not necessarily intuitive at first [Raskin, 1994]. This is the reason it is important to teach the player, no matter how simple the controls would be. One related example is text and programming editors such as Vim. They are not intuitive to use and require a long process to learn and modify to your own purposes, but eventually large boosts in productivity may be experienced [Robbins et al., 2008]. However, games are in most cases a voluntary pursuit for the player, not a tool, and are works of imagination and art to be enjoyed mostly as entertainment, so making a one size fits all collection of heuristics is rather impossible. It is important to have

tools to analyze tutorials and games, but games that in itself feel like a chore to the player rarely succeed. One central thing emerges from the concept of a tutorial: it does not generally matter if you have a separate tutorial, or a sort of in-game help and instruction that guides the player in to the game. This can even happen for a very long period of time, as a game gradually introduces new concepts as the characters develop and e.g. gain new abilities for hours into the game, up until the very last levels even. It is not always just "learn everything and start playing". The game design choice of having mechanics come up later in the game means that it is not necessarily a good idea to teach all those mechanics beforehand in a tutorial at the very beginning of the game. Having small *separate* tutorials along the game could be distracting in the long run, so it is better to embed them within the actual game as new properties are unlocked. The tutorial, then, becomes a little ambiguous as a concept. A game might instruct the player really well but not have a separate tutorial to do that, or the game might have a separate tutorial and also instruct the player really well in that as well. It is safe to say that when we are talking about a tutorial, we can talk about the way the game attempts to transfer the required knowledge to the player, as we defined tutorials in an earlier chapter. A common nominator is that the tutorial should begin with things of lower complexity and then move on to higher level concepts. This is also called *priority learning* [Bycer, 2016]. With these results in mind, we will go through the game usability heuristics found in the literature and discussed before, and attempt to enhance our view on how video game tutorials can be analyzed regarding their usability based on the games we discussed.

5.1 Tutorials and existing heuristics

1. Easy to learn, hard to master. How can we say that a tutorial should be easy to learn, but hard to master? In the case of games like Dark Souls 2,

the presence of a tutorial might not be completely evident, but it is still not a very difficult tutorial. If anything, if there is a tutorial, the whole point of it is to transfer information. This means that if there is one, it should do its job well, or not exist at all, like with This War of Mine. The lack of a tutorial in Minecraft: Pocket Edition is different, because there is nothing on the screen that tells us that there are possible actions, unlike in This War of Mine. Using this heuristic for a tutorial, we can just say that the game—or rather the tutorial—should be *easy to learn*.

2. The goals are clear. This heuristic applies to tutorials as well. They should be clear about what they aim to teach and what the player's next action should if applicable.

3. The skills needed to attain goals are taught early enough to play or use later, or right before the new skill is needed. When the game design is such that all of the possible actions are not immediately accessible, like in Hori and the Blind Forest, it is a good idea give that information to the player only right before the new mechanic becomes available to the player.

4. The first ten minutes of play and player actions are painfully obvious and should result in immediate and positive feedback for all types of players. This feels like something that is present in all of the games except Minecraft: Pocket Edition, which shows that this lack of information can have a very detrimental effect on learning the game even with a seemingly simple game.

5. Player does not need to read the manual or documentation to play. What is documentation? If there is a tutorial that has text in it, even if presented in small parts, doesn't it essentially mean that the player is reading the documentation? Accessing a separate document or file, however, is not necessarily desirable. Also, there are complicated genres like flight simulators where a huge documentation is a

necessity. Having all that information presented in-game in a fluid manner would still be a good thing, so this heuristic is a good goal in that sense, but not always practical.

6. Player does not need to access the tutorial in order to play. The ability to skip a tutorial is an important one, just design it so that someone playing the game for the first time does not do accidentally.

7. Status score indicators are seamless, obvious, available and do not interfere with game play. This becomes meaningful in e.g. sequenced tutorials where the next action is not available until the current one has been completed, so it is important to communicate that necessity properly.

8. The game provides feedback and reacts in a consistent, immediate, challenging and exciting way to the player's actions. Like in the previous point, making clear that the player just did the correct required action is important. The Guilty Gear Xrd Revelator tutorial is a great example of this.

9. Provide appropriate audio/visual/visceral feedback (music, sound effects, controller vibration. This, again, adds to the previous point in question. It should just be considered that there are multiple ways to communicate a success simultaneously, i.e. not just in text but also in color, sound and e.g. controller vibration.

10. The player is given controls that are basic enough to be learned quickly, yet expendable for advanced options for advanced players. This is very true with a game like Guilty Gear Xrd Revelator. The fact that even the tutorial in the game goes in that depth to demonstrate advanced combinations extended from the basic controls makes this an important heuristic to have.

11. Player interruption is supported, so that players can easily turn the game on and off and be able to save the game in different states. Insurgency has a moderately long tutorial section, and it had a tendency to crash a number times while playing it. This meant that the tutorial had to be started again from the beginning which was frustrating. Adding a save game option to be used from a separate tutorial context is thus a good idea.

12. Upon turning on the game, the player has enough information to begin play. This is self-evident in a way. If there is a separate tutorial, it is a good idea to have it accessible from the same menu as the main game, so that the option is clear.

13. Players should be given context-sensitive help while playing so that they are not stuck and need to rely on a manual for help. In a tutorial it should always be clear what the next expected player action is.

14. All levels of players are able to play and get involved quickly and easily with tutorials and/or progressive or adjustable difficulty levels. Here this essentially means that the tutorial should be easily accessible if there is a separate one, or that the in-game tutorial should be good enough according to the other heuristics. In general, it should not take too long for player to be able to actually do something in the game, which is mostly likely some part of the tutorial.

15. Get the player involved quickly and easily. This is practically same as the previous one. With the core idea being the tutorial, it does not need to be separately mentioned in the later heuristics. Also, as the heuristics regard tutorials, it can be assumed that the tutorial is already in progress so how quickly the player gets to the tutorial is not in essence the problem.

16. The game should give hints, but not too many. For progress in the actual game this is likely a good idea, but a tutorial should not give hints. It should be as clear as possible, not "maybe you need to press A to jump, maybe B, who knows...". This might still be enjoyable if humor was the intent, and rules are meant to be broken anyway.

17. Allow users to skip non-playable and frequently repeated content. This is very important for a tutorial too, just so that it is not done accidentally.

18. Provide instructions, training and help. These have all been mentioned earlier, and are in a way the essence of tutorials. Training, like it is in New Star Soccer, is a great way to enhance the gaming experience in addition to some other basic information that would be considered a tutorial. To also be able to practise on your own, not in a guided setting, can be beneficial.

5.2 Heuristics for tutorials

Now that we have distilled our experience with the games in question through the heuristics used with general game design and usability found from the literature, it is time to think about heuristics for video game tutorials specifically. One difficult thing with this is the wording of heuristics. As discussed earlier, not all heuristics need apply in order for the tutorial to be efficient. It also depends a lot on the genre of the game for example. There is a tendency with heuristics to have a sort of a "you should do this" type of tone, whereas in practise, they are only things to be considered that might fit the game's needs or not. We need to think about how to present the information so that everything does not come across as a necessity, but rather a point to consider. Not all games necessarily need a training mode for example, so it can be problematic to say "have a training mode". Using the takeaways from the experiences with the games earlier and the general heuristics

discussed in the context of tutorials, we can combine them to create a list of heuristics that is more applicable to the usability of video game tutorials specifically. We were able to find and create a total of 27 heuristics, which are as follows:

1. The tutorial should reflect the intended gameplay experience.

2. The tutorial should reflect the pacing of the gameplay and not introduce mechanics before they are accessible in-game.

3. There can be multiple tutorials for different aspects of the gameplay.

4. Controls should be taught interactively when possible.

5. Learning through example is a possibility.

6. Using a training opponent before the actual game is a possibility.

7. Sometimes it is good to limit the possible controls to the ones being currently taught and pace the tutorial accordingly.

8. Teach applications and combinations of the basic controls that go beyond the basic control scheme.

9. Using a separate video to describe the usual game elements and dynamics is a possibility.

10. A short tutorial for a complex game can still be good.

11. Use different visual and audible ways to present information and color code keywords.

12. Have an option to turn off in-game tutorials.

13. Real-world scenarios can be copied to mimic training sessions in applicable genres.

14. Consider not showing all possible information depending on the setting, lack of knowledge can serve as a game dynamic.

15. Completing a whole mission step by step in an applicable genre can be helpful.

16. Use context-sensitive information throughout the game.

17. Do not underestimate the importance of a tutorial even in a simple game.

18. Present simple things in a clear and noticeable manner.

19. Consider a separate practise or free mode that can be used at the player's own pace.

20. Realism is not an excuse to not have additional visual elements in a tutorial setting.

21. The tutorial can also be about what the character can or should do, not (just) how.

22. The tutorial should not be made unnecessarily difficult.

23. The goals of the tutorial should be clear at all times.

24. The tutorial should begin with the most common interaction in the game. n

25. The player should be able to skip the tutorial but not by accident.

26. The tutorial should give feedback of the player's required actions with visual and audible elements, and controller vibration if possible.

27. The player should be able to save the game during the tutorial.

5.3 How to use the heuristics

As said, these heuristics are not to be treated as a list of elements that all need to apply to a single tutorial. If such a tutorial exists where this is possible, good. Games come in so many forms that it can not nevertheless be expected, and is not intended as a specifier for a good tutorial. What matters is that when specific heuristics can be taken into account, it might generally be a good idea to do so. 27 is a relatively large number of heuristics here, but variations in genres and mechanics can already limit that number considerably. Since there can also be a link between game genre preferences and personality types, it might also not be a good idea to artifically force cross-genre tutorial conventions, at least if player retention is a goal [Peever et al., 2012]. This means that unfamiliar genres might drive some people away, which goes back to our heuristic number one, that the tutorial should reflect the intended gameplay experience. What matters is understanding what is necessary for the game and what it is trying to convey, and even what the target audience is like, if there is such a thing in question.

Chapter 6

Conclusion

Starting with discussing what usability is, we have come to a list of heuristics that can help us evaluate the usability of video game tutorials. Along the way a thought arises that usability seems generally more universal as a concept than the usability of video game tutorials, meaning that the usability of a tutorial can not necessarily be generalized in the same way as easily as the general principles for the usability of an interface may suggest, which is what a lot of traditional usability literature and research seems to be concerned with. There is a lot more going on in video game usability than just the interface. Games come in different forms and genres, and usability therein—especially with tutorials—has a lot more layers than just the interface of the game. The information is more dynamic and in a state of change. Tutorials generally aim to convey information about how the game is actually played, and designing an interface that would alone fill that requirement is rarely, if ever, satisfactory. We can also see a divide between productivity software and video games, which are another type of software. A game is usually a contained system that we engage with voluntarily. Productivity software involving e.g. word processors and database management are necessary tools for certain fields of work, and can be connected to a larger whole. This means e.g. modifying a database which causes boxes to be moved in a real world warehouse and shipped

to another country. It can not always be predicted what type of work will the end users be doing, which is perhaps a reason tutorials are not a thing with productivity software. They are also usually more complex than games and come with manuals and extensive documentation, which is not desirable with games, as heuristics in previous studies suggest. Video games, however, know themselves inside out, and tutorials can be designed and tailored to the whole experience and the possibilities it offers within a single game. This is not always the case in practise, and heuristics for tutorial design can be a welcome addition to the development process of the game. These usability heuristics are essentially rules of thumb derived from testing in the literature combined with the author's personal experience with a number of chosen games from different genres, as listed in chapter four. The heuristics are not meant to be applied all at one to a single game. The designer needs to use their own judgement whether a given heuristic is applicable to their game or not, or whether any of them are.

The list of games we have looked at here contains titles from a number of different genres. This is because the idea has been to try and recognize improvements on a larger scale, not within a single genre. No two games are generally alike, so having a set of heuristics that can be used with designing a game of any genre or style makes the result applicable on a wider scale. Also, any single heuristic is not bound to a single genre. Their usefulness can be realized in any context by going through the list and seeing if it contains an applicable idea for the game or tutorial that is being developed. It can be useful because different games can borrow things from different genres and mix things up, so the divide between game styles does not have to be set in stone or defined in an extremely accurate way.

Judging from the success of certain recent years' hit games such as the Dark Souls series—one of which we analyzed here—it can also be argued that if you set out to make a game based on certain recognized industry standards and principles and an enjoyable experience in mind, you might have already taken your first wrong

turn. Not all people like difficult and punishing games with death of the player as an integral mechanic, and where things are not explained to you in a hand-holding manner, but such games can still become recognized classics. Different target groups can want two opposite things, so it's difficult to generalize, and in the game development process that is a whole another thing to take into account, i.e. who are you making your game for, if that even is a question.

Bibliography

[Adams, 2013] Adams, E. (2013). *Fundamentals of Game Design*. New Riders, Berkeley, CA, 3 edition edition.

[Almeida et al., 2010] Almeida, S., Mealha, Ó., and Veloso, A. (2010). Heuristic Evaluation of 'FarmVille'. *Videojogos 2011 - 3rd Annual Conference in the Science and Art of Video Games*, (1990):21–30.

[Andersen et al., 2012] Andersen, E., O'Rourke, E., Liu, Y.-E., Snider, R., Lowdermilk, J., Truong, D., Cooper, S., and Popovic, Z. (2012). The impact of tutorials on games of varying complexity. *SIGCHI Conference on Human Factors in Computing Systems*, (Pajitnov 1984):59–68.

[Bauckhage et al., 2012] Bauckhage, C., Kersting, K., Sifa, R., Thurau, C., Drachen, A., and Canossa, A. (2012). How players lose interest in playing a game: An empirical study based on distributions of total playing times. *2012 IEEE Conference on Computational Intelligence and Games, CIG 2012*, pages 139–146.

[Bernhaupt et al., 2007] Bernhaupt, R., Eckschlager, M., and Tscheligi, M. (2007). Methods for evaluating games: how to measure usability and user experience in games? *Proceedings of the international conference on Advances in computer entertainment technology*, pages 309–310.

[Bevan, 2006] Bevan, N. (2006). International Standards for HCI. *Encyclopedia of Human Computer Interaction*, 55(May):1–15.

[Brignull, 2013] Brignull, H. (2013). Dark Patterns: inside the interfaces designed to trick you http://www.theverge.com/2013/8/29/4640308/dark-patterns-inside-the-interfaces-designed-to-trick-you 2017-8-17.

[Bycer, 2016] Bycer, J. (2016). How to Improve Education via Game Tutorials http://ubm.io/2w7QTpj 2017-8-18 Gamasutra 2017-8-18.

[Carter et al., 2014] Carter, M., Downs, J., Nansen, B., Harrop, M., and Gibbs, M. (2014). Paradigms of games research in HCI. *Proceedings of the first ACM SIGCHI annual symposium on Computer-human interaction in play - CHI PLAY '14*, pages 27–36.

[Carvajal et al., 2013] Carvajal, L., Moreno, A. M., Sánchez-Segura, M. I., and Seffah, A. (2013). Usability through software design. *IEEE Transactions on Software Engineering*, 39(11):1582–1596.

[Cavallin et al., 2007] Cavallin, H., Martin, W. M., and Heylighen, A. (2007). How relative absolute can be: SUMI and the impact of the nature of the task in measuring perceived software usability. *AI and Society*, 22(2):227–235.

[Chen et al., 2015] Chen, Y.-H., Rorissa, A., and Germain, C. A. (2015). Usability Definitions in a Dynamically Changing Information Environment. *portal: Libraries and the Academy*, 15(4):601–621.

[Cheung et al., 2014] Cheung, G. K., Zimmermann, T., and Nagappan, N. (2014). The First Hour Experience: How the Initial Play Can Engage (or Lose) New Players. *Proceedings of the first ACM SIGCHI annual symposium on Computer-human interaction in play - CHI PLAY '14*, pages 57–66.

[Desurvire et al., 2004] Desurvire, H., Caplan, M., and Toth, J. a. (2004). Using heuristics to evaluate the playability of games. *CHI'04 extended abstracts on Human factors in computing systems*, pages 1509–1512.

[Desurvire and Wiberg, 2009] Desurvire, H. and Wiberg, C. (2009). Game usability heuristics (PLAY) for evaluating and designing better games: The next iteration. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 5621 LNCS, pages 557–566.

[Dubey et al., 2012] Dubey, S., Gulati, A., and Rana, A. (2012). Integrated Model for Software Usability. *International Journal on Computer Science and Engineering*, 4(03):429–437.

[Febretti and Garzotto, 2009] Febretti, A. and Garzotto, F. (2009). Usability, playability, and long-term engagement in computer games. In *Conference on Human Factors in Computing Systems - Proceedings*, pages 4063–4068.

[Federoff, 2002] Federoff, M. a. (2002). Heuristics and usability guidelines for the creation and evaluation of fun in video games. *FUN in Video Games Thesis University Graduate School of Indiana University Dec*, Master of(December):52.

[Ferré et al., 2001] Ferré, X., Juristo, N., Windl, H., and Constantine, L. (2001). Usability basics for software developers. *IEEE Software*, 18(1):22–29.

[Holzinger, 2005] Holzinger, A. (2005). Usability {Engineering} {Methods} for {Software} {Developers}. *Commun. ACM*, 48(1):71–74.

[Isbister and Schaffer, 2008] Isbister, K. and Schaffer, N. (2008). *Game Usability*.

[ISO, 1998] ISO (1998). ISO 9241-11.

[Kahneman et al., 1982] Kahneman, D., Slovic, P., and Tversky, A. (1982). Judgment under uncertainty.

[Kjeldskov et al., 2004] Kjeldskov, J., Skov, M. B., and Stage, J. (2004). Instant data analysis: conducting usability evaluations in a day. *Proceedings of the third Nordic conference on Human-computer interaction*, pages 233–240.

[Livingston et al., 2010] Livingston, I. J., Mandryk, R. L., and Stanley, K. G. (2010). Critic-Proofing : How Using Critic Reviews and Game Genres can Refine Heuristic Evaluations. *Evaluation*, pages 48–55.

[Merriam-Webster, 2017a] Merriam-Webster (2017a). https://www.merriam-webster.com/dictionary/heuristic.

[Merriam-Webster, 2017b] Merriam-Webster (2017b). Merriam-Webster. https://www.merriam-webster.com/dictionary/tutorial.

[Moss, 2017] Moss, R. (2017). 7 introductory levels that all game developers should study http://ubm.io/2i8NQqw Gamasutra 2017-8-18.

[Nielsen, 1993] Nielsen, J. (1993). *Usability Engineering*, volume 44.

[Nielsen, 2012] Nielsen, J. (2012). Usability 101: Introduction to Usability. *Nielsen Norman Group*.

[Nielsen and Molich, 1990] Nielsen, J. and Molich, R. (1990). Heuristic Evaluation of user interfaces. *CHI '90 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, (April):249–256.

[Norman, 2013] Norman, D. A. (2013). *The Design of Everyday Things*. Basic books.

[Peever et al., 2012] Peever, N., Johnson, D., and Gardner, J. (2012). Personality & video game genre preferences. *Proceedings of The 8th Australasian Conference on Interactive Entertainment Playing the System - IE '12*, pages 1–3.

[Pinelle et al., 2008a] Pinelle, D., Wong, N., and Stach, T. (2008a). Heuristic evaluation for games: usability principles for video game design. *Proceedings of SIGCHI Conference on Human Factors in Computing Systems*, pages 1453–1462.

[Pinelle et al., 2008b] Pinelle, D., Wong, N., and Stach, T. (2008b). Using genres to customize usability evaluations of video games. In *Proceedings of the 2008 Conference on Future Play Research, Play, Share - Future Play '08*, page 129.

[Raskin, 1994] Raskin, J. (1994). Intuitive equals familiar. *Communications of the ACM*, 37(9):17–19.

[Robbins et al., 2008] Robbins, A., Hannab, E., and Lamb, L. (2008). *Learning Vi And Vim Editors 7th Ed - Jul.2008.*

[Schaffer, 2007] Schaffer, N. (2007). Heuristics for usability in games. Technical report.

[Tarkkanen et al., 2013] Tarkkanen, K., Reijonen, P., Tétard, F., and Harkke, V. (2013). *Back to User-Centered Usability Testing*, volume 7946.

Appendix A

General heuristics from the bibliography

1.	Easy to learn, hard to master.
2.	The goals are clear.
3.	The skills needed to attain goals are taught early enough to play or use later, or right before the new skill is needed.
4.	The first ten minutes of play and player actions are painfully obvious and should result in immediate and positive feedback for all types of players.
5.	Player does not need to read the manual or documentation to play.
6.	Player does not need to access the tutorial in order to play.
7.	Status score indicators are seamless, obvious, available and do not interfere with game play.
8.	Game provides feedback and reacts in a consistent, immediate, challenging and exciting way to the players actions.
9.	Provide appropriate audio/visual/visceral feedback (music, sound effects, controller vibration).

10.	Player is given controls that are basic enough to learn quickly, yet expandable for advanced options for advanced players.
11.	Player interruption is supported, so that players can easily turn the game on and off and be able to save the games in different states.
12.	Upon turning on the game, the player has enough information to begin play.
13.	Players should be given context sensitive help while playing so that they are not stuck and need to rely on a manual for help.
14.	All levels of players are able to play and get involved quickly and easily with tutorials, and/or progressive or adjustable difficulty levels.
15.	Get the player involved quickly and easily.
16.	The game should give hints, but not too many.
17.	Allow users to skip non-playable and frequently repeated content.
18.	Provide instructions, training, and help.

Appendix B

Heuristics for tutorials