Usability of video game tutorials: an expert review

Tuomas Yli-Kovero

Master's Thesis

University of Turku

Department of Information Technology

2017

Abstract

UNIVERSITY OF TURKU

Department of Information Technology

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. This is important also for a type of software called video games. 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 an expert review of selected video game tutorials, 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). 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: "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 next chapter. 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.

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't 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 comes 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 to 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).

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

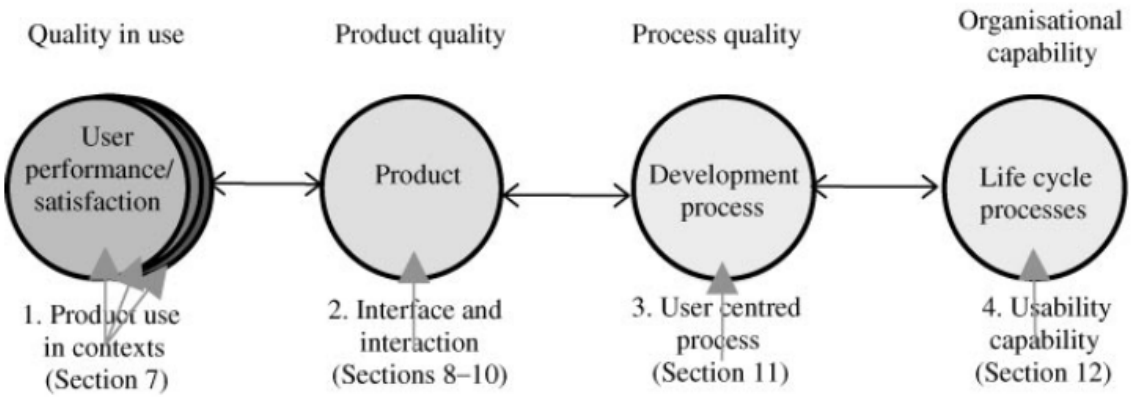

Figure 2.1: The logical relationships of standards related to usability [Bevan, 2006]

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.

2.1.2 Usability in video games

In the context of usability, the term user satisfaction comes up often. The foremen-tioned *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 type of software – it might be important to think about how to achieve player satisfaction first. 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 doesn't sound like 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

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 we can base our evaluation on and then proceed to go through a number of tutorials and see how well they fit within the usability guidelines defined in the heuristics we are going to form. But what is a heuristic? The Merriam-Webster dictionary defines the word heuristic as follows [Merriam-Webster, 2017]:

- 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.

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. 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 [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 don't 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 used in part static screen dumps and old voice response systems using telephone buttons. Those systems are not exactly similar to modern video games, so we can't yet say with certainty how a heuristic approach might work for video game tutorials specifically, even if it hasn't 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

Tutorial can be interpreted as the transferring of knowledge. 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 into many hours in the game, as new mechanics or strategies are introduced.

Chapter 3

Methods for expert review

Previously we have talked about what heuristics are and how they relate to usability testing. 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. We also have to form a selection of games that will be the target of our evaluation.

3.1 Building 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. 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. 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], [Federoff, 2002], [Pinelle et al., 2008] 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.

No.	Heuristic and description
1.	There is an interesting and absorbing tutorial that mimics game play.
2.	The first player action is painfully obvious and should result in immediate positive feedback.
3.	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.
4.	Tutorial is repeatable.
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 user's 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.

Figure 3.1: Tutorial-specific heuristics compiled from Desurvire et. al [2008].

The heuristics in Table 3.1 are from Heuristic Evaluation of Playability (HEP) [Desurvire et al., 2004]. The original HEP contains 41 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.

3.2 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.

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, 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 have to be constantly paused and wouldn't feel like the actual gameplay anyway. On the other hand, a more complex 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.

4.1 Dark Souls 2

There's no separate tutorial section. The player can read stone tablets he comes across, which show the controls separately, one by one. First which button to press to attack. Then target lock. Then you get to kill two enemies. Then how to dash. Then you kill an enemy. Then how if you attack from behind enemy's back it's a critical hit. Then how to roll. Then kill two enemies again. Then how to switch weapons. Then how to backstep. Then kill an enemy. Then how to use an item. Then how you move the camera (RS). Then you can enter another area that gives more stone tablets on the floor. How to wield your weapons with both hands. Then kill to enemies. Then how to perform a dashing jump, and you actually can jump over a chasm and pick an item. Then you can climb up a ladder (if you notice it) and it tells you how to parry, perform a plunging attack (dropping on an enemy) and then how to guard break.

4.2 Ori and the Blind Forest

No separate tutorial. As the player moves, informs to hold A longer to jump higher. Then to hold down and press A to 'jump down through platforms'. Let's the player know how collectibles work as they are found. Then how to hold B to save your game (saving is an in-game mechanic, not 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 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.

4.3 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. Then you get the overview of the character you chose, i.e. what kind of abilities she has. Then you get a "Tip Panel" on the top left of your screen which has the basic controls, and are told to familiarize yourself with them. Then you get a short cinematic about how you gain experience (xp) when killing enemy minions and are shown what some basic info on the screen means. Then you get 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). Then you're told about 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. Then your 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.

4.4 Psychonauts

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. 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 this constant information is given 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.

4.5 Quake Live

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.

4.6 Rocket League

Rocket League has a training option in the main menu, from which you can access the game's tutorial, which includes to 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.

4.7 Guild Gear Xrd Revelator

https://www.youtube.com/watch?v=0oWBwcYr1LM Guilty Gear Xrd Revelator is a 2D fighting game that has an extensive tutorial 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. 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. This can feel like the tu:qtorial 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 to 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."

4.8 Quake Champions (beta)

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".

4.9 Dead Cells

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.

4.10 The Banner Saga

The game jumps right into a combat tutorial after the first cutscenes. Your 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.

4.11 Child of Light

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 a certain level of complexity in the controls because the player has to guide to characters on the screen simultaneously. In addition to the movement

4.12 SWAT 4

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.

Chapter 5

Analyzing expert review results

Here we are looking at the results and finding answer to the question of usability in video game tutorials.

Chapter 6

Conclusion

Who knows...

Judging from the success of certain recent years' hit games such as the Dark Souls series, it can also be argued that if you set out to make a game based on certain recognized industry standards and an enjoyable experience in mind, you might have already taken your first wrong turn. Different target groups can want two opposite things, so it's difficult to generalize.

Bibliography

[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.

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

[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.

[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.

[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.

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

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

[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.

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