

**HOCHSCHULE FÜR ANGEWANDTE  
WISSENSCHAFTEN HAMBURG**  
Hamburg University of Applied Sciences

**Are meaningful choices for combat systems  
more relevant than polished gameplay?**

---

**Master Thesis  
by Andrej Wiehagen**

<b>1. Introduction - Basics of combat design</b>	
1.1 Analysis on constraints for a combat system in games.	2-3
1.2 Analysis on the purpose of combat systems in video games.	4-5
1.3 Approach to differentiate combat systems	6-8
1.4 The elements of combat systems	8
1.4.1 Locomotion for melee combat systems	9-11
1.4.2 Offensive option in combat systems	11-13
1.4.3 Defensive option in combat systems	13-14
1.4.4 Combat choreography	15-16
<b>2. How does meaningful choice and polished gameplay impact game feel in games</b>	
2.1 An approach to define game feel	17-20
2.2 What is game polish?	20-25
2.3 What are meaningful choices in games?	25-33
2.4 Why do meaningful choices and polished gameplay not go together	33-34
<b>3. Meaningful choices and polished gameplay in different areas of combat design</b>	
3.1 Combat design: Enemy and player character	35-36
3.1.1 Enemy design: Design a puzzle for the player to solve	36-40
3.1.2 Player character abilities: Give the player the right tools	40-41
3.1.2.1 Properties of player character abilities	41-42
3.1.2.2 How abilities give meaningful choices	42-44
3.1.2.3 Why abilities need polish	45-47
3.2 Analysis of polished gameplay and meaningful choice design on the example of Diablo III and Path of Exile	47-51
3.3 Using the principles of polishing game elements and crafting meaningful choices to design a graphical user interface and balancing content in combat systems	51
3.3.1 Polish - Display Information: Gui & Feedback	52-56
3.3.2 Meaningful choice by balancing	56-59
3.3.3 Balancing the math with the user experience in mind when buffing or nerfing abilities	59-60
<b>4.0 Conclusion</b>	
Are meaningful choices for combat systems more relevant than polished gameplay?	60
<b>Attachment</b>	
Bibliography	61-62
Game reference	62
Illustration dictionary	63-66

## **1.**

### **Introduction - Basics of combat design**

Combat is not an element that exists in every game on the market but it's undeniably a large part of video games as we know them. From punching, kicking, blocking, shooting to blowing things up or sneaking up from behind a target. Different combat systems reward the player with level increases, progression in story line, new weapon or skills, gold or a higher rank in player vs player games. But, what is often overlooked is the reward for the player within the combat itself.

This scientific paper is an analysis of game mechanics that make combat feel fun and rewarding for the player from an emotional perspective. By analyzing different combat systems this paper attempts to define the anatomy of satisfying combat.

#### **1.1**

##### **Analysis on constraints for a combat system in games**

To design a combat system the designer has to start somewhere. Let's assume the game designers want to construct an enemy first. It is important to not design in a vacuum but to make constraints for the system first. To make an example: Mark Acero talked about the enemy design of a dragon in a presentation at Game Design Expo 2012<sup>1</sup>.

The designer decides the said dragon should be massive compared to the player. Now the wings need to be enormously large to physically support the weight while flying to be a realistic entity. Further taking a look at dinosaurs the designer concludes that the tail needs to be large in size just to support the weight of the dragon.

---

<sup>1</sup> Acero, Mark:  
„Presentation: Combat Design“

The next step is to tell the design team about this enemy design. But with the heavy weight, the amount of bones alone the dragon would need to have to support the wanted epic wing movement the designers just bursted the budget for the enemy design. The enemy might be a very strong design itself but developing in a nutshell comes with a cost. Without considering constraints the designer are flying blind.

Let's assume the dragon would need 5 weeks in the first iteration before deciding that the dragon will be fought in a dungeon, and never uses it wings. A lot of the work would be done on the wrong elements. Knowing about the constraints can save a lot of money. A flame breath might look very awesome in theory but not when it lowers the game to three frames per second in performance because the systems were never built for it beforehand.

Knowing about the technological constraints is essential for a productive design flow.

The more contains the designer has worked out before start developing, the more time can be spend on the main elements of the design. Considering when the enemy will be fought, where, and how, as well as knowing about the experience the players should have helps focusing the design process.

Let's assume time passed and the dragon has gone through 5 iterations. The fight looks amazing. The attacks work as intended and the cinematics are ready to put in the actual game. But there is no guarantee that the fight will actually be fun for the player. It might be to difficult, not visually pleasing, or there are not enough meaningful choices for the player to choose from. There will never be a guarantee for a design to work. Iterations and testing is needed to make a good design. The more iterations the design has, the better the chance for the design to be a success. Knowing as much as possible about the environment of the design as well as considering the player's expectations can give the designers a better idea of the scoap for the projekt.

## 1.2

### Analysis on the purpose of combat systems in video games.

One of the constraints the designer should have in mind is the purpose of the combat inside the game. There are many different combat systems, but whatever system the designer chooses for the game there can be many different roles for the combat.

Taking a look at the combat system for »Final Fantasy VII«<sup>2</sup> as example: Final Fantasy games are mainly based on telling a story interactively. One of the strengths of Final Fantasy VII design is the pacing between story and combat. The story is without a doubt the core of final fantasy VII.

So what purpose does the combat for final fantasy VII has. The players fight enemies and bosses that are story relevant. The fight against a giant machine that had previously hunted the protagonists and holding up against their special attacks give the player the feeling of actually having an impact on the outcome of the game. So the combat is an interface for the player to interact with the story of the game.



Fig. 1: Final Fantasy VII's turn based combat system

---

<sup>2</sup> Final Fantasy VII:

Square Enix; Square Play Station; Release: 31. Januar 1997



Fig. 2: Dark Souls III - Player fights enemies in an action based combat system

While the battles in Final Fantasy VII occur randomly, players have the choice of meeting an enemy in Dark Souls unless ambushed. When sneaking around an encounter while having an exit strategy, or just running in blind, the decision tells a story in itself.

The story in Dark Souls is not told directly. It is experienced while wandering through the game. From hidden traces on the ground, to the environment, to the enemies and their appearance. The fight is not just an interface to interact with a linear story. It tells a story. Players must learn how each enemy behaves and how it reacts in order to dodge, block and counter to beat the game. This process gives the player an appealing feeling. It is less important to the player how strong the character is, but their performance in a particular encounter makes the player feel good.

Knowing why the battle exists in the game is a caveat that can help determine the purpose of the game's combat, and hence understands the limitations that developers should consider when designing combat elements.

### 1.3

#### Approach to differentiate combat systems

Combat is an extensive topic and can mean many different things, depending on who you ask or what game you work on. The most common combat systems are designed for either action-based games, turn-based games, or real-time strategy games.

The category of action-based fighting is the most widespread one. Ego Shooter, Combat Brawler, Hack 'n Slash or Platformer are some of them. The strongest attribute of an action-based game is the time the player has at disposition for a particular decision. The faster the players make their input, the faster the reaction of the game. "Diablo III" is a good example. The player controls the avatar with a top down perspective and kills dozens of monsters with one click of a button. A fast, experienced player can defeat more and stronger opponents with quicker inputs.



Fig. 3: Diablo III - Top down action based hack and slay combat system

Different are turn-based games like card games, puzzles or even strategy games like chess. They normally give the player a potentially infinite amount of decision time. But only a limited amount of inputs can be made, before the opponent player gets its turn. Real time strategy games are a mix out of both of the previous categories. While being an action based game where the input speed does make a difference, the strategic elements and the display of information are very similar to turn based games.

Apart from the input and output differences between these three possibilities, combat systems differ in display of information, time for strategic decisions and multitasking requirements.

Based on my own observation on different game types I come to the conclusion that the type of game influences the time a player has for decision making, doing multiple tasks at the same time and his ability to process different amounts of information. Strategic decisions have to be made fast in action based games since the input time matters. Because of this the display of information has to be instantly recognizable and in limited amount. The information can be more complex the more time the player can spend on making a decision.

The type of combat changes the complexity of the game:

	<b>Time for decisions</b>	<b>Possible multitasking</b>	<b>Information via interface</b>
<b>Action-based</b>	short	medium	few
<b>Real Time Strategy</b>	medium	high	medium
<b>Turn-based</b>	high	low	a lot

Table 1: Complexity table of different game types.

Additionally to the different complexity, the kind of skill that is needed for the game changes with different combat types. Action-based games need the player to act quickly therefore having a higher demand on physical skill. Fast games are very reactive where a decision has to be made quick before the output will change



and the next action is required again.

In turn-based games the player has to solve more problems with one input changing the required skill to a cognitive one. The player has to think not only about the consequences of their current actions but also have to plan ahead for the next turns.

So to keep the focus on one element the following analysis of combat systems will give examples and discuss the thesis mainly based on action based games. But most of the following design processes and conclusion can be applied to multiple combat systems.

#### **1.4**

#### **The elements of combat systems**

Combat design needs limitations to define itself. Based on Mark Acero's presentation<sup>3</sup> on Combat Design at Game Design Expo 2012, combat design includes elements for locomotion, offensive and defensive.

Let's start with the movement of combat systems as the movement limits the player's offensive and defensive abilities. These restrictions facilitate the definition of the various combat conditions. Then take a look at the offensive design of the avatar. Finally, make sure the players have the ability to defend themselves. With all of the above mentioned elements framed, the developer can design a choreography and assemble various combinations of locomotion, offensive, and defensive to define the feeling of the fight.

---

<sup>3</sup> **Acero, Mark:**  
„Presentation: Combat Design“

### 1.4.1

#### **Locomotion for melee combat systems**

Once the purpose of the fight is known, the constructors must restrict the movement for combat. Can the player run, fly, swim, crawl or hide? For every type of movement, the designer must set the exact procedure.

Suppose we do a melee game with different locomotion states. The avatar of the player has different states in which he can be. Suppose the avatar can run to get from one goal to the next. So we have an idle and operating state. The next state is the combat state in which the character draws a sword and engages an enemy. In this state, the character is moving slowly but precisely in relation to the enemy and is now ready to attack from this movement. The third state can be a stealthed state. Here, the player's movement is slowed down dramatically, and the attack option is limited to an attack animation that can be activated while standing behind the enemy. Idle, run, combat and stealth states define the various moves of the example combat system.

Now let's define the different types of attack states. They can be influenced by interactions with the environment. The avatar may make another attack depending on the distance to the enemy. In this way, we can design a shield charge for the Avatar to reduce the distance, or a powerful shield slam beats the enemy back with a shield by pressing the same button. Another attack state can be activated when the player is at a height. A stone for example. By pressing the attack button twice, the avatar could jump off a edge while holding a sword over his head and dealing massive damage to the enemy on contact.



Fig. 4: Skyrim - Jump attack.

It is important to think about all the different movement options as long as they will be available via combat even when the player can't attack out of these states. When the avatar is climbing a ladder he might not be able to attack, but it's important to think about the possible ways that he can be attacked.

Can the player be hit while dodging or does he get a so called super armor making him immune to knock backs and stunes, or is he completely invincible within the duration? It is important to know how different combat states interact with each other to make it possible to balance the combat. A fast attack with a dagger might not deal much damage, but the attack is fast and allow the players to move right afterwards. A swing with a heavy sword might induce massive amounts of damage and knocks the enemy back, but leaves the player completely vulnerable during the wind up of the attack.

Designing the locomotion is not only important for the combat but the hole game design process. Knowing the maximal movement speed will define the required world size. Knowing how high the avatar can jump will indicate how blockades have to be designed so the players can't reach areas they are not supposed to go.

The locomotion and the different states the avatar can be in will limit the offensive and defensive systems while at the same time opening up design space.

#### **1.4.2**

##### **Offensive option in combat systems**

Before the game designer goes deeper into the development, he needs to know what kind of combat system the game will have. For example, if the game has a range-based combat system, it already dictates which camera angles you can use or not.

For our combat system example, we add an arrow-and-bow attack to the sword-wielding avatar. This gives the player more tactical decisions on how to enter a combat situation. If the player receives only one melee option for combat, his strategic options are defined by the distance at which the mission designers spawn<sup>4</sup> the enemies. By giving bow and arrow to the character, the act of self-positioning is not only running from one enemy to the next, but a strategic decision.

---

<sup>4</sup> **spawning:**

“In video and computer games spawn is the reforming or appearance of a player somewhere in the game level.”; Vengie Beal;

<https://www.webopedia.com/TERM/S/spawn.html>; URL:

<https://www.youtube.com/watch?v=xHG3-Nwc4VM>; Access on 22.03.2019



Fig. 5: Invisible inc - Tactical movement on a grid.

The same applies to the design of the offensive system for enemies. By giving range-attacks to certain opponents, the player must make a tactical decision on where to position himself in order to gain the advantage. Giving the player a range option not only increases the player's strategic potential, it also opens up the scope for the enemies. In this way, enemies can be given the option of blocking arrows with a shield or reducing the distance to the player with a charge. The fight will become more than just a “beat 'em up”-system or a “close up”-brawler.

But we do not only differentiate between melee and ranged combat. To sneak around enemies in stealth oriented games changes the tactical option for the player as well. Sneaking makes the player slower, and allows him to only take paths in which he can't be seen but gives the option to completely ignore fights. Another combat style is mounted combat. Having high speed, a height advantage while at the same time being less able to make sudden turns or cancel attacks is an interesting change. Giving the player all of these different styles to choose

from increases the amount of possible meaningful choices. They also change the pacing of the game. Riding a horse while rushing through enemies is very fast paced while sneaking around gives the players a lot more time to think about their actions. So not only the type of the combat changes the complexity but also the style of the combat the player chooses (compare to : “table 1” complexity table on page 8).

One element of the design the offensive is lethality. Lethality in this context means the different strength for each attack. Mark Acero explained in his presentation on combat design:

*“If all attacks are of same power level than combat becomes a matter of who hits first. That's not fun or interesting”<sup>5</sup>.*

He distinguishes between normal attacks that can occur in a combination by repeatedly pressing the attack button, and heavy attacks that can cause various hit interactions, like smashing them into the air or knocking them back. Heavier attacks normally come with a higher cost like longer vulnerability frames. For more on balancing can be read chapter “**3.4. Combat balancing - Math vs. Player experience**”.

### **1.4.3**

#### **Defensive option in combat systems**

In addition to the offensive for a character is the defense. It's nice to beat yourselves through hordes of monsters, but it's the defense that really makes a player think about a strategy. If you have to choose a heavy attack that will make you vulnerable for a short time, or if you just play it safe by blocking an incoming attack, you make a meaningful choice.

---

<sup>5</sup> Acero, Mark:  
„Presentation: Combat Design“

The most commonly used defense systems are blocking and parrying as well as evasion and toughness. Blocking or parrying attacks are a direct form of defense. The attack is stopped immediately and the player will most likely be pushed back or at least forced into a brief stun. Sometimes the player is able to counter an attack by setting or parrying a block at the right moment.



Fig. 6: For Honor - Parry of attack with a sword.

By dodging the player will be able to roll out of harm's way to completely evade an attack but has to reconsider the new positioning and could be vulnerable to an attack while recovering.

Toughness on the other hand will not negate an attack. The player will get hit and take the consequences. Most commonly the player will lose some kind of hit points upon hit. This is where resistances against special attack type or elements come into play.

On top of that, depending on the game, armor can be a damage reducing factor. Using heavy armor could reduce a lot of damage but maybe slows down the avatar quite a bit. For more on different damage calculation see chapter **”3.3.2 Meaningful choice by balancing”**

#### 1.4.4

### Combat choreography

Knowing the various player movement limitations and the various defense and attack options, it is time to examine the connection of these elements. It is crucial for the game feel to have a good flow between the different states and defining which state can be activated from which state.

One example is having, or not having, animation cancels. Animation canceling is the action of exiting a attack state early. For example entering a dodge while attacking therefore canceling the attack.

Taking a look at Dark Souls III<sup>6</sup> and Bayonetta's<sup>7</sup> combat system it is clear that the different choreographies in each game have a huge impact on the game feel. The player cannot leave an attack state in Dark Souls III once entered. This means that the players have to think carefully about their next attack since they are vulnerable during the animation.

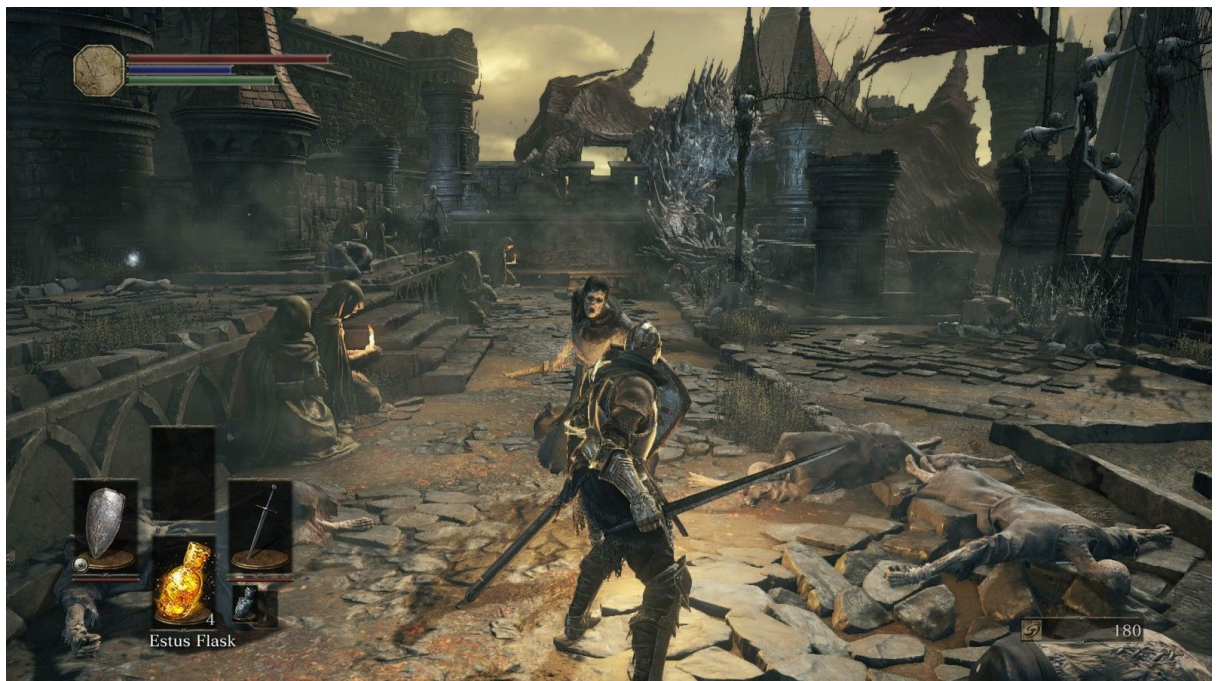


Fig. 7: Dark Souls III - Attacks can't be canceled.

---

<sup>6</sup> **Dark Souls III;**

From Software; Playstation, PC; Release: 24. März 2016

<sup>7</sup> **Bayonetta;**

Sega, Platinum Games, Nex Entertainment; Nintendo Switch, PlayStation 3, Wii U, Xbox 360, Microsoft Windows; Release: 24. März 2016



In Bayonetta, the player has the ability to roll out of the way and avoid an attack at any time. The attack combinations can be fast and the players are not stuck with their decision. They can always change their mind if they want to. This allows for a faster choreography and a smooth combat flow.



Fig. 8: Bayonetta - Backflip to cancel an attack and dodge the enemy attack.

Basically, an attack is just a chain of frames. By defining on which frame the user can make an input for specific combinations and changing the frame on what the transition occurs changes the flow of the choreography. Ensuring that the player has enough tactical and meaningful choices in each state as well as delivering a polished flow of the animation both have a huge impact on the game feel.

Dark Souls and Bayonetta have two very different combat feels. But both are well balanced and well polished which gives a good game feel for the player. These games have a good connection between the states, which leads to a good choreography.

## 2.

### **How does meaningful choice and polished gameplay impact game feel in games**

Balance and design the tactical decisions of meaningful choices or polishing different game elements are two different ways to make a combat choreography feel good. But what exactly is game feel? And how can polish and meaningful decisions improve the gaming experience?

#### 2.1

##### **An approach to define game feel**

There is no direct definition for game feel. Game developers speak a common language, but game feel has never been defined collectively. If you think about the feel of the game, you may have a good idea of what it is, but the moment you begin to define it, it becomes mostly about personal experiences. Steve Swink tried to define game feel in his book “Game Feel A Game Designer’s Guide to Virtual Sensation” as:

*“... game feel is about intuitive controls. A good-feeling game is one that lets players do what they want when they want, without having to think too much about it. Good game feel is about making a game easy to learn but difficult to master. The enjoyment is in the learning, in the perfect balance between player skill and the challenge presented. Feelings of mastery bring their own intrinsic rewards.”<sup>8</sup>*

A good game feel is about the fluent communication between the player and the game via the input of commands and the output of information. The feedback of this information must be exactly in line with the expectations of the player.

---

<sup>8</sup> **Steve Swink:**

“Game Feel A Game Designer’s Guide to Virtual Sensation”; page 1

# Game input loop

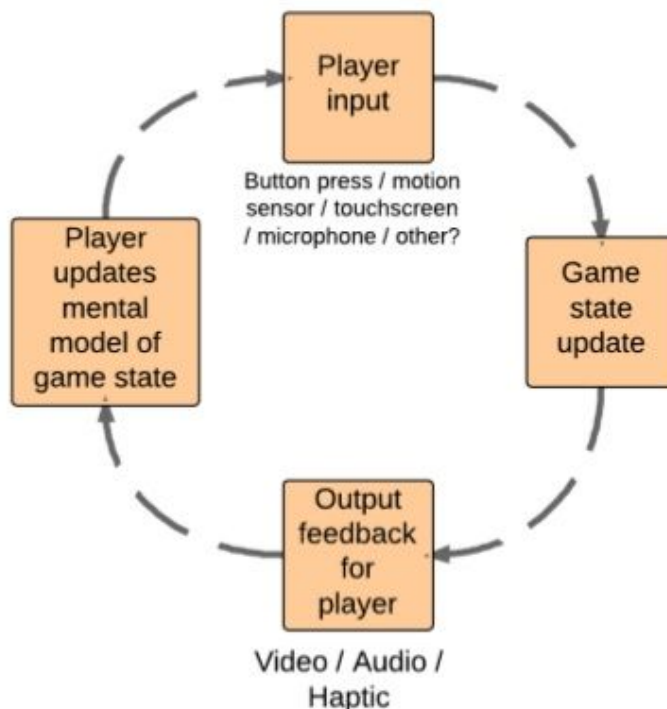


Fig. 9: Player - Game, Input-Output model.

A clear communication from input to output (fig. 9) is important to give the player the right feedback. If players expect a jump by pressing the A-button and are about to jump over a ledge, it is only confusing if they do not reach their destination for some inexplicable reason. However, if the animation of the avatar's indicates that they are slowing down, affecting their speed and thus the jump distance, the players can adjust the movement accordingly or change their actions.

The communication between input and output can encounter different difficulties.

“Easy to learn and hard to master” - this is a criterion for good games that you hear frequently when you evaluate the learning curve of a game. A game with easy access for all types of players allows them to stay motivated until they have learned enough mechanics to actually make it to later stages of the game. Games with a simple start have a stronger hook and attract the attention of the player much easier.

"Difficult to master", however, is the challenge that experienced players seem to find interesting from my own experience. By putting a lot of energy and investment into learning all the mechanisms of the game and their interactions, the player can stand out from the crowd. I've learned in my own observations that it feels good for the player to be rewarded for the effort and dedication one puts into the game. To play a game is to learn. Learn new skills or use skills that you already know to solve a puzzle. And successfully solving a puzzle feels good.

By learning all the possible inputs of the game and getting an idea of the possible outcome, the player can make meaningful decisions. When you solve a puzzle by analyzing the possible options and finding the way to solve it, you get a good feeling. Learning the rules of the game world and understanding the possible options for overcoming an obstacle that the game developers send to you brings their own intrinsic reward.

But Gamefeel is more than just the relationship between input and output and learning the possibilities the game offers to solve a puzzle. The quality of the feedback also has a big impact on the feel of the game. Steve Swink explains:

*“Another camp focuses on physical interactions with virtual objects. It’s all about timing, about making players really feel the impact, about the number of frames each move takes, or about how polished the interactions are.”<sup>9</sup>*

Standing in front of an enemy without a weapon, pressing the attack button and watching as the avatar slowly pulls back his fist and then suddenly bounces. Listening to the sound and enjoying the falling enemy as the avatar recovers a natural position can give players an intense feel when the animations are polished and well timed. There are many examples of good, sophisticated gameplay that allow the player to fully immerse itself in the game and enjoy the interaction for the moment. But polish in games is not just found in attack-animation. It can be as subtle as for example tiny particles that appear

---

<sup>9</sup> **Steve Swink:**

“Game Feel A Game Designer’s Guide to Virtual Sensation”; page 1

when the character is jumping. Even a slight camera shake at certain moments can help to improve the game feel if done properly.

Game feel is important to give the player a positive feedback about their actions. It's mostly an invisible part of the game, but if done right, it gives the feeling of satisfaction.

## 2.2

### What is game polish?

What elements can be polished, and how to polish a game element to get a satisfying feeling?

*“Polish refers to any effect that artificially enhances interaction without changing the underlying simulation. This could mean dust particles at a character’s feet as it slides, a crashing sound when two cars collide, a “camera shake ” to emphasize a weighty impact, or a keyframed animation that makes a character seem to squash and stretch as it moves.”<sup>10</sup>*

Steve Swink defines polish as everything that can be stripped from the game but still lets the game be playable as it was. Every sound, sprite, animation is a polish.

*“When two objects collide and the code tells them that they’re each solid and so should rebound with a certain force in a certain direction, this is not polish. This is part of the simulation, part of the game’s response to input.”*

Swink talks of polishing as a supplement to the core programming of the game and believes that physics in games does not underlie the polishing process. An argument can be made that there are physics that needs to be polished. The way different objects collide with each other can be very different not only on an

---

<sup>10</sup> **Steve Swink:**

Game Feel A Game Designer’s Guide to Virtual Sensation; page 5

animated level but also on a physical level. How far the player slips after different jump angles can have a positive or negative effect on the game feel. But even if the physics can be polished, I consider this type of polish as the process of balancing core mechanics. Changing mechanics at this level affects the player's meaningful choices. Further I will differentiate between effects of polishing game element and polishing mechanics to improve meaningful choice

When I talk about polish in this article, I'm not talking about the individual elements. I rather refer to the refinement of these elements to give the player satisfactory feedback.

*“Polish effects add appeal and emphasize the physical nature of interactions, helping designers sell those objects to the player as real.”*

The player's perception attempts to apply a realistic feedback to interactions within the game, which means the expected behavior of real world physics and consequences. In a video game, when you see a large stone falling next to a small rock, you expect a bigger reaction from the big stone, like a bigger vortex of dust. The more expected and clearer the feedback is, the better the chances are for a satisfactory feedback for the player.

This also includes the pacing between interactions. It is important to change things up. For e.g. if you like explosions, just adding more and more explosions will not give you a good game. Polishing the passing, anticipation and suspense has a huge impact on how the created content is perceived.

Polishing games is one of those elements that pass unnoticed when done well. But it is clearly visible if it is missing or poorly done.

*“In a general sense, our group of developers defined a polished game as one that lacks issues that pull the player out of the gaming experience.”<sup>11</sup>*

---

<sup>11</sup> **Zoss, J. Matthew:**  
„The Art Of Game Polish: Developers Speak“

It is possible that you are working on a combat design and the moves of your main character are feeling weak or not having the powerful impact you hoped for. One reason why this could be is because the pacing of your animation is not well timed.

Emmeline Dobson splits in her presentation “Combat game systems, animation, and flipbook making workshop”<sup>12</sup> the animation of a strike into three different parts: Anticipation or wind-up, strike, and recovery.



Fig. 10: 3 Phases of the animation of an attack.

Every melee attack has those three stages. Resident Evil VI<sup>13</sup> three is a good example. The main character Chris has a finishing move in form of a punch. Chris leans back while he winds up. A sudden strike to the enemy’s face, a short pause on impact, and then a lengthy recovery while Chris’s arm is hanging in the air afterwards, before he pulls his arm back, going into his idle position.

---

<sup>12</sup>

**Dobson, Emmeline:**

„Combat game systems, animation, and flipbook making workshop“

<sup>13</sup> **Resident Evil 6:**

Playstation 3, Xbox 360; Capcom, Release: 2. Oktober 2012



Fig. 11: Resident Evil VI: Punch wind up.



Fig. 12: Resident Evil VI: Punch Hit.





Fig. 13: Resident Evil VI: Recovery to normal state.

In this game with 60 frames per second the anticipation phase is 40 frames, the hit itself just 11 frames, and the recovery time 64 frames long. This ratio gives the punch the feeling of a powerful attack. Looking closer at the attack animation you can see a shatter and blur effect upon the impact and a screenshake during the first part of the recovery phase.

It is not only important that every attack feels strong and powerful. Rather it is important to display the sort of damage that has been done to the target. Sure, the designers could just display the damage done to the enemy with a digital number displaying the amount of damage. But polishing a game to the extent that the player receives the actual feedback of the strength of the attack being performed is part of a good game feel.

The swinging of a broad sword with immense strength in a wide arc towards the enemy can be supported by a strong knockback, slashing gore, a screenshake or different camera effects, a loud impact sound and a clear animation of the pain the enemy is in. This kind of attack has a different polish than a short backstab with a knife. A sudden strike, an enemy screaming through the hand of the attack and slowly falling to the ground.

Depending on the how the attack is displayed and polished, both attacks can feel very strong, even if much less strength is involved when hitting with a dagger. A short left punch has more subtle polish than a strong right, but this does not mean a combination of left punches cannot feel rewarding.

Polishing game elements has a huge impact on the game feel. Giving the player a reward for the input they have just made, exceeding their expectations with polished animation and designs.

### **2.3**

#### **What are meaningful choices in games?**

There are many ways to give your players seemingly random decisions. Meaningful decisions are more difficult to make. They often cause the player to ask questions such as: Where should I go next? How should I best spend my resources? How fast should I go to the next location? Should I focus more on my offensive or defense? Is it better to play it safe or risky? Should I go for more loot or more experience?

*“A good game gives the player meaningful choices. Not just any choices, but choices that will have a real impact on what happens next, and how the game turns out.”<sup>14</sup>*

Not every choice made in a game can be called meaningful. For example, changing the look of your hero without changing the abilities will not change the way the game will play out. The player has only an illusion of choice.

*“...for example, in a racing game, you might have 50 vehicles to choose from, but if they all drive the same way, it is like having no choice at all.”<sup>15</sup>*

---

<sup>14</sup> **Shell Jesse:**

“The Art of Game Design: A Book of Lenses”; page 179

<sup>15</sup> **Shell Jesse:**

“The Art of Game Design: A Book of Lenses”; page 180

Meaningful decisions are an important part of good games. These decisions should be thought-provoking rather than meaningless. But even if the player makes many different and meaningful decisions, he may not have a free choice at all. If the impact of the decisions is not balanced, a dominant strategy could emerge.

*“You might offer a soldier ten guns, all different, but if one of them is clearly better than the rest, again it is like having no choice at all.”<sup>16</sup>*

In this example the players controlling the soldier might want to use a different weapon but will be weaker by not choosing the stronger strategy. This is called a dominant strategy where choices are offered, but there’s a reason to pick one over the others. Once a dominant strategy is found, it can take the fun out of the game due to a lack of choices to make, and therefore a lack of variety in gameplay. In that case, it seems that there is no choice at all, because choosing another weapon will disadvantage you. These dominant strategies can ruin the fun of a game if it is not balanced properly.

“Dr. Boom” from Blizzard’s Hearthstone<sup>17</sup> is a good example. During the time to its release Dr. Boom was considered overpowered due to the good statline combined with the very strong “Doom Bots” which he was summoning two of the moment he entered the battlefield. The effect of the summoned Doom Bots deals 1-4 damage points upon their death to a random enemy which was considered very strong. There was no card at the time that was as much played in a variety of decks like Dr.Boom. With a card like this it is difficult not to include it in your deck without losing potency. This is a great example of a dominant strategy.

---

<sup>16</sup> **Shell Jesse:**

“The Art of Game Design: A Book of Lenses”; page 180

<sup>17</sup> **Hearthstone:**

Heroes of Warcraft: Blizzard Entertainment; Pc, Mobile; 11. März 2014



Fig. 14: Dr. Boom - Powerful Hearthstone Card.

Meaningful decisions are an important part of good games. However, making meaningful choices does not mean that these choices are pure fun. The challenges faced by the players may exceed their skill. Or they are too easy and the players get bored. A popular theory of the relationship between skill and challenge comes from Mihaly Csikszentmihalyi, a Hungarian psychology professor and author of *Flow: The Psychology of Optimal Experience*, called "the zone"<sup>18</sup>. He defines being in the zone as a state of "flowing" in which a person is completely immersed in what he is doing.

---

<sup>18</sup> Csikszentmihalyi Mihaly:  
The Flow, 1990, p. 74

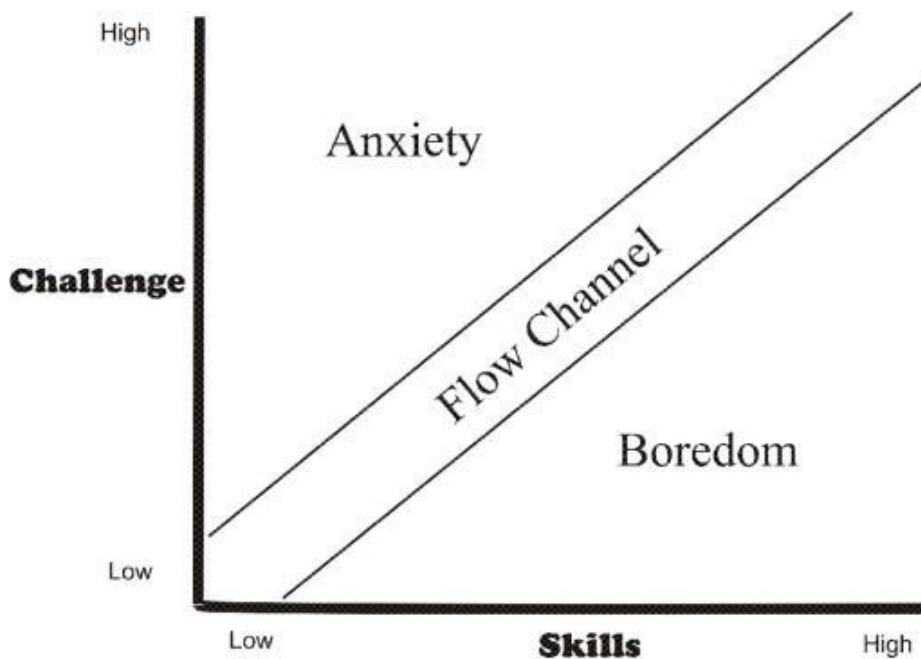


Fig. 15: The Flow After Mihaly Csikszentmihalyi, *The Flow*(1990) p. 74.

If your skills are low and the challenge is simple, or if the challenge is challenging and your skills are high, you can perform in the zone.

However, if you are a massive underdog, you will be overwhelmed by the challenge and you will feel anxiety, not flow. Conversely, if you expect to easily beat a weak opponent, you'll get bored by the challenge after a while and fall out of the zone.

In my own experience while working on “Rogue’s Awakening”, a tactical combat game, gamers would get exhausted when being too long on one spot in the flow channel. Challenging the players constantly but not giving them any room to breathe might be too exhausting over time.



Fig. 16: Rogue's Awakening - Turn based combat system.

Rogue's Awakening challenges the player over a long period of battle with many different combinations and strategies that must be found in order to solve the puzzle that the enemy composition gives the player to beat them.

The game was tested with a version where the players had to fight in endless battles. They were supposed to survive as long as they could. The feedback was that players could not take a break even if they were challenged to the correct amount. The game forced them to make meaningful decisions throughout the game. Any decision could put the player on a path to win or lose. The players felt exhausted by this pressure after a while.

In a later test phase the version of the game included a campfire scene between the fight, where the player could rest and make smaller actions, such as switch items between heroes, choosing the next opponent, reading the skills of the characters, level them up or learn new skills. This phase gives the player room to recover from an intense fight in which every decision was made under high pressure. The pacing between a phase of high pressure and relaxing increases the players stamina and lets them play longer sessions.



Fig. 17: Rogues Awakening - Campfire scene a place to recover after a fight.

Jesse shell describes this phenomenon in his book as “cycle of tense and release”:

*“As I proceed through the game, the enemies grow more numerous, increasing the challenge. If I rise to the challenge, though, and defeat enough enemies, I might be rewarded with a gun that lets me destroy the enemies with only two shots. Suddenly the game is easier, which is very rewarding. This easy period doesn’t last though, because soon new enemies that take three and even four shots to destroy, even with my new gun, will start to appear, taking the challenge to new heights.”<sup>19</sup>*

<sup>19</sup> Shell Jesse:

“The Art of Game Design: A Book of Lenses”; page 121

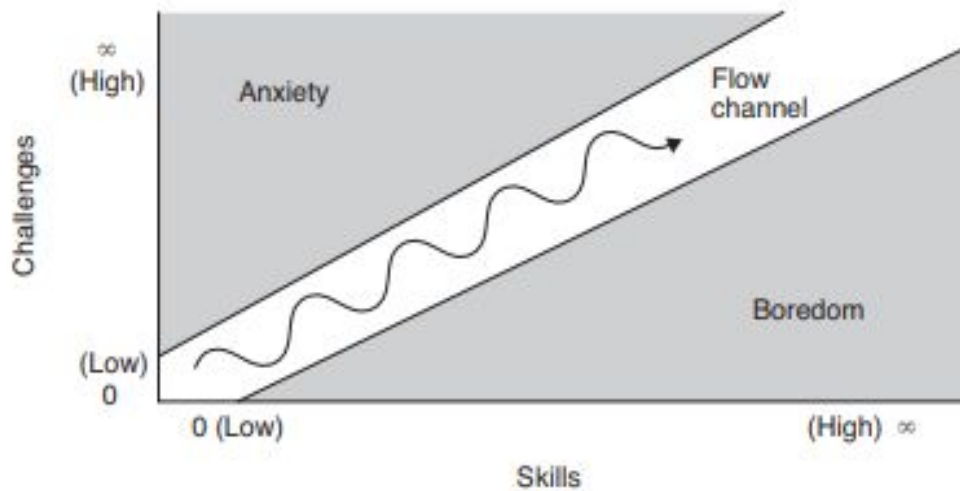


Fig. 18: Balance Type #2: Challenge vs. skill.

*“This cycle of “tense and release, tense and release ” comes up again and again in design. It seems to be inherent to human enjoyment. Too much tension, and we wear out. Too much relaxation, and we grow bored. When we fluctuate between the two, we enjoy both excitement and relaxation, and this oscillation also provides both the pleasure of variety, and the pleasure of anticipation.”<sup>20</sup>*

Rogue's Awakening has a steep learning curve. So the tutorial phase is so broad that the player has enough time to learn the basics of the game until he finds out his own combination and tactics. In most games, the player goes through some kind of learning curve. It is important to know quickly the skills of the player playing the game are increasing in order to be able to design content that increases the difficulty accordingly. This way you keep the player within the Flow and the curve will look similar to figure 18.

But how many options should we give the player? Is having more equal to having better choices? Jessel Shell explained:

<sup>20</sup> **Shell Jesse:**

“The Art of Game Design: A Book of Lenses”; page 122



*“If Choices > Desires, then the player is overwhelmed.*

*If Choices < Desires, the player is frustrated.*

*If Choices = Desires, the player has a feeling of freedom and fulfillment.”*

For an example of an overwhelming amount of choices let's take a look at League of Legends<sup>21</sup>. As of 25 January 2019 there are currently 143 released champions<sup>22</sup>. If a new player has this number of characters to choose from, that's too much if he's never played the game before.



Fig. 19: League of Legends - Immense amount of different characters to choose from.

To determine the amount of choices the designers should give the player, they have to find out what the player wants to do. Giving the player a choice between paying three different heroes might give the player some kind of control he might enjoy. But choosing between over 140 different heroes, that you know nothing about them, might be overwhelming. Knowing how much information the player can process to make the right choices can dramatically improve the feel of the game.

---

<sup>21</sup> **League of Legends:**  
Riot Games; PC; Release 27. Oktober 2009

<sup>22</sup> **Leagueoflegends.fandom :**  
„List of champions“

To take away from this:

Having the right amount of meaningful, balanced choices in your game and space the pacing just right while increasing the difficulty relatively to the increasing skill of the player is the key to a good game feel.

## 2.4

### Why do meaningful choices and polishing not go together

At its core, refining meaningful choices and polishing the game are both approaches to improving the game feel. They even complement each other, as a lack of choice can be overlaid with an effective polish. So, why not use both?

It comes down to time management. Polish takes time, but is often scheduled only in the buffer time. The development of a system that allows meaningful decisions is also not completed in a few days. Aron Davis wrote in his article “Should You Polish Your Game, or Add More Depth and Features?”:

*“Although some days it might feel like you've got all the time in the world, the truth is you don't. When it comes to game development, your time, your finances, and the rest of your resources are always going to be limited.”<sup>23</sup>*

The main problem is time as the most valuable resource in the development process. Epic's Fergusson said in “The Art Of Game Polish: Developers Speak” by J. Matthew Zoss

*“A team achieves polish by allocating time in the schedule for polish, understanding its quality bar and by playing the game over and over.”<sup>24</sup>*

---

<sup>23</sup> **Aron, Davis:**

„Should You Polish Your Game, or Add More Depth and Features?“

<sup>24</sup> **Zoss, Matthew:**

„The Art Of Game Polish: Developers Speak“

He explains that it is important to give polish their own segment in the project management, instead of just allocating it to the buffer time.

Balancing the gameplay, adding new features, testing the game, and coming to the conclusion features have to be removed to increase meaningful choice and therefore game feel. This costs time and resources. So does polishing animations, camera movement, visual effects and sound. Choosing to go for one or the other might depend on a number of things.

First, the nature of the game. Some games simply do not need a lot of polish because the audience enjoys simplicity and focuses more on meaningful decisions and depth in the game. Other games like to tell a story and need less depth, but want more polish to give the player the perfect feeling for the moment.

It also depends on the skills of your team. If you have more designers than programmers, it may be more lucrative to opt for polish. Here, it is probably the best choice to deal more with polish.

Or the game is just not fun. Figuring out if more depth and mechanics or more polish is needed can be very individual and depending on the state of the game.

There are a lot of factors why working on both could give the game a better feel. This article attempts to define the influence of polish and depth on the development of a combat system. In particular, the effects on the design of enemies, the player character, the design of the user interface and the balancing process.

### **3.**

#### **Meaningful choices and polished gameplay in different areas of combat design**

Now we have established the basic terminology for game feel, game polish and meaningful choice. Based on these concepts, different components of combat design are analyzed below.

How can you design a combat system that makes you think carefully about every single move you make? A game that is so deep you will still learn new mechanics after a hundred hours you are in. A game that just feels satisfying when you score a perfectly timed block.

To give an answer to these questions we will take a look at enemy design, player avatar design, the display of information and the basics of combat balancing.

Like most things in game design there is no one size that fit solution. The different needs for each combat system are very specific. So going forward I will attempt to work out the underlying structure and basics on a good design by analyzing specific games. Then I will work out the impact of meaningful choice and game polish for each topic.

#### **3.1**

##### **Combat design: Enemy and player character**

Designing a good combat system is very similar to designing a puzzle. The complexity table on page 8 shows the different complexity levels of different combat types. But independent of the individual complexity each encounter of any of the combat types can be described as a puzzle that has been designed for the player to solve.

You design a question in a given world and give the player the tools to solve it. This puzzle can be easy or hard but it is always a task for the player to solve.

There are two ways to start the design. Begin with the player design first, then design enemies that are shaped to amplify the skills of the player or design a baseline of enemy types first and then design a toolkit for the player to be able to deal with them.

From my personal experience on designing a combat system, it is easier to begin by designing the question before giving the players the tools to answer it.

However, the design progress is iterative and beginning with the one or the other might depend on the type of game you are making.

### **3.1.1**

#### **Enemy design: Design a puzzle for the player to solve**

But how can the puzzle change the players game feel? Well, if we think back to the offence and defense in combat design: Ideally we do not want to give the player to aggressively chue through enemies without ever thinking about defense. But we also don't want the player to hide behind its defense and just wait for the right time to parry. An exciting and dynamic back-and-forth between attack and defense should be encouraged. The best way to do this is by an interesting enemy design. Interesting enemies are a major way to encourage the player to use the full moveset on offer and not rely on just one move to beat the game.

So let's take a look at Super Mario World's<sup>25</sup> enemy design and analyze the game polish and the meaningful choices given to the player.

Super Mario World is a popular Super Nintendo side-scrolling platformer from 1990 with a very simple combat system. This is a good example because the simple combat system highlights the enemy design.

---

<sup>25</sup> **Super Mario World:**

Riot Games; Super Nintendo; Release 21. November 1990

Mario and Luigi travel through different environments on their journey to save a princess. On this journey they encounter various enemies. Some are moving, swimming or flying. Others are stationary and only move on close contact. The enemies are strategically placed on the map and give the player a puzzle to solve. The players main option to deal with these enemies is to jump on their head or outrun them. There are some other options to deal with the enemies but for the sake of simplicity let's focus on jumping on the head.

Firstly there are goombas. They are never introduced as enemies, but the design gives very clear hints about the hostile nature of the entity. They have sharp teeth and angry looking eyebrows. But their shape is very round, especially their head. This tells the players that they can squash them down by jumping on their head. On the other hand, spikes on their back would indicate that the enemy is not vulnerable by jumping on them. For example "Spiny", a spiked turtle. This patrolling enemy has sharp spikes on his back indicating that jumping on its back will result in taking damage. This clear design is good polish. The enemy designs show exactly what they need to show. Their movement speed fits the length of the legs and their weak points are shown by a look at their silhouette. Jumping on a head of an enemy gives a nice sound feedback, as well as a cloud of dust with some stars shooting out.

But what about meaningful choice and strategic combat? Enemies in Super Mario World do not have a complex artificial intelligence (A.I.). They follow very simple patterns. The players can analyze these and change their strategy accordingly. The simple A.I. allows for a clear level design in which the enemies allow for meaningful choice and encourage different strategies to beat the level.

Another example is the green "Koopa Troopa". This enemy is patrolling on a route within the map. If the player gets hit he loses a live or gets shrunk down if he is using a mushroom. But jumping on the head pops out the Koopa and separates it from its shell. In its new state it can be killed by jumping on the vulnerable body. On top of that, the shell itself can be hold, carried around and thrown at other enemies to knock them out.

The player can throw the shell to attack not only enemies, but also to break blocks, or to activate power-ups. By this design the puzzle itself becomes a tool to solve puzzle of how to defeat the level and reach the finish.

Often enemies are designed and reused as subclasses. In which their abilities change but their core design is kept the same.

*“Designing subclasses is a very good way to increase or decrease the difficulty of a class without breaking the defining rules of a class.”<sup>26</sup>*

The Koopa Troopa has a sub-arch type. The flying Koopa. The flying Koopa gets in addition to its normal abilities wings that allows him to patrol on a predetermined flight pattern. This way the enemy is not only an obstacle on the ground that can be jumped over and ignored. Jumping on its head does not shoot the Koopa out of the shell yet. The first jump removes the wings and the Koopa falls to the ground losing its ability to fly.

*“...one of the most engaging feelings a player can experience with a video game is to feel smart and proud of his or her cleverness. And a combat system is a great tool to let the player experience this feeling.”*

This should be considered when designing or analyzing the enemy design. The goal should always be to enhance the gaming experience by either polishing the interaction and viewing of items or improving the strategic gameplay.

Imagine an enemy in Super Mario World that can only be killed by a fireball attack or using a very specific weapon? Implementing enemies that can only be killed by one specific move, or action in very specific situation does not improve the players strategic choices. It simply reinforces a dominant strategy.

---

<sup>26</sup> **Lambottin, Sebastien:**

„The Fundamental Pillars of a Combat System“; page 2

For example: To improve the strategic choices for the player with enemy design the enemy could adapt to the player's inputs. In Super Mario World this is not possible due to the very simple A.I. But in "Middle Earth: Shadow of Mordor" it is. Here the players encounter an enemy named "Ghûra Stonefist". They can jump over the head of this enemy a few times to avoid his attacks and get themselves to a favourable position. But after some jumps the enemy knocks the players away when they attempted to jump over his head again, forcing them to search for a new way to deal with this new situation. Having enemies adapting to your personal playstyle makes sure that the player uses every tool in its arsenal. Since the players could not jump over his head anymore they have to dodge his attacks with his shield. The Orc winds up his shield and twirls it around for a few moments before he leans forward and smashes the shield down dealing an area of effect wave upon impact. This attack will knock the players over if they don't roll away in time. But the very clear communication and the long expectation phase allow the players to dodge in time. Shadow of Mordor is a good example of game polish such as camera movement, sounds and animations helps keep track of what happens at a given time in combat.

To take away for meaningful choice:

- Combat design is a puzzle and enemies are obstacles to overcome.
- Make sure the impact of the meaningful choices lets the players feel good about themselves.
- Try to avoid dominant strategies.
- Include mechanics that lets the player search experience all of their abilities

To take away for polish:

- Use design to make enemy behavior clear
- Use anticipation frames to make attacks visible
- Use sound and camera movement to highlight specific moments for visibility and better understanding and also for a better game feel.
- Get a good balance between special effects for game feel and clear communication to understand what is happening.



The purpose of an enemy is to give a challenge to the player. Sebastien Lambottin states in his article on gamasutra “The Fundamental Pillars of a Combat System“ the main function of an enemy:

*“Define a precise challenge for each enemy. Because the main function of an enemy is to attack the player, and because most of the time the player can destroy the enemies, there are usually two important questions I ask myself when designing the behavior of an enemy:*

*How can the player attack and destroy an enemy?*

*How can the player defend himself against an enemy?”<sup>27</sup>*

To answer his questions we will take a look at the player’s toolkit: Abilities.

### **3.1.2**

#### **Player character abilities: Give the player the right tools**

The main objective of Combat Designs is to develop a system in which the player can select the right skill for the right situation at the right time.

*“The main objective we have in mind when we design the gameplay mechanics of a combat system is to push the player to make clever choices and use the right ability at the right time. We want the player to be able to anticipate the next action he’ll perform and also to develop a tactical plan during the combat.”<sup>28</sup>*

---

<sup>27</sup> **Lambottin, Sebastien:**

„The Fundamental Pillars of a Combat System“; page 2

<sup>28</sup> **Lambottin, Sebastien:**

„The Fundamental Pillars of a Combat System“; page 2

Sebastien Lambottin believes that a good tactical fight must give players the opportunity to make the right decisions and develop a strategy. They decide where to go, what goal they want to focus on, and what skill they want to use. Player character abilities are one of the most important combat features that can be developed for a combat system. They are a collection of tactical decisions from which the player can choose. But how to design a good ability and what properties do these need to have?

### **3.1.2.1**

#### **Properties of player character abilities**

Abilities can take many different forms, depending on the type of combat for which they were built. The players could swinging with a sword, blocking with a shield, or throwing a grenade, shooting an arrow or casting a blizzard. But all these skills must have one thing in common - they have to be tactical. Sebastien Lambottin says that to make an ability tactical, they must have two things:

*“Each ability has a unique function: hit a specific area, stun an enemy...  
Each ability is balanced with the reward vs. the risk of using it.”<sup>29</sup>*

However, it is not enough to give each ability its own function. In my own experience, a bloated combat system can ruin a game. Rogue's Awakening started with many skills. But play tests showed that only a few skills that were quickly understood were selected. The player was overwhelmed and the complexity did not allow the depth of the combat system to take shape.

The more mechanics the designers implement in the system, the more fragile it becomes. So, if you keep a system clean, you can better communicate how the system works.

---

<sup>29</sup> **Lambottin, Sebastien:**

„The Fundamental Pillars of a Combat System“; page 1



Fig. 20: World of Warcraft - Complex combat system with a huge amount of effects.

Figure 20 shows a screenshot of a user interface of Blizzard's game "World of Warcraft"<sup>30</sup>. World of Warcraft is known for the variety of abilities per class. Not to say, it has a bad combat system, but it definitely has an overwhelming one. Even though each skill has a different effect, it takes a while for the player to learn the various possibilities and interactions. Just think of Fig. 19 and the overwhelming state of character selection in the League of Legends. Complexity is not bad, but an overwhelmed state is.

### 3.1.2.2

#### How abilities give meaningful choices

Regardless of its form, each ability must have a unique function. For example: A sword swing is used for close combat, a bow shot for ranged attacks, potions aids with healing or buff effects, a grapple hook offers movement options and a dodge roll can evade attacks.

<sup>30</sup> World of Warcraft:  
Blizzard Entertainment; PC; 23. November 2004

All these abilities are unique and give the player more tactical decisions. However, it is not enough to equip abilities with a unique feature. The effect of each ability must be different. It is not enough to give the player knives for throwing at the enemy, and let them behave just like a shot with a bow. However, changing the speed, damage, and resource costs gives the player a sensible tactical decision.

The goal for a good design of the abilities is to challenge the player's cleverness, says Sebastien Lambottin:

*“... when we design a combat system, we are really aiming to challenge the cleverness of the player, and the tactics he'll be able to apply during the battle. So basically we want a system with multiple choices, but in which the player has to evaluate and choose the best option for each situation.”<sup>31</sup>*

Not only do skills need to be unique, but the mechanics must be clear in their impact and effect so the player can make a meaningful choice.

Let's assume the player chooses to swing a sword at an enemy. They might choose between a fast stab or a heavy blow. A fast stab will give the player a low damaging effect but it comes with low risk. Striking fast could be a valid choice of the player interrupting an enemy's slower attack. But a heavy blow can give the player the damage that is needed to defeat the enemy. On the other hand a slow swing makes the player vulnerable for a counterattack making it a high risk, high reward ability. Giving the player these tools gives them the decision on how to solve the puzzle.

Suppose the player swings a sword at an enemy. They could choose between a quick stab or a heavy swing. A fast stab gives the player a small damage effect, but carries a low risk. A quick hit can be a good choice if the player interrupts an opponent's slower attack. However, a heavy swing from the player can deal the damage needed to defeat the enemy. On the other hand, a slow swing makes the player vulnerable to a counter-attack, making the attack a high-risk,

---

<sup>31</sup> **Lambottin, Sebastien:**

„The Fundamental Pillars of a Combat System“; page 1

high-rewarding choice. By giving these tools to the player, the players choose how they want to solve the puzzle.

Instead of just giving the player unique abilities a good approach to design abilities is to think from the players perspective.

What decisions does the player make? Where can he choose different paths for different results? For real-time combat systems Sebastien Lambottin answer these questions with the three challenges the player evaluates when choosing an ability:

*“Evaluate the distance  
Evaluate the time  
Cleverness and anticipation”<sup>32</sup>*

First, players evaluate the distance to an enemy decide which skill to perform. At the same time they evaluate where the character will end up once the ability is performed.

Secondly, they evaluate the time it takes to perform an ability and anticipate how long the ability will last, since everything evolves in time.

Finally, they anticipate a combination of actions to perform in different situations or which ability will counter the incoming attack of an enemy.

In this way, a skill becomes the right tool to solve a particular puzzle while challenging the intellect by making a meaningful choice.

Now we make sure that the players know that they made an impact with their meaningful choice.

---

<sup>32</sup> **Lambottin, Sebastien:**  
„The Fundamental Pillars of a Combat System“; page 1

### 3.1.2.3

#### Why abilities need polish

But how do players know they just made an impact? The short answer is: By showing the player that they did. However, this simple answer becomes more complicated in practice. Mostly, the player has an expectation of what will happen, and the game must meet or exceed those expectations. But not only does the information has to fit, the design also has to match the overall design of the game to improve immersion.

Providing the player with an immersive experience enhances a positive gaming experience, says Jamie Madigan:

*“Completeness of sensory information means that the fewer blanks about the mental model of the game world that the player has to fill in, the better.”<sup>33</sup>*

Players create a mental model of the game world and decide how they want to perceive it. Keeping this model consistent is called immersion:

*“So, basically, the process starts with players forming a mental model of the game's make-believe space by looking at various cues (images, movement, sounds, and so forth) as well as assumptions about the world that they may bring to the table.”<sup>34</sup>*

Basically, the players form a world based on the real world in which they are currently located. It does not have to be the same, it just has to make sense to the perception of the player. Everything has to fit into this new world and when it does, immersion occurs. This also means that the expectations in this world can be different from those of the real world. Most importantly, all expectations must be met in order not to break the immersion.

---

<sup>33</sup> **Madigan, Jamie:**

„Analysis: The Psychology of Immersion in Video Games“; page 1

<sup>34</sup> **Madigan, Jamie:**

„Analysis: The Psychology of Immersion in Video Games“; page 1

*“Believable behavior from things in the game world means that characters, objects, and other creatures in the game world behave like you’d expect them to.*

*It’s also worth noting that the cues need to make sense and be constant throughout the experience.”<sup>35</sup>*

This means that you expect a horse to move faster than a donkey. The same applies for a blow with the hammer, that knocks the target further away than a sword would. The designers can make slight iterations based on the expected behavior if it gives the players a positive feedback for their actions and do not break immersion.

For example, a shot with a bow can knock back an enemy to give the attack additional flair, even if in this reality a shot with an arrow does not have much recoil potential.

Polishing skills to make sure that the player understands the choice made and challenges the player's intellect keep them in suspense and immersed:

*“Cognitively demanding environments where players have to focus on what’s going on and getting by in the game will tie up mental resources. This is good for immersion, because if brain power is allocated to understanding or navigating the world, it’s not free to notice all its problems or shortcomings that would otherwise remind them that they’re playing a game.”<sup>36</sup>*

This includes a seamless transition between input and output. Polishing the communication to the player is a process that needs various design iterations and testing until the player gets exactly the feeling the designers are aiming for.

For example, the decision as to whether an additional user interface is needed for visualization can change the feel of the game. A floating number over the

---

<sup>35</sup> **Madigan, Jamie:**

„Analysis: The Psychology of Immersion in Video Games“; page 1

<sup>36</sup> **Madigan, Jamie:**

„Analysis: The Psychology of Immersion in Video Games“; page 1

opponent, or the loss of blood of the opponent, are both possible display representations of the damage taken.

To decide what interface is best for a game it is important to set a goal that the game wants to achieve and then make decisions accordingly.

Regardless of the form, the main goal when polishing abilities is to make sure the players get a good feedback to reward the players for their decision they just made. Making a decision and seeing the impact with a good feeling works well together. Different games often decide for more of one or the other. A good example is Diablo III<sup>37</sup> and Path of Exile<sup>38</sup>.

### **3.2**

#### **Analysis of polished gameplay and meaningful choice design on the example of Diablo III and Path of Exile**

Diablo III and Path of Exile are famous top down action rpgs and dungeon crawler in which players slay their way through a horde of monsters. They are great examples since both have the same core in their gameplay but execute game feel very differently.

In both games the players choose a class, fight through various acts and get stronger by leveling their character and looting items. The biggest difference is how they handle the game feel. Diablo III's focus is mainly on strong graphical polish and feel while Path of Exile gives the players meaningful choice by delivering deep content and complex mechanics.

---

<sup>37</sup> **Diablo III;**

Blizzard Entertainment; Release: 15. Mai 2012

<sup>38</sup> **Path of Exile;**

Grinding Gear Games; Release: 23. Oktober 2013





Fig. 21: Diablo III - Highly polished graphics.



Fig. 22: Path of Exile - Graphic example.

Fig. 21 and 22 show in comparison the graphics of Diablo III vs the look of Path of Exile. It is hard to compete with Blizzards artistic and graphical resources even with Path of Exile's good graphics at that time of their release. But the level of polish gives the player a great game feel due to excellent feedback and satisfying effects while still keeping the game mechanics simple to target many players at once.

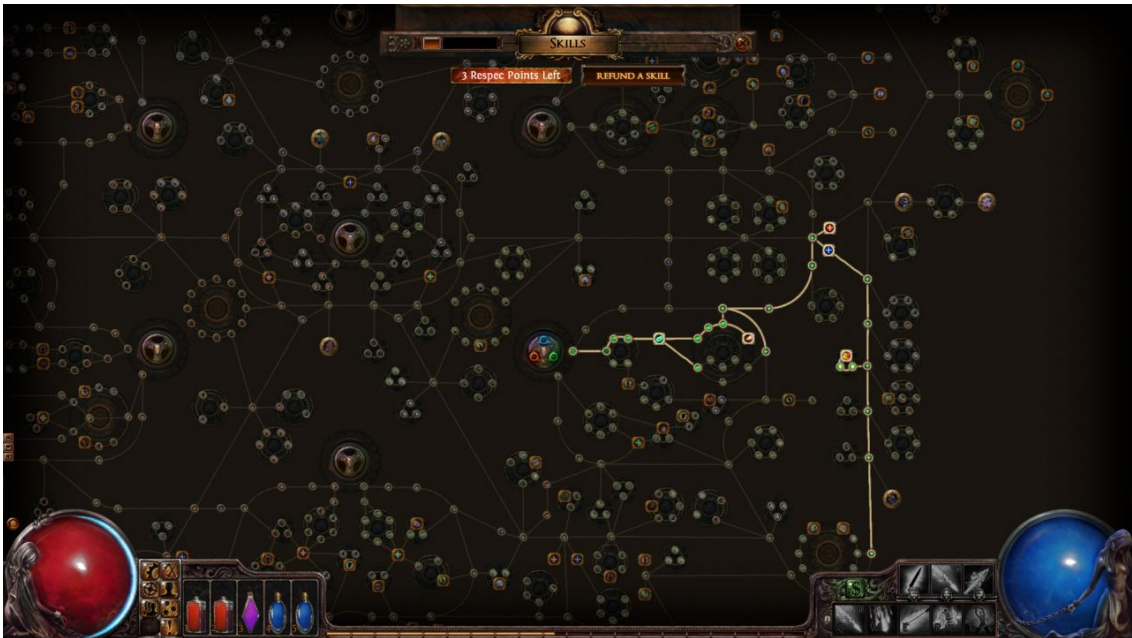


Fig. 23: Path of Exile - Complex skill tree system.

On the other hand, Path of Exile has a very complex skill system in the form of a skill tree, which the player can explore and unlock his own path. Because of the high complexity the character planning and building could be described as a game within a game. The results of this game will affect the playstyle and results of the upcoming fights highly making path of exile a game with a system that support meaningful choice to a high degree.

The core mechanics of Diablo III are quite simple in comparison. They are easier to understand for newer players, but offer few customization options (see Figure 23 and 24).



Fig. 24: Diablo III - Skill System - Selecting runes.

David Allen has published a detailed comparison<sup>39</sup> between Diablo III and Path of Exile, detailing how Path of Exile exceeds gameplay through various endgame content and various character-building options.

He concludes that Diablo III may be more suitable for casual gamers when focusing on categories that improve the player's overall experience. These include a system for trading items, a polished design and feel for the combat, and an easier learning curve for newer players.

---

<sup>39</sup> **Allen, David:**  
 „Diablo 3 vs Path of Exile: A Detailed Comparison”

Path of Exile surpasses with diverse and challenging endgame content and a strong customization, which includes many meaningful decisions in building the character.

Diablo III aims for a much more casual target group with the high amount of polish compared to Path of Exile where more hardcore players, players that are looking for complex game elements and hard challenges. Both games have their charm, but obviously each of the games comprises more development time, either in polish or in mechanics. Diablo 3 wants to please the players through visual satisfaction. Path of Exile wants to involve the player in challenging their own thoughts.

A good polish allows a quick immersion and an easy understanding of the elements. Good mechanics and a well thought-out selection make complexity and depth possible.

### **3.3**

#### **Using the principles of polishing game elements and crafting meaningful choices to design a graphical user interface and balancing content in games**

The design of the information display is a process in game development that needs a lot of polish for working well. It must give the player information without interrupting the immersion.

Balancing is a process that refines the game mechanics to make sure that the player has not only a choice, but furthermore a meaningful choice. The following describes how polish impacts interface design and balancing allows for meaningful choice in combat systems.

### **3.3.1**

#### **Polish - Display Information: GUI & Feedback**

There are many different ways to notify information to the players. When you hit somebody in the face in a boxing match, you can simply see the damage done by hovering a number over your head. Or you can reduce a health bar, as many “beat em-up”-games do.

Another approach is not to use the user interface as an overlay element, but to integrate it into the game flow. Your opponent may bleed, lose teeth, and have blue eyes to indicate the total damage done. The immediate damage can be indicated by the head having different effects on the shot and the player knowing that he has just made a strong hit. Or the opponent's head could just stay as it is which would mean that its very tough. Designing an interface is a process of polishing.

Regardless of how the UI is designed, a UI is always a display of information and an answer to an input from the player. As described above, apparently uncomplicated communication with the player is important to maintain immersion and to provide a good feel.

A fantastic game design with real depth is only good if players can access it. Games hide their depth behind more or less complexity.

Depth in this context means that a player has access to variety of game content within the game and the ability to make meaningful decisions.

When I speak about complexity, I'm referring to the number, frequency, and difficulty of mental calculations a player must perform throughout the game within a given time frame. The less time the player has to solve a problem the more complex it becomes.



Fig. 25: Dwarf Fortress - Roguelike simulator with high depth but unintuitive interface

If the complexity is not properly broken up for the player, he will be overwhelmed or even easily lost.

The image above shows the interface for Dwarf Fortress, where the player controls dwarves and builds a huge fortress to defend himself against attackers. In this game, dwarves have complex individual needs, can get depression, get married, have mood swings, or just are too lazy to get to work. Thousands of events can occur. Dragons can attack the entrance gate, melting the whole fortress down, or a dwarf can be bitten by a werewolf without anyone noticing, and become a ticking time bomb before it transforms and infects others. It is one of the deepest games ever. However, the user interface is not intuitive at all. The players cannot use the mouse to make inputs. They must remember key combinations to make different commands. Also, the output of information is unusual. Figure 25 shows a city on a river with hundreds of inhabitants. But it is displayed in ASCII code. It takes a while for the player to really understand the individual elements and combinations to oversee what's going on. The ASCII code may have its charm, but is not involved in a clear visualization of the elements.

In my experience, it's easy to recognize different structures and interacting with the game as soon as the player understands the interface language in Dwarf Fortress. But it takes a long time to get there.

It's great when players make a meaningful choice, but players need to know they're making one. A good story and lots of details make for an immersive feeling. However, it is not worth it if the players are overwhelmed by the complexity and cannot experience it.

Complex systems allow a deep game, but complexity limits the depth. A large set of rules increases the space in which meaningful decisions can be made. Low complexity, on the other hand, limits the depth.

In Tic Tac Toe for example are very few rules in which players mark one field at a time in a three-by-three grid. The player wins if he achieves three marks in a row first. This complexity space allows very little depth. But it is easy to understand. A possible increasing complexity allows deeper gameplay, but this complexity can quickly become overwhelming.

One way to deal with a complex system is to allow the player to slowly learn the mechanics and integrate an interface that filters the complexity. For example, the game could only display information that is actually relevant to the players when they need it. The UI must sort the information for the player and make it easier for the player to receive it.

Desi Quintans talks in her article “Game UI By Example: A Crash Course in the Good and the Bad”<sup>40</sup> about user interface design:

*“A good UI tells you what you need to know, and then gets out of the way.”*

---

<sup>40</sup> **Quintans, Desi:**

„Game UI By Example: A Crash Course in the Good and the Bad“; page 1

To make it short - The goal is to give information without letting the players work for it or distracting them with large interface elements that do not belong into the game world.

Jamie Madigan states about polishing an interface is to make sure it fits into the game reality and does not break the immersion.:

*“An unbroken presentation of the game world means that the spatial cues about the imaginary world your game has created should not just up and vanish. Which is exactly what happens every time you get a loading screen, a tutorial, or a game menu. When that happens, the game world literally disappears for a few minutes, and we can’t feel immersed in something that isn’t there.”<sup>41</sup>*

Loading screens can be as disruptive to immersion as a phone call while watching a movie. A good example of a successful transition between combat and exploration is “Darkest Dungeon”<sup>42</sup>. Here the screens remain almost unchanged except for the monsters that suddenly appear when changing from the exploration mode into an encounter with an enemy. (Fig. 26).

The Darkest Dungeon core design is to let the player feel the fear his character experiences as he travels through a dreadful dungeon. Not interrupting this immersion is crucial to the success of this concept. For this reason, there is as little transition as possible to immerse the player into the experience.

---

<sup>41</sup> **Madigan, Jamie:**

„Analysis: The Psychology of Immersion in Video Games“; page 1

<sup>42</sup> **Darkest Dungeon;**

Red Hook Studios; Release: 19. Januar 2016





Fig. 26: Darkest Dungeon - A clean transition between exploration and combat

### 3.3.2

#### Meaningful choice by balancing

Good feel can be achieved by polishing the user interface to reduce the complexity that comes with depth and sensible choices. But depth and meaningful choices must be designed and balanced. A combat system can contain elements on many different levels that interact with each other to give players the space to make meaningful choices. Balancing in combat systems is mostly

rearranging mathematical systems that interact with each other and with the players. But it is not the math that the player perceives, but the gaming experience. It is not always best to design a perfectly balanced mathematical system. Sometimes it's best to focus on the gaming experience. The next section analyzes various combat elements and describes their relevance for a meaningful choice.

There are many different elements in a combat system. Those that I will talk about in this chapter are value-based combat systems where the player tries to zero the opponent's hit points before its own. Here, many different variables interact with each other to calculate how much damage has been achieved.

### **Damage calculation: Toughness vs Damage**

Damage is the value that reduces the HP or hit points of the target. The damage is calculated differently in each game. This may depend on the type of weapon or the statistics of the player character. Sometimes it is fixed, sometimes it has a variance and is calculated randomly.

Toughness is the defensive variable that stands between the goal and its defeat. This term includes the number of hit points a character has and all damage reduction values. For example, armor or resistors can reduce incoming damage by a certain amount. How much exactly the damage will be inflicted varies from game to game.

### **Risk vs. reward**

In the chapter on abilities we talked about the effects of various attacks. How a heavy blow needs a long wind-up, making the player vulnerable. To compensate for various attacks, the positive effects must have a certain disadvantage. For e.g when players swing a heavy sword that causes massive damage but takes a long time to execute it. Another common approach to counterbalancing strong attacks is to combine their usage with a cost or give them a cooldown that effectively prevents players from spamming the ability.

Another ability could cleave through enemies and be very effective for crowd controlling, but without much damage being dealt.

Another example is a powerful attack that stuns the opponent on a hit could inflict retail damage on the player to balance the ability. There are many different examples to control the risk of reward.

Sebastian Lambotin listed in his article of the advantages and tradeoffs<sup>43</sup>:

<b>Advantages</b>	Damage Output	Stun	Repel	Damage over Time	Blindness	HP Regeneration
<b>Trade-Offs</b>	Consumable Points	Cool-down	Time to Activate	Recovery Time		

### **(Player chosen) Difficulty -**

Difficulty is handled very differently in many games. If you think about the difficulty in a game often you think about the option before you start a new game in which you are supposed to choose your difficulty for the playthrough. But if you have never played the game before, how are you supposed to know if you have the most fun in a normal or hard playthrough without knowing what will change and what impact this will have?

The difficulty is handled very differently in many games. If you have to choose the difficulty in a game, you often hit the option before starting a new game where you have to choose your difficulty level to play through. But how do games change their difficulty?

Generally it is not a good design to just increase the health points or stats of an enemy. This just makes a fight lasting longer without giving a harder challenge for the mind. Instead you could increase the complexity of the fight to increase

---

<sup>43</sup> **Lambottin, Sebastien:**

„The Fundamental Pillars of a Combat System“; page 1

the room in which the player makes a decision based on risk vs reward management, as explained above.

It is not always needed to increase the difficulty of a game. To make room for more tactical decisions the interface can get more complex. An experienced player is in general able to process more information at the same time. So allowing them to access more information at once might be a good way to customize the difficulty of your game.

### **3.3.3**

#### **Balancing the math with the user experience in mind when buffing or nerfing abilities**

Today, many games patch their content to fix bugs, add mechanisms, or balance content. Especially when a skill or an effect could be too powerful and needs a nerf. Nerfing is the process of changing the attributes of an element and weakening its effect. Strengthening a strength that is too weak to match the current balance is called buffing.

However, changing content may not always be fun for players. Buffing and nerfing skills is a very mathematical process. But just because a skill is perfectly balanced may not always be fun as well. Once a player is accustomed to the power of an ability and enjoys using it in combat, a change can have a very negative impact. In particular, if a player uses a strategies and plans around a mechanic, changing those mechanics could make his work useless. It is better to increase the strength of weak abilities as long as the game mechanics are up to the task. At the mathematical level, it doesn't matter if you buff or nerf content, but players can easily bond to their powerful abilities that make them feel good. For this reason, it is sometimes better not to change the content or find a different solution to the problem.

In your games you are not always looking for a perfect balance. Sometimes it is better to deliberately unbalance the content. Especially in Esports, you want to allow the player to figure out strong tactics and let them make the dominant

strategy in the Meta. In this way, other players try to counteract this strategy until that strategy dominates and so on. By not balancing everything perfectly, the meta is constantly evolving and makes the game more engaging over time.

Always consider what your game wants to do when you balance content. Be careful when changing content. Do not focus mainly on math, but think about the gaming experience. And look for the perfect imbalance in your game.

#### **4.0**

#### **Conclusion - Are meaningful choices for combat systems more relevant than polished gameplay?**

This paper reveals no direct answer to this question. Both can have a high impact on a positive game feel.

Both the polishing of feedback and the design of combat systems, which allow for meaningful choices, require a lot of time. It is important to know what your game wants to do and who it is for by limiting the core elements of the system.

Focus on either casual or hardcore gamers as casual gamers often ask more polish and hardcore gamers for deeper gameplay.

A mechanically driven game requires enough polish to reduce the complexity to a point that allows players to access the depth. This kind of games requires a high degree of balance, so that the systems do not break.

On the other hand games that focus on the feel of the environment or the story that is told need a lot of polish to keep the players immersed.

## **Attachment**

## **Bibliography**

### **Acero, Mark:**

„Presentation: Combat Design“; in: Youtube, 30.10.2012;  
URL: <https://www.youtube.com/watch?v=xHG3-Nwc4VM>; Access on 16.03.2019

### **Aron, Davis:**

„Should You Polish Your Game, or Add More Depth and Features?“;  
in: Gamedevelopment, 8 Jun 2014; URL:  
<https://gamedevelopment.tutsplus.com/articles/should-you-polish-your-game-or-add-more-depth-and-features--cms-21155>; Access on 18.03.2019

### **Allen, David:**

„Diablo 3 vs Path of Exile: A Detailed Comparison“;  
in: requinix; June 2016; URL:  
<https://www.requinix.com/diablo-3-vs-path-of-exile>

### **Csikszemmihalyi Mihaly:**

The Flow; 1990

### **Dobson, Emmeline :**

„Combat game systems, animation, and flipbook making workshop“; in:  
Slideshare, 26. Mai 2014; URL:  
<https://de.slideshare.net/emeraldsong/combat-game-systems-animation-and-flipbook-making-workshop> , page 23; Access on 18.03.2019

### **Leagueoflegends.fandom :**

„List of champions“;  
in: leagueoflegends.fandom., 26. Mai 2014; URL:  
[https://leagueoflegends.fandom.com/wiki/List\\_of\\_champions](https://leagueoflegends.fandom.com/wiki/List_of_champions); Access on 18.03.2019

### **Madigan, Jamie:**

„Analysis: The Psychology of Immersion in Video Games“; in: Gamasutra,  
August 25, 2010; URL:  
[https://www.gamasutra.com/view/news/29910/Analysis\\_The\\_Psychology\\_of\\_Immersion\\_in\\_Video\\_Games.php](https://www.gamasutra.com/view/news/29910/Analysis_The_Psychology_of_Immersion_in_Video_Games.php); Access on 10.05.2019

### **Quintans, Desi:**

“Game UI By Example: A Crash Course in the Good and the Bad”  
<https://gamedevelopment.tutsplus.com/tutorials/game-ui-by-example-a-crash-course-in-the-good-and-the-bad--gamedev-3943> Access on 10.05.2019

### **Shell Jesse:**

The Art of Game Design: A Book of Lenses; Taylor & Francis Ltd;

2nd edition, 2014

**Steve Swink:**

Game Feel A Game Designer's Guide to Virtual Sensation;  
Morgan Kaufmann **Game** Design Books; 2008

**Zoss, J. Matthew:**

„The Art Of Game Polish: Developers Speak“; in: Gamasutra, December 22,  
2009; URL:  
[http://www.gamasutra.com/view/feature/132611/the\\_art\\_of\\_game\\_polish\\_developers\\_.php](http://www.gamasutra.com/view/feature/132611/the_art_of_game_polish_developers_.php); Access on 18.03.2019

## **Game reference**

**Bayonetta;**

Sega, Platinum Games, Nex Entertainment; Nintendo Switch, PlayStation 3, Wii U, Xbox 360, Microsoft Windows; Release: 24. März 2016

**Dark Souls III;**

From Software; Playstation, PC; Release: 24. März 2016

**Darkest Dungeon;**

Red Hook Studios; PC; Release: 19. Januar 2016

**Diablo III;**

Blizzard Entertainment; Release: 15. Mai 2012

**Hearthstone:**

Heroes of Warcraft: Blizzard Entertainment; Pc, Mobile; 11. März 2014

**League of Legends:**

Riot Games; PC; Release 27. Oktober 2009

**Path of Exile;**

Grinding Gear Games; Release: 23. Oktober 2013

**Resident Evil 6:**

Playstation 3, Xbox 360; Capcom, Release: 2. Oktober 2012

## Illustration directory

**Fig. 1: Final Fantasy VII's turn based combat system**

<https://www.technobuffalo.com/wp-content/uploads/2014/01/Final-Fantasy-VII.jpg>

access: 23.03.2019

**Fig. 2: Dark Souls III - Player fights enemies in a action based combat system**

<https://forums.sherdog.com/threads/dark-souls-3.3066567/>

access: 23.03.2019

**Fig. 3: Diablo III**

[https://5images.cgames.de/images/gamestar/279/diablo-3-nachtest-2\\_2351314.jpg](https://5images.cgames.de/images/gamestar/279/diablo-3-nachtest-2_2351314.jpg)

access: 23.03.2019

**Fig 4.: Skyrim - Jump attack**

<https://www.nexusmods.com/skyrim/images/32717/>

access: 23.03.2019

**Fig. 5: Invisible inc - Strategic movement**

<https://www.imore.com/best-strategy-games>

access: 23.03.2019

**Fig. 6: For Honor - Parry of an attack with a sword**

<https://gamerant.com/for-honor-guide-how-to-parry/>

access: 23.03.2019

**Fig. 7: Dark Souls III - Attacks can't be canceled**

<https://www.windowcentral.com/dark-souls-iii-first-impressions>

access: 23.03.2019

**Fig. 8: Bayonetta - Backflip to cancel an attack and dodge the enemy attack**

<https://www.pastemagazine.com/articles/2016/11/the-best-dodge-mechanics-in-games.html>

access: 23.03.2019



**Fig. 9: Player - Game, Input-Output model**

*<https://www.slideshare.net/emeraldsong/combat-game-systems-animation-and-flip-book-making-workshop>, slide 20*  
access: 23.03.2019

**Fig. 10: 3 Phases of the animation of an attack**

*<https://www.slideshare.net/emeraldsong/combat-game-systems-animation-and-flipbook-making-workshop>, slide 24*  
access: 23.03.2019

**Fig. 11: Resident Evil VI: Punch wind up.**

*Resident Evil, Playstation, Screenshot*  
taken: 23.03.2019

**Fig. 12: Resident Evil VI: Punch Hit.**

*Resident Evil, Playstation, Screenshot*  
taken: 23.03.2019

**Fig. 13: Resident Evil VI: Recovery to normal state.**

*Resident Evil, Playstation, Screenshot*  
taken: 23.03.2019

**Fig. 14: Dr. Boom - Powerful Hearthstone Card**

*[https://hearthstone.gamepedia.com/Dr.\\_Boom](https://hearthstone.gamepedia.com/Dr._Boom)*  
access: 23.03.2019

**Fig. 15: The Flow**

*After Mihaly Csikszentmihalyi, *The Flow*(1990) p. 74*  
access: 23.03.2019

**Fig. 16: Rogues Awakening - Combat**

*[https://www.andrejwiehagen.de/?page\\_id=6](https://www.andrejwiehagen.de/?page_id=6)*  
access: 23.03.2019

**Fig. 17: Rogues Awakening - Campfire scene a place to recover after a fight**

<https://levion.itch.io/roguesawakening>

access: 23.03.2019

**Fig. 18: Balance Type #2: Challenge vs. Success**

**Shell Jesse:** The Art of Game Design: A Book of Lenses; Taylor & Francis Ltd;  
2nd edition, 2014, page 121

**Fig. 19: League of Legends - Immense amount of different characters to choose from**

<https://www.fullesports.com/ventajas-desventajas-campeones-complejos/>

access: 23.03.2019

**Fig. 20: World of Warcraft - Complex Combat**

<https://www.denofgeek.com/us/games/world-of-warcraft/240887/>

world-of-warcraft-is-blizzard-dumbing-down-the-experience

access 12.04.2019

**Fig. 21: Diablo III - Highly polished graphics**

<https://www.notebookcheck.net/Diablo-III-Benchmarked.74918.0.html>

access 16.05.2019

**Fig. 23: Path of Exile - Graphics**

<https://steamcommunity.com/sharedfiles/filedetails/?id=1149668295>

access 16.05.2019

**Fig. 23: Path of Exile - Complex skill tree system**

<https://d3rants.wordpress.com/2012/08/01/path-of-exile-vs-diablo-3-2/>

access 16.05.2019

**Fig. 24: Diablo III - Skill System - Selecting runes**

<https://www.gamestar.de/artikel/diablo-3-runen-sind-keine-items-mehr-skills-in-kategorien,2564954.html>

access 16.05.2019

**Fig.25 Dwarf Fortress - Roguelike simulator with high depth.**

<https://arstechnica.com/gaming/2013/04/challenge-accepted-dwarf-fortress->

pros-show-ars-up-with-insane-10-hour-forts/  
access 16.05.2019

**Fig. 26: Darkest Dungeon - Combat System - Clean display of skills**

<https://www.wired.com/2016/01/darkest-dungeon-review/>  
screenshot 01.07.2019